

Eszterházy Károly Főiskola  
Matematikai és Informatikai Intézet

# Komputergrafika

Alcím: Matematikai alapok

Dr. Kovács Emőd

Eger, 2011

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>4</b>
<b>2. Vektorok (<i>Vectors</i>)</b>	<b>5</b>
2.1. Helyvektorok a térben . . . . .	6
2.2. Műveletek vektorokkal . . . . .	7
2.2.1. Vektorok normalizálása ( <i>Normalization</i> ) . . . . .	7
2.2.2. Vektorok összeadása ( <i>Addition</i> ) . . . . .	7
2.2.3. Két vektor különbsége ( <i>Subtraction</i> ) . . . . .	8
2.2.4. Vektor szorzása számmal (skalárral) ( <i>Scalar multiplication</i> ) . . . . .	9
2.2.5. Két vektor skaláris szorzata ( <i>Dot product</i> ) . . . . .	11
2.2.6. Példák skaláris szorzás használatára . . . . .	12
2.2.7. Két vektor vektoriális szorzata ( <i>Cross product</i> ) . . . . .	14
2.2.8. Példák vektoriális szorzat használatára . . . . .	16
<b>3. Mátrixok (<i>Matrices</i>)</b>	<b>18</b>
3.1. Nevezetes mátrixok . . . . .	19
3.2. Mátrix determinánsa ( <i>Determinant</i> ) . . . . .	20
3.3. Mátrixműveletek ( <i>Matrix Operations</i> ) . . . . .	23
3.3.1. Mátrixok összeadása ( <i>Matrix addition and subtraction</i> )	23
3.3.2. Mátrix szorzása számmal ( <i>Matrix skalar multiplication</i> )	24
3.3.3. Mátrixok szorzása ( <i>Matrix multiplication</i> ) . . . . .	25
<b>4. Koordináta-rendszerek (<i>Coordinate system</i>)</b>	<b>27</b>
4.1. Descartes-féle koordináta-rendszer ( <i>Cartesian coordinate system</i> ) . . . . .	27
4.1.1. Kétdimenziós Descartes-féle koordináta-rendszer . . . . .	27
4.1.2. Háromdimenziós Descartes-féle koordináta-rendszer . . . . .	28
4.2. Görbevonalú koordináta-rendszer . . . . .	30
4.2.1. Polárkoordináta-rendszer ( <i>Polar coordinate system</i> ) . . . . .	30
4.2.2. Hengerkoordináta-rendszer ( <i>Cylindrical coordinates</i> ) . . . . .	32
4.2.3. Gömbi koordináta-rendszer ( <i>Spherical coordinates</i> ) . . . . .	33

<b>5. Homogén koordináták</b>	<b>37</b>
5.1. Síkbeli homogén koordináták . . . . .	37
5.2. Két pontra illeszkedő egyenes egyenlete . . . . .	39
5.3. Pont és egyenes távolsága . . . . .	41
5.4. Két egyenes metszéspontjának meghatározása . . . . .	42
5.5. Térbeli homogén koordináták . . . . .	43
<b>6. Ponttranszformációk <i>Linear transformation</i></b>	<b>47</b>
6.1. Egybevágósági transzformáció ( <i>Congruence transformation</i> ) . . . . .	47
6.1.1. Identitás ( <i>Identity</i> ) . . . . .	48
6.1.2. Eltolás ( <i>Translation</i> ) . . . . .	48
6.1.3. Forgatás ( <i>Rotation</i> ) . . . . .	49
6.1.4. Tükrözés ( <i>Reflection</i> ) . . . . .	52
6.2. Hasonlósági transzformáció ( <i>Similarity transformation</i> ) . . . . .	52
6.3. Affin transzformációk ( <i>Affine transformations</i> ) . . . . .	53
6.3.1. Skálázás ( <i>Scaling</i> ) . . . . .	54
6.3.2. Nyírás ( <i>Shear</i> ) . . . . .	55
6.4. Projektív transzformáció ( <i>Projective transformation</i> ) . . . . .	57
6.5. Térbeli transzformációk szorzata ( <i>Concatenation of transformations</i> ) . . . . .	58
6.6. Forgatás a három koordináta-tengely körül . . . . .	61
6.7. Tetszőleges tengely körüli forgatás ( <i>Rotation about an arbitrary axis</i> ) . . . . .	62
6.8. Tetszőleges síkra való tükrözés ( <i>Reflection about an arbitrary plane</i> ) . . . . .	65
<b>7. Koordináta-transzformációk (<i>Coordinatesystem transformation</i>)</b>	<b>68</b>
<b>8. Tér leképezése a síkra</b>	<b>71</b>
8.1. Párhuzamos vetítés . . . . .	72
8.2. Centrális vetítés . . . . .	75

<b>9. Görbék megadása</b>	<b>79</b>
9.1. Interpoláció . . . . .	81
9.1.1. Hermite-görbe . . . . .	81
9.2. Approximáció . . . . .	83
9.3. Bézier-görbe . . . . .	83
9.3.1. A de Casteljaun-algoritmus . . . . .	84
9.3.2. A Bézier-görbe előállítás Bernsteinn-polinonnal . . . . .	85
9.3.3. A Bézier-görbe néhány tulajdonsága . . . . .	85
9.3.4. Harmadfokú Bézier-görbék . . . . .	86
9.3.5. Kapcsolódó Bézier-görbék . . . . .	89
9.3.6. Cardinal spline . . . . .	90
<b>10. B-spline görbe és felület</b>	<b>94</b>
10.1. Normalizált B-spline alapfüggvény . . . . .	94
10.2. B-spline görbe . . . . .	97
10.2.1. Tulajdonságok: . . . . .	97
10.3. B-spline görbék kapcsolódása . . . . .	100
10.4. B-spline görbe előállítás . . . . .	101
10.4.1. Cox-de Boor algoritmus . . . . .	102
10.5. Interpoláció B-spline görbével . . . . .	106
10.6. Racionális görbék . . . . .	108
10.6.1. Racionális Bézier görbe . . . . .	109
10.6.2. Racionális B-spline görbe, NURBS . . . . .	110
10.7. B-spline felület . . . . .	111
<b>11. Függelék</b>	<b>114</b>
11.1. Window-viewport transzformációk . . . . .	114
11.1.1. Uniform eszköz transzformáció . . . . .	116
11.1.2. Window-viewport transzformáció 2. változat . . . . .	118
11.2. Programkódok . . . . .	120
<b>Irodalomjegyzék</b>	<b>131</b>

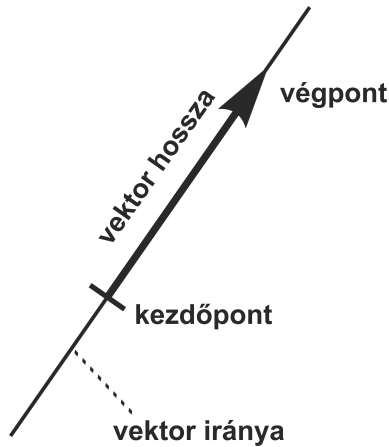
# 1. Bevezetés

A komputergrafika az informatika (Computer Science) egyik legdinamikusabban fejlődő területe. Ma már szinte minden számítógép iránt egy kicsit is érdeklődő ember fantáziáját megmozgatják a mai fantasztikus grafikai képességek, amire korunk eszközei lehetőséget teremtenek. Rengetegen elkezdenek foglalkozni a számítógépi grafikával, de igen hamar szembetalálkoznak olyan matematikai hiányosságokkal, ami a későbbi továbbhaladásukat megakadályozza. Ez a jegyzet elsősorban a programtervező informatikusok BSc hallgatóknak készült, ahol a képzési programban egy bevezető grafikai kurzus után lehetőség van a komputergrafikát emelt szinten is tanulni. Mivel nagyon különböző előképzettséggel, és matematikai tudással jelentkeznek a hallgatók, ezért tartottuk szükségesnek a jegyzet elkészítését. Bizonyos helyeken – elsősorban a címeznél – a magyar elnevezés angol terminológiában használt megfelelőjét is megadjuk, ezzel az idegen nyelvű irodalomban való tájékozódást kívánjuk segíteni. A példákat C#-ban készítettem el, a kódolás során pedig a Microsoft Visual Studio környezetet használtam. A programkódokat a könyv végén külön fejezetben lehet megtalálni. Mivel ez a jegyzet ingyenesen elérhető mindenki számára, reméljük, hogy nemcsak a hallgatók, hanem más érdeklődők is szívesen olvassák. Minden esetleges hibát, észrevételt szívesen veszünk a kedves olvasótól az alábbi email címen: [emod@ektf.hu](mailto:emod@ektf.hu). Az olvasók segítségével, az elektronikus jegyzet adta formai lehetőség révén a javítást rövid időn belül tudjuk megtenni.

## 2. Vektorok (*Vectors*)

[105] A vektorokat a legkülönbözőbb összefüggésekben használjuk a komputergrafikában. Használatosak például az árnyalásnál, ahol szükségünk van felület normálisa és a fénysugár iránya által bezárt szögre. A számítógépes játékokban ábrázolhatjuk egy tárgy mozgásának irányát és sebességét is vektorral.

A vektor két jellemző adattal rendelkezik: nagysággal és iránnyal (beleértve az irányítást is). A vektort az iránya (különbözteti meg a skaláris mennyiségektől, amelyeknek csak nagyságuk van. Háromdimenziós vektort egy számhármassal írunk le  $(v_1, v_2, v_3)$ , melynek minden  $v_i$  komponense skálár.



2.1. ábra. Vektor - irányított szakasz

A vektor, mint irányított szakasz, abban különbözik a szakaszoktól, hogy valójában két pont kapcsolatát írja le. A vektorok elemi geometriai használata mellett (például: eltolás jellemzése) felmerül az ötlet, hogy vektorokat feleltessünk meg pontoknak. Minden pontot egy ponthoz viszonyítunk, az irányított szakaszok kezdőpontját közösnek választva. Ennek segítségével a már megismert koordináta-rendszerbeli problémákat egy új szempögből vizsgálhatjuk meg. Az irányított szakaszokat a végpontjuk koordinátaival jellemezzük.

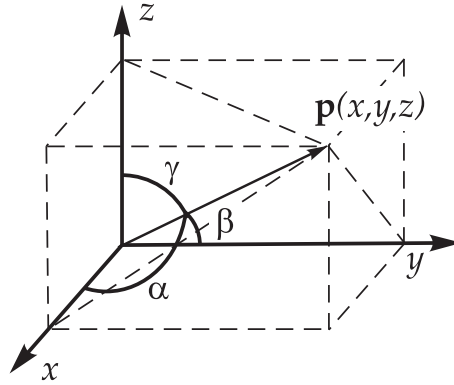
**Bázisvektorok** olyan speciális vektorok, melyek lineáris kombinációjával felírhatjuk a tér (sík) bármely vektorát. A bázisvektorok vagy ortogonálisak, ekkor páronként merőlegesek egymásra, normáltak, azaz a vektorok egységnyi hosszúak, vagy teljesítik mindkét előző tulajdonságot, akkor ortonormáltak.

## 2.1. Helyvektorok a térben

[93] Megtehetjük, hogy pontokat egy adott pontból kiinduló vektorokkal határozzunk meg. Ehhez egy vonatkozási pontot kell rögzíteni, általában ez megegyezik a koordináta-rendszer origójával. Az origóból induló vektorokat helyvektoroknak nevezzük. A helyvektorok és a sík (tér) pontjai között egyértelmű megfeleltetés van. Valamint a sík (tér) bármely vektorának meg tudunk feleltetni egy vele egyenlő helyvektort. Egy helyvektort végpontjával, illetve végpontjának koordinátaival adhatunk meg. Így a helyvektorok kölcsönösen megfeleltethetők a sík pontjainak.

A vektorokat három adat jellemzi: az irány, az irányítás és a hossz (abszolútérték):

- Minden  $P(x, y, z)$  pontot egyértelműen megadhatunk egy  $\mathbf{p} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$  helyvektorral, amelynek jelölése:  $\mathbf{p}(x, y, z)$ . Az  $\mathbf{i}$ ,  $\mathbf{j}$ ,  $\mathbf{k}$  alapvektorok (bázisvektorok) rendre a koordináta-rendszer  $x$ -,  $y$ -,  $z$ - tengelyei irányába mutató egységvektorok.
- A vektor irányát a vektor irányszögével, vagy annak valamely szögfüggvényével adhatjuk meg. **Irányszögeknek** a vektor és az egyes tengelyek által bezárt szögeket nevezzük. Ezek koszinuszai az **iránykoszinuszok**:  $l = \cos \alpha$ ,  $m = \cos \beta$ ,  $n = \cos \gamma$ , ahol teljesül az  $l^2 + m^2 + n^2 = 1$  egyenlőség. Az  $l_1$ ,  $m_1$ ,  $n_1$  illetve  $l_2$ ,  $m_2$ ,  $n_2$  iránykoszinuszokhoz tartozó vektorok egymással bezárt szögére érvényes a  $\cos \delta = l_1 l_2 + m_1 m_2 + n_1 n_2$  összefüggés. Ha  $l_1 l_2 + m_1 m_2 + n_1 n_2 = 0$ , akkor a két irány merőleges egymásra.
- Bármely vektor **hosszát**, **abszolútértékét** megkapjuk, ha a koordiná-



2.2. ábra. A vektor irányszöge

táinak négyzetösszegéből négyzetgyököt vonunk:

$$|\mathbf{p}| = \sqrt{x^2 + y^2 + z^2}$$

## 2.2. Műveletek vektorokkal

### 2.2.1. Vektorok normalizálása (*Normalization*)

Vektor normalizálása esetén a vektor iránya nem változik, viszont a hossza egy lesz. A normalizált vektort úgy kaphatjuk meg, hogy az eredeti vektort elosztjuk a hosszával:

$$\mathbf{a}' = \frac{\mathbf{a}}{|\mathbf{a}|}$$

ahol

$$|\mathbf{a}'| = 1$$

Az  $\mathbf{a}(a_1, a_2, a_3)$  helyvektor esetén:

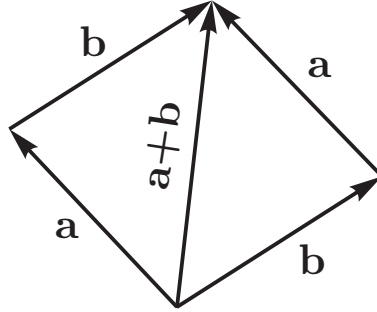
$$\mathbf{a}' = \frac{\mathbf{a}}{\sqrt{a_1^2 + a_2^2 + a_3^2}}$$

### 2.2.2. Vektorok összeadása (*Addition*)

Adott két vektor. Az egyik vektor végpontjából indítjuk a másik vektort. Az első kezdőpontjából a második végpontjába mutató vektort a két vektor



összegvektorának nevezzük. Több vektor összeadása esetén először két vektort összegzünk, majd az összeghez adjuk hozzá a következő összeadandót.



2.3. ábra. Vektorok összege

Tulajdonságai:

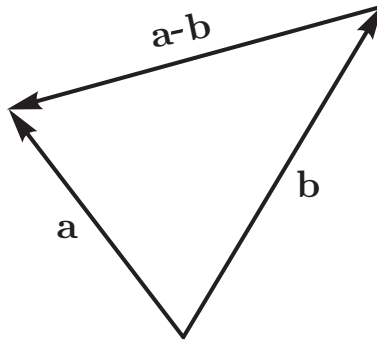
- Kommutatív:  $\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}$
- Asszociatív:  $(\mathbf{a} + \mathbf{b}) + \mathbf{c} = \mathbf{a} + (\mathbf{b} + \mathbf{c})$
- Bármely vektorhoz a nullvektort hozzáadva visszkapjuk az eredeti vektort:  $\mathbf{a} + \mathbf{0} = \mathbf{a}$
- Egy  $\mathbf{a}$  vektorhoz megadható olyan  $-\mathbf{a}$  vektor, hogy a két vektor összege nullvektor. Az ilyen módon megadott vektort az eredeti vektor **ellentettjének** nevezzük:  $\mathbf{a} + (-\mathbf{a}) = \mathbf{0}$

Helyvektorok esetén az összeadásvektor koordinátáit az egyes vektorok megfelelő koordinátáinak összege adja. Azaz  $\mathbf{a}(a_1, a_2, a_3)$  és  $\mathbf{b}(b_1, b_2, b_3)$  helyvektorok esetén:

$$\mathbf{a} + \mathbf{b} = (a_1 + b_1)\mathbf{i} + (a_2 + b_2)\mathbf{j} + (a_3 + b_3)\mathbf{k}$$

### 2.2.3. Két vektor különbsége (*Subtraction*)

Az  $\mathbf{a} - \mathbf{b}$  különbségvektorán az  $\mathbf{a} + (-\mathbf{b})$  összegvektort értjük, azaz azt a vektort, amelyet úgy kapunk, hogy  $\mathbf{a}$ -hoz hozzáadjuk  $\mathbf{b}$  ellentettjét. Ha az



2.4. ábra. Vektorok különbsége

$\mathbf{a}$  és  $\mathbf{b}$  vektorokat egy pontból indítjuk ki, akkor a  $\mathbf{b}$  végpontjából az  $\mathbf{a}$  végpontjába mutató vektor az  $\mathbf{a} - \mathbf{b}$ .

Tulajdonságai:

- A kivonás nem kommutatív, és nem asszociatív művelet.
- Ha nullvektort vonunk ki egy  $\mathbf{a}$  vektorból, visszakapjuk az  $\mathbf{a}$  vektort:  
 $\mathbf{a} - \mathbf{0} = \mathbf{a}$
- Ha a nullvektorból vonjuk ki az  $\mathbf{a}$  vektort, akkor az  $\mathbf{a}$  vektor ellentettjéhez jutunk:  $\mathbf{0} - \mathbf{a} = -\mathbf{a}$
- Ha egy vektorból önmagát vonjuk ki a művelet  $\mathbf{0}$  vektort eredményez:  
 $\mathbf{a} - \mathbf{a} = \mathbf{0}$

Helyvektorok különbségének koordinátáit az egyes vektorok megfelelő koordinátáinak különbsége adja. Azaz  $\mathbf{a}(a_1, a_2, a_3)$  és  $\mathbf{b}(b_1, b_2, b_3)$  helyvektorok esetén:

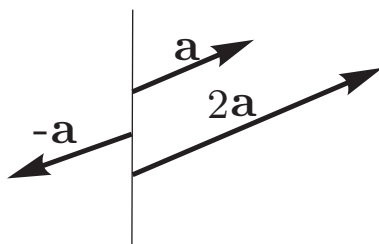
$$\mathbf{a} - \mathbf{b} = (a_1 - b_1)\mathbf{i} + (a_2 - b_2)\mathbf{j} + (a_3 - b_3)\mathbf{k}$$

#### 2.2.4. Vektor szorzása számmal (skalárral) (*Scalar multiplication*)

Adott egy  $\mathbf{a}$  vektor és egy  $\lambda$  szám. Az  $\mathbf{a}$  vektor  $\lambda$  számszorosán a következő vektort értjük:

- Ha  $\mathbf{a} = \mathbf{0}$  vagy  $\lambda = 0$ , akkor  $\lambda \mathbf{a} = \mathbf{0}$
- Ha  $\mathbf{a} \neq \mathbf{0}$  és  $\lambda \neq 0$ , akkor  $|\lambda| \cdot |\mathbf{a}|$  hosszúságú vektort kapunk, melynek iránya:
  - $\lambda > 0$  esetén  $\mathbf{a}$ -val megegyező,
  - $\lambda < 0$  esetén  $\mathbf{a}$ -val ellentétes.

Tehát egy vektornak egy  $\lambda$  számmal való szorzata a vektor hosszának növekedését vagy csökkenését vonja maga után.



2.5. ábra. Vektor szorzása

Tulajdonságai:

- $\alpha \mathbf{a} + \beta \mathbf{a} = (\alpha + \beta) \mathbf{a}$
- $\alpha(\beta \mathbf{a}) = \beta(\alpha \mathbf{a}) = \alpha\beta \mathbf{a}$
- $\alpha(\mathbf{a} + \mathbf{b}) = \alpha \mathbf{a} + \alpha \mathbf{b}$
- Bármely vektor 0-val vett szorzata nullvektort eredményez:  $0 \cdot \mathbf{a} = \mathbf{0}$ .
- A nullvektornak bármely számmal vett szorzata nullvektor:  $\alpha \cdot \mathbf{0} = \mathbf{0}$ .

Egy helyvektor skalárszorosának koordinátáit a vektor koordinátáinak adott skalárral való szorzásával kapjuk. Azaz  $\mathbf{a}(a_1, a_2, a_3)$  helyvektor esetén:

$$\lambda \mathbf{a} = \lambda a_1 \mathbf{i} + \lambda a_2 \mathbf{j} + \lambda a_3 \mathbf{k}$$

Adottak  $\mathbf{a}$  és  $\mathbf{b}$  vektorok, valamint az  $\lambda_1$  és  $\lambda_2$  valós számok, akkor a velük képzett  $\mathbf{v} = \lambda_1 \mathbf{a} + \lambda_2 \mathbf{b}$  vektort az  $\mathbf{a}$  és  $\mathbf{b}$  vektor *lineáris kombinációjának*

nevezzük, ha  $\lambda_1 + \lambda_2 = 1$ , akkor *konvex lineáris kombinációról* beszélünk. Pl. a szakasz tetszőleges pontjába mutató helyvektor a szakasz kezdő és végpontjába mutató vektorok konvex lineáris kombinációja.

### 2.2.5. Két vektor skaláris szorzata (*Dot product*)

Euklideszi térben adott  $\mathbf{a}$  és  $\mathbf{b}$  vektorok skaláris szorzata a két vektor hosszának és az általuk közbezárt szög koszinuszának szorzata. Azaz:

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| \cdot |\mathbf{b}| \cdot \cos \gamma, \text{ ahol } 0^\circ \leq \gamma \leq 180^\circ$$

Tehát a skaláris szorzat egy valós szám.

A művelet tulajdonságai:

- Kommutatív:  $\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a}$
- Nem asszociatív:  $\mathbf{a} \cdot (\underbrace{\mathbf{b} \cdot \mathbf{c}}_{\text{skalár}}) \neq (\underbrace{\mathbf{a} \cdot \mathbf{b}}_{\text{skalár}}) \cdot \mathbf{c}$ , mivel a szorzás bal oldala az  $\mathbf{a}$  vektor egy számszorosa, míg a jobb oldal a  $\mathbf{c}$  vektoré.
- Számmal való szorzásra asszociatív:  $\lambda \cdot (\mathbf{a} \cdot \mathbf{b}) = (\lambda \cdot \mathbf{a}) \cdot \mathbf{b} = \mathbf{a} \cdot (\lambda \cdot \mathbf{b})$
- Összeadásra nézve disztributív  $(\mathbf{a} + \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot \mathbf{c} + \mathbf{b} \cdot \mathbf{c}$
- Ha  $|\mathbf{a}| = 1$  és  $|\mathbf{b}| = 1$ , akkor  $\mathbf{a} \cdot \mathbf{b} = \cos \gamma$ , azaz a skaláris szorzat egységvektorok esetén megegyezik a közbezárt szög koszinuszával.

Két vektor skaláris szorzata akkor és csak akkor 0, ha a két vektor merőleges egymásra. Tehát ha merőlegesek, akkor skaláris szorzatuk nulla, ha pedig a skaláris szorzatuk nulla, akkor merőlegesek.

A derékszögű Descartes-féle koordináta rendszerben a párhuzamos egységvektorok skaláris szorzatainak értéke egység, míg az egymásra merőleges egységvektorok skaláris szorzatai nulla értéket ad:

$$\begin{aligned} \mathbf{i} \cdot \mathbf{i} &= 1, \mathbf{j} \cdot \mathbf{j} = 1, \mathbf{k} \cdot \mathbf{k} = 1 \\ \mathbf{i} \cdot \mathbf{j} &= 0, \mathbf{i} \cdot \mathbf{k} = 0, \mathbf{j} \cdot \mathbf{k} = 0 \end{aligned}$$

A vektorok skaláris szorzatának eredménye a koordináta komponensekkel is megadható, ha figyelembe vesszük az egységvektorok skaláris szorzataira vonatkozó összefüggéseket. Így az  $\mathbf{a}(a_1, a_2, a_3)$  és  $\mathbf{b}(b_1, b_2, b_3)$  helyvektorok skaláris szorzata a következő alakban írható:

$$\mathbf{a}\mathbf{b} = a_1b_1 + a_2b_2 + a_3b_3$$

Az irodalomban a skaláris szorzatra a zárójeles ill. az alsópont jelölést is szokás használni:

$$(\mathbf{a}, \mathbf{b}), \quad \mathbf{a} \cdot \mathbf{b}$$

### 2.2.6. Példák skaláris szorzás használatára

[105] *Két vektor hajlásszögének kiszámítása:* A skaláris szorzás legáltalánosabb használata két vektor hajlásszögének a meghatározása. Hajlásszöget számolunk például árnyékolásnál, vagy láthatósági tesztelésnél. Árnyékolásnál a fény irányvektora és a felületi normális szögét számítjuk ki, míg láthatósági vizsgálatnál a nézeti irány, illetve a felületi normális által bezárt szög szükséges.

Használjuk a két vektor különbségére a koszinusz-tételt. Ebből azt kapjuk, hogy:

$$|\mathbf{a} - \mathbf{b}|^2 = |\mathbf{a}|^2 + |\mathbf{b}|^2 - 2|\mathbf{a}||\mathbf{b}|\cos\gamma$$

ahol  $\gamma$  a két vektor által bezárt szög.

Valamint a négyzetre emelést elvégezve teljesül, hogy

$$|\mathbf{a} - \mathbf{b}|^2 = |\mathbf{a}|^2 + |\mathbf{b}|^2 - 2 \cdot \mathbf{a} \cdot \mathbf{b},$$

ezért fennáll az

$$|\mathbf{a}||\mathbf{b}|\cos\gamma = \mathbf{a} \cdot \mathbf{b},$$

egyenlőség, amiből átrendezéssel az alábbi összefüggést kapjuk:

$$\cos \gamma = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} = \frac{a_1 b_1 + a_2 b_2 + a_3 b_3}{\sqrt{a_1^2 + a_2^2 + a_3^2} \sqrt{b_1^2 + b_2^2 + b_3^2}}$$

Ami pontosan azt jelenti, hogy két vektor hajlásszögének koszinuszát megkaphatjuk a normalizált vektoraik skalárszorzatával.

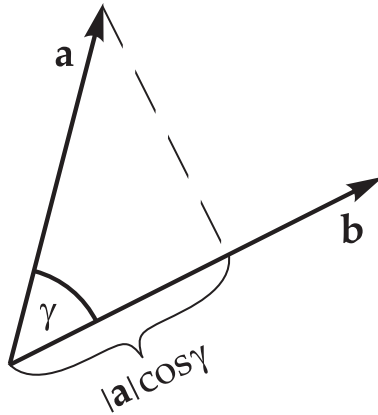
*Vektornak egy másik vektorra történő merőleges vetítése:* A skaláris szorzat másik alkalmazása vektornak egy másik vektorra történő vetületének megadása. Adott egy  $\mathbf{b}$  egységvektor. Az  $\mathbf{a}$  vektornak szeretnénk a  $\mathbf{b}$  vektorra történő vetületét megadni, az eredményvektort jelöljük  $\mathbf{a}'$ -vel. Ekkor azt kapjuk, hogy

$$\mathbf{a}' = |\mathbf{a}| \cos \gamma = |\mathbf{a}| \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} = \mathbf{a} \cdot \mathbf{b},$$

mivel

$$|\mathbf{b}| = 1$$

Tehát az  $\mathbf{a}$  és  $\mathbf{b}$  vektorok skaláris szorzata megegyezik az  $\mathbf{a}$  vektornak a  $\mathbf{b}$  vektorra vonatkozó vetületének hosszával.



2.6. ábra. Az  $\mathbf{a}$  vektor merőleges vetülete a  $\mathbf{b}$  vektorra

*A skaláris szorzat előjele:* A közebezárt szög jellemzésére szolgál a skaláris szorzat előjele. Két vektor skaláris szorzatának előjelét a közebezárt szögük

koszinusza határozza meg

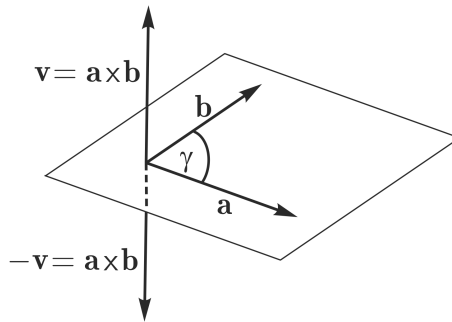
$\mathbf{a} \cdot \mathbf{b} > 0$ , ha  $\gamma < 90^\circ$ , azaz hegyesszög,

$\mathbf{a} \cdot \mathbf{b} = 0$ , ha  $\gamma = 90^\circ$ , azaz derékszög,

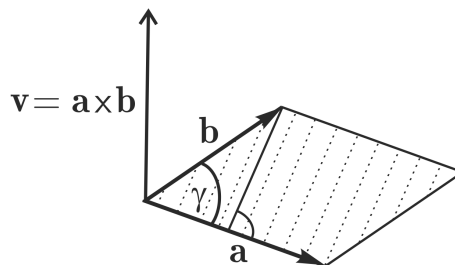
$\mathbf{a} \cdot \mathbf{b} < 0$ , ha  $\gamma > 90^\circ$ , azaz tompaszög.

### 2.2.7. Két vektor vektoriális szorzata (*Cross product*)

Két vektor vektoriális szorzata vektort eredményez. Az  $\mathbf{a}$  és  $\mathbf{b}$  vektorok  $\mathbf{a} \times \mathbf{b}$  vektoriális szorzatának azt a  $\mathbf{v}$  vektort tekintjük, amelynek az  $|\mathbf{a}| |\mathbf{b}| \sin \gamma$  hosszúsága az  $\mathbf{a}$  és a  $\mathbf{b}$  vektorok által kifeszített paralelogramma területével egyenlő, iránya merőleges mind az  $\mathbf{a}$ , mind a  $\mathbf{b}$  vektorra, olyan irányítással, hogy az  $\mathbf{a}$ ,  $\mathbf{b}$  és  $\mathbf{v}$  vektorok jobbsodrású hármast alkotnak. Ezért vektoriális szorzattal lehet a legegyszerűbben két vektorra merőleges vektort megadni.



2.7. ábra. Vektoriális szorzat



2.8. ábra. A vektoriális szorzat hossza:  $|\mathbf{v}| = |\mathbf{a}| |\mathbf{b}| \sin \gamma$

Tulajdonságai:

– Antikommutatív:  $\mathbf{a} \times \mathbf{b} = -(\mathbf{b} \times \mathbf{a})$

– Az összeadásra nézve disztributív:

$$\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c}$$

$$(\mathbf{b} + \mathbf{c}) \times \mathbf{a} = \mathbf{b} \times \mathbf{a} + \mathbf{c} \times \mathbf{a}$$

Mivel a vektoriális szorzat nem kommutatív, ezért mindkét alak felírására szükség van.

– Skalárral való szorzás:  $(\lambda \mathbf{a}) \times \mathbf{b} = \mathbf{a} \times (\lambda \mathbf{b}) = \lambda(\mathbf{a} \times \mathbf{b})$

– Nem asszociatív:  $(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} \neq \mathbf{a} \times (\mathbf{b} \times \mathbf{c})$

– Teljesíti a Jacobi azonosságot:  $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) + \mathbf{b} \times (\mathbf{c} \times \mathbf{a}) + \mathbf{c} \times (\mathbf{a} \times \mathbf{b}) = \mathbf{0}$

– Az  $\mathbf{a}$  vektornak az  $\mathbf{e}$  egységvektorra merőleges összetevője előáll  $\mathbf{a}_m = (\mathbf{e} \times \mathbf{a}) \times \mathbf{e}$  alakban.

Két vektor vektoriális szorzata akkor és csak akkor nullvektor, ha a két vektor párhuzamos.

A derékszögű, Descartes-féle koordináta-rendszerben a párhuzamos egységvektorok vektoriális szorzatai nulla hosszúságú vektorokat eredményeznek, míg az egymásra merőleges egységvektorok vektoriális szorzatai mindkét vektorra merőleges, egységnyi hosszúságú, egységvektorokat adnak.

$$\mathbf{i} \times \mathbf{i} = 0, \mathbf{j} \times \mathbf{j} = 0, \mathbf{k} \times \mathbf{k} = 0$$

$$\mathbf{i} \times \mathbf{j} = \mathbf{k}, \mathbf{j} \times \mathbf{k} = \mathbf{i}, \mathbf{k} \times \mathbf{i} = \mathbf{j}$$

Az  $\mathbf{a}(a_1, a_2, a_3)$  és  $\mathbf{b}(b_1, b_2, b_3)$  vektorok vektoriális szorzatát a legegyszerűbben a következő mátrix determinánsának kiszámításával (Sarrus-szabály) kapjuk:

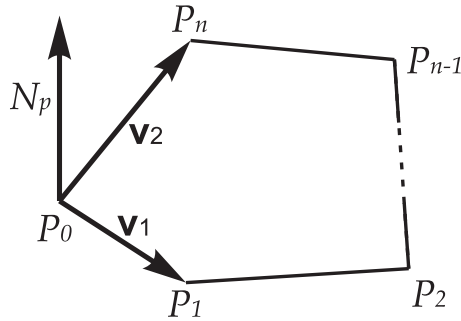
$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = (a_2 b_3 - b_2 a_3) \mathbf{i} + (b_1 a_3 - a_1 b_3) \mathbf{j} + (a_1 b_2 - b_1 a_2) \mathbf{k}$$



## 2.2.8. Példák vektoriális szorzat használatára

[105] *Poligon felületi normálisa:* A grafikában kitüntetett szerepe van azoknak a vektoroknak, melyek merőlegesek egy felületre. Egy testet határoló **poligon** (sokszögvonala) **normálvektora** a poligon három pontja segítségével számítható ki. Három pont két vektort  $\mathbf{v}_1$  és  $\mathbf{v}_2$  definiál, melyek vektoriális szorzatából előállítható a felületi normálisa a következő alakban:

$$N_p = \mathbf{v}_1 \times \mathbf{v}_2$$



2.9. ábra. Poligonlap normálvektora

Amikor meghatározzuk egy adott poligon felületi normálisát, akkor a vektoriális szorzatnak a poligont tartalmazó test szempontjából kifelé kell mutatnia. Egy jobbsodrású rendszerben a vektoriális szorzat követi a jobbkézsabályt. Ha a hüvelykujj és a mutatóujj a  $\mathbf{v}_1$ , illetve  $\mathbf{v}_2$  vektor irányába mutat, akkor a  $N_p$  iránya megegyezik a középsőujj irányával.

*Görbe felületi normálisa:* Amennyiben a felszín egy **másodrendű paraméteres** (bicubic parametric) **görbe**, akkor a normálvektor iránya folyamatosan változik a felületen. Valamelyik  $(u, v)$  pontban kiszámítjuk a felületi normálist a vektoriális szorzás segítségével. Ehhez szükséges ebben a pontban az érintősíkot kifesztő két érintővektor. Amennyiben a felszínt a  $\mathbf{Q}(u, v)$  függvény állítja elő, akkor a két érintővektor a parciális deriváltak segítségével áll elő

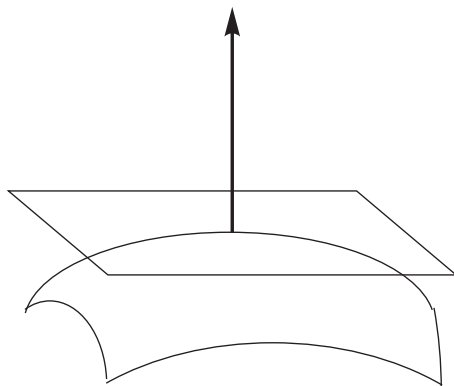
$$\frac{\partial}{\partial u} \mathbf{Q}(u, v)$$

illetve

$$\frac{\partial}{\partial v} \mathbf{Q}(u, v)$$

alakban. Tehát a felületi normálist ezen két parciális derivált vektoriális szorzataként definiálhatjuk:

$$N_p = \frac{\partial \mathbf{Q}}{\partial u} \times \frac{\partial \mathbf{Q}}{\partial v}$$



2.10. ábra. Felületi normális

### 3. Mátrixok (*Matrices*)

[92] A matematikában – de általában az életben is – gyakoriak az olyan rendszerek, amelynek "jellemzéséhez" több szám kell. (Például egy tetraéder jellemezhető az éleinek hosszával, egy elektromos hálózat a csomópontok közötti részek ellenállásának nagyságával, stb.)

Ezekben az esetekben természetesen nem csak az számít, hogy milyen adatok szerepelnek, hanem az is, hogy ezek az adatok egymáshoz képest hogyan helyezkednek el. Ezzel lehet meghatározni, hogy egy-egy adat mit jellemez. Ilyen esetekben a szóban forgó alakzathoz tartozó mátrixról beszélünk.

A mátrixoknak különböző alakjuk lehet annak megfelelően, hogy a benne szereplő számokat hogyan helyeztük el. A leggyakrabban téglalap alakú mátrixokat használunk.

Mátrixaknak tekintjük tehát adatoknak téglalap alakban való elrendezését. E téglalapokról feltesszük, hogy véges sok soruk és véges sok oszlopuk van. A mátrixban szereplő adatokat a mátrix elemeinek nevezzük. Ezek az adatok általában számok, de más adatok is előfordulhatnak, például függvények, vagy újabb mátrixok. Az elemeket azzal határozzuk meg, hogy egy-egy elemről megmondjuk, melyik sorban és melyik oszlopban áll. Az itt lévő sor- és oszlopszámot az elemhez írjuk kettős indexként. Az  $m_{ij}$  valamely mátrixnak az  $i$ -edik sorában és  $j$ -edik oszlopában lévő elemet jelöli. Tekintsünk egy általános  $n \times m$ -es mátrixot, amelynek  $n$  sora és  $m$  oszlopa van:

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1m} \\ m_{21} & m_{22} & & m_{2m} \\ \vdots & & \ddots & \vdots \\ m_{n1} & m_{n2} & \dots & m_{nm} \end{bmatrix}$$

A mátrixokat rendszerint zárójelbe tesszük, jelölve hogy a táblázatot egyetlen egységként kezeljük. Az matematikai irodalomban általában kerek zárójelet használnak, ebben a könyvben a komputergrafikában szokásos szögletes zárójeles jelölést alkalmazzuk.

Tehát a mátrixot azzal tudjuk meghatározni, ha megadjuk a sorainak és oszlopainak számát, valamint azt, hogy egy-egy előírt helyen milyen szám áll. Ez a szám tehát egy helynek a függvénye. A hely pedig nem más, mint egy számpár. Így a mátrixot egy olyan függvénynek tekintjük, amely egy természetes számokból álló számpárhoz egy számot rendel hozzá. A számpárban szereplő első számnak a sorok számánál, a második számnak pedig az oszlopok számánál kell kisebb vagy egyenlőnek lennie. Vegyük figyelembe, hogy számos programnyelvben nullától kezdődik a számozás, ekkor érteleszerűen 1-et le kell vonni minden indexelésből. Speciálisan egyetlen sort vagy egyetlen oszlopot, sőt egyetlen elemet is mátrixnak tekinthetünk.

Két mátrixot akkor tekintünk egyenlőnek, ha ugyanannyi soruk és oszlopuk van, és a megfelelő helyeken álló elemek megegyeznek. Viszont nem kell megkövetelnünk az értékészletek megegyezését, vagyis, ha például egy valós elemű mátrix minden eleme racionális, akkor nem tekintjük különbözőnek attól a mátrixtól, amelyik "ugyanaz", csak éppen racionális elemű mátrixnak értelmeztük.

### 3.1. Nevezetes mátrixok

- Ha a mátrixnak egy sora vagy egy oszlopa van, azaz  $1 \times n$ -es vagy  $m \times 1$ -es, akkor **sor-**, illetve **oszlopvektornak** nevezzük.
- Amennyiben a mátrix sorainak száma megegyezik az oszlopainak számával, a mátrix **négyzetes**.
- A **diagonálmátrix** olyan négyzetes mátrix, melynek csak főátlójában vannak 0-tól eltérő elemek.
- Az **egységmátrix** olyan diagonálmátrix, melynek főátlóbeli elemei egységek, jele: **E**.
- **Nullmátrix** minden eleme 0.
- Egy mátrix **transzponáltja** a sorok és oszlopok felcserélésével nyert mátrix.

Pl:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}, \quad \begin{bmatrix} a & b & c \end{bmatrix}^T = \begin{bmatrix} a \\ b \\ c \end{bmatrix},$$

továbbá érvényesek az alábbi szabályok:

$$(\mathbf{A}^T)^T = \mathbf{A} \quad (\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$$

- Egy  $\mathbf{A}$  négyzetes mátrixnak akkor létezik  $\mathbf{A}^{-1}$  inverze, ha az  $\mathbf{A}^{-1}$  mátrix kielégíti az  $\mathbf{A}^{-1} \mathbf{A} = \mathbf{E}$  és  $\mathbf{AA}^{-1} = \mathbf{E}$  egyenleteket, ekkor  $\mathbf{A}^{-1}$  az inverzmátrix. Egy mátrixnak csak abban az esetben lehet inverze, ha a determinánsa nem nulla,  $|\mathbf{A}| \neq 0$ . Érvényesek az alábbi szabályok:

$$(\mathbf{A}^{-1})^{-1} = \mathbf{A} \quad (\mathbf{AB})^{-1} = \mathbf{B}^{-1} \mathbf{A}^{-1} \quad (\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T.$$

- $2 \times 2$  -es mátrix inverze:

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$
$$\mathbf{A}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

- Az **ortogonális** mátrix olyan négyzetes mátrix, melynek transzponáltja egyben az inverze is. A térben minden tengely körüli elforgatás olyan ortogonális mátrixszal írható fel, melynek determinánsa 1. A tengely körüli elforgatás és síkra vonatkozó tükrözés szorzata pedig -1 determinánsú ortogonális mátrix.

### 3.2. Mátrix determinánsa (*Determinant*)

A determináns fogalmával mindenütt lehet találkozni, ahol négyzetes (vagy quadratikus) mátrix szerepel, azaz olyan mátrixok, amelyeknél a sorok szá-

ma megegyezik az oszlopok számával. A determináns nem más, mint egy négyzetes mátrixhoz rendelt szám.

A determináns kiszámolható kifejtéssel. az egyelemű mátrix determinán-sa megegyezik az elem értékével. A másodrendű determinánst a következőképp lehet kiszámítani:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

Harmadrendű determinánst a Sarrus-féle szabállyal számítjuk ki (lásd [http://en.wikipedia.org/wiki/Rule\\_of\\_Sarrus](http://en.wikipedia.org/wiki/Rule_of_Sarrus)).

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} + \\ -a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12}$$

A magasabb rendű determináns értékét úgy kapjuk meg, hogy valamely sorának elemeit megszorozzuk a hozzájuk tartozó előjeles aldeterminánsokkal, és ezeket a szorzatokat összeadjuk.

$$\det \mathbf{A} = \sum_{j=1}^n (-1)^{j+k} a_{jk} \det \mathbf{A}_{jk}$$

Bármely  $n$  természetes szám esetén az  $n$ -ed rendű determinánsokra igazak az alábbiak:

- Ha a mátrix főátlója fölött (alatt) csupa 0 áll, akkor a determináns értéke a főátlóban álló elemek szorzata. Speciálisan, ha a fődiagonális minden eleme 1, és a többi elem 0, akkor a determináns értéke 1 lesz.
- Ha a mátrix valamely sorának vagy oszlopának minden eleme 0, akkor a determináns értéke is 0.
- Ha a mátrix egy sorát (vagy oszlopát) egy  $c$  valós számmal megszo-

rozzuk, akkor a determináns értéke is  $c$ -szeresére változik.

- Ha a négyzetes mátrix sorait permutáljuk, akkor páros permutálás esetén nem változik a determináns, páratlan permutálás esetén előjelet vált. Speciálisan, ha a determináns két sorát felcseréljük, az értéke  $(-1)$ -szeresére változik.
- Ha egy mátrix két sora (vagy két oszlopa) megegyezik, akkor a determináns értéke 0.
- A mátrix transzponáltjának a determinánsa megegyezik az eredeti mátrix determinánsával.
- A determináns értéke nem változik, ha a mátrix egyik sorához (oszlopához) hozzáadjuk valamely másik sor (oszlop) számszorosát.

A mátrix determinánsa *Gauss-eliminációval* is kiszámolható. Az eljárás lényege, hogy a sorokon vagy oszlopokon végrehajtott lineáris transzformációt addig kell folytatni, míg felső háromszögmátrixhoz nem jutunk. Ekkor a determináns értékét a főátlóbeli elemek szorzataként kapjuk. A Gauss-elimináció lépései során a determináns értéke nem változik meg.

A Gauss-elimináció lépései a következők:

- Keresünk egy nem nulla elemet, és sor- vagy oszlopcserevel a főátlóba tesszük. Mivel a páratlan számú cserék megváltoztatják a determináns előjelét, ezért a cserék számát feljegyezzük.
- Az elem alatt lévő elemeket kinullázzuk. Mindezt két lépéssel lehet megoldani, a mátrix sorának számmal szorzásával, és egyik sorhoz a másik sor skalárszorosának hozzáadásával.
- Az előző két lépést kell ismételni úgy, hogy a kiválasztott elemet mindig másik sorból és másik oszlopból választjuk.
- Ha két sor vagy oszlop megegyezik, illetve ha egy sor vagy oszlop minden eleme 0, akkor nincs szükség további átalakításokra, mert a determináns értéke 0.

- Különbösen felső háromszögmátrixot kapunk, melynek determinánsa a főátlóban álló elemek szorzata.

A mátrix inverze pedig *Gauss-Jordan eliminációval* határozható meg. Az algoritmus lényege az, hogy a mátrix mellé felírjuk a vele azonos méretű egységmátrixot, majd addig használjuk a sorokon vagy oszlopokon végrehajtott lineáris transzfomációt, amíg az eredeti mátrix helyén egységmátrix keletkezik. Ekkor az egységmátrix helyén kapjuk meg az inverzmátrixot. Numerikus algoritmusok megoldására ajánljuk a "Numerical Recipes: The Art of Scientific Computing" irodalmat (lásd [106]), ahol programkódokat is közölnek a szerzők.

### 3.3. Mátrixműveletek (*Matrix Operations*)

A mátrixokat a számok egy általánosításaiként tekinthetjük. Pontosabban szólva, az egyelemű mátrixokat majdnem azonosnak tekinthetjük a számokkal. Tekintettel arra, hogy számokkal különböző műveleteket lehet elvégezni, ezért lehetőség van, hogy a műveleteket általában mátrixokra is kiterjesszük. E kiterjesztésnél természetesen vigyázni kell arra, hogy speciális esetben – vagyis egyelemű mátrixokra – a művelet ugyanaz legyen, mint számokra. Két eltérés tapasztalható csak a számműveletek és a mátrixműveletek között, ugyanis a mátrixműveletek között nem szerepel az osztás, és nincs szükség a szorzás kommutativitására sem.

#### 3.3.1. Mátrixok összeadása (*Matrix addition and subtraction*)

Az  $\mathbf{A}$  és  $\mathbf{B}$  mátrix összeadása akkor végezhető el, ha alakjuk megegyezik, azaz mindkettő  $m \times n$ -es. Ennek megfelelően az eredménymátrix alakja is  $m \times n$ -es lesz, és az  $\mathbf{A} + \mathbf{B}$  mátrix  $i$ -edik sorában és  $j$ -edik oszlopában lévő elem az  $\mathbf{A}$ , illetve  $\mathbf{B}$  mátrix  $i$ -edik sorában és  $j$ -edik oszlopában lévő  $a_{ij}$  és  $b_{ij}$  elemek összege. Például:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 5 \\ 7 & 2 & 9 \\ 3 & 8 & 6 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 8 \\ 11 & 7 & 15 \\ 10 & 16 & 15 \end{bmatrix}$$



Tulajdonságai:

- Kommutatív:  $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$
- Asszociatív:  $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$
- Ha egy  $\mathbf{0}$  nullmátrix ugyanolyan típusú, mint az  $\mathbf{A}$  mátrix, akkor teljesül:  $\mathbf{A} + \mathbf{0} = \mathbf{A}$ .
- Minden  $\mathbf{A}$  mátrixhoz található olyan  $\mathbf{B}$  mátrix, melyre teljesül:  $\mathbf{A} + \mathbf{B} = \mathbf{0}$ .
- Tetszőleges  $\mathbf{A}$  és  $\mathbf{B}$  mátrixokhoz létezik pontosan egy olyan  $\mathbf{X}$  mátrix, amelyre  $\mathbf{X} + \mathbf{A} = \mathbf{B}$ . Ezt az  $\mathbf{X}$  mátrixot a  $\mathbf{B} - \mathbf{A}$  jelöli. E mátrix előállítását értelmezhetjük a mátrixokon végzett kivonás műveleteként.

### 3.3.2. Mátrix szorzása számmal (*Matrix skalar multiplication*)

Egy  $\mathbf{A}$  mátrix  $r$  valós számmal való szorzatán azt a mátrixot értjük, melyet  $\mathbf{A}$ -ból úgy kapunk, hogy  $\mathbf{A}$  minden elemét megszorozzuk  $r$ -rel. Például:

$$2 \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix}$$

Tulajdonságai:

- $r(\mathbf{A} + \mathbf{B}) = r\mathbf{A} + r\mathbf{B}$
- $(r + s)\mathbf{A} = r\mathbf{A} + s\mathbf{A}$
- $r(s\mathbf{A}) = (rs)\mathbf{A}$
- $1\mathbf{A} = \mathbf{A}$
- $0\mathbf{A} = \mathbf{0}$
- $p(\mathbf{A}\mathbf{B}) = \mathbf{A}(p\mathbf{B})$

Az utolsó tulajdonság esetében értelmeznünk kell a mátrix közötti szorzást.

### 3.3.3. Mátrixok szorzása (*Matrix multiplication*)

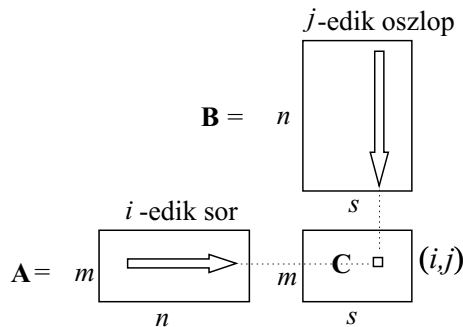
**A** és **B** mátrixok között a szorzás művelete csak akkor értelmezhető, ha **A**-nak ugyanannyi oszlópa van, mint ahány sora **B**-nek. Amennyiben **A**  $m \times n$ -es és **B**  $n \times s$ -es mátrix, akkor a **C** szorzat mátrix egy  $m \times s$ -es típusú mátrix lesz. A szorzat mátrix  $i$ -edik sorának és  $j$ -edik oszlopának találkozásánál lévő  $c_{ij}$  elemet úgy kapjuk meg, hogy az első mátrix  $i$ -edik sorának minden elemét összeszorozzuk a második mátrix  $j$ -edik oszlopának minden elemével, majd a szorzatokat összeadjuk. A definíciót megfogalmazhatjuk a skaláris szorzat segítségével is. A  $c_{ij}$  elem az **A** mátrix  $i$ -edik sorvektorának és a **B** mátrix  $j$ -edik oszlopvektorának a skaláris szorzata, melyet a következő képlet alapján kapunk meg:

$$c_{ij} := a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj} = \sum_{k=1}^n a_{ik}b_{kj},$$

ezt kell elvégeznünk minden elemen, azaz

$$i \in 1 \dots m, j \in 1 \dots s$$

tehát három egymásba ágyazott ciklussal lehet megoldani a feladatot (lásd programozási melléklet)



3.1. ábra. Az  $m \times s$ -es **C** szorzatmátrix  $c_{ij}$  elemének kiszámítása

Például:

$$\begin{aligned} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 7 & 2 \\ 3 & 8 \end{pmatrix} &= \begin{pmatrix} 1 \cdot 1 + 2 \cdot 7 + 3 \cdot 3 & 1 \cdot 0 + 2 \cdot 2 + 3 \cdot 8 \\ 4 \cdot 1 + 5 \cdot 7 + 6 \cdot 3 & 4 \cdot 0 + 5 \cdot 2 + 6 \cdot 8 \end{pmatrix} = \\ &= \begin{pmatrix} 24 & 28 \\ 57 & 58 \end{pmatrix} \end{aligned}$$

A művelet tulajdonságai:

- Asszociatív:  $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$
- Mindkét oldalról disztibutív:
  - $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$
  - $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$
- Skalárral való szorzás:  $k(\mathbf{AB}) = (k\mathbf{A})\mathbf{B} = \mathbf{A}(k\mathbf{B})$
- Egy  $n \times n$ -es  $\mathbf{A}$  mátrixot akár balról, akár jobbról szorzunk egy  $n \times n$ -es  $\mathbf{E}$  egységmátrix-szal, egyaránt az eredeti  $\mathbf{A}$  mátrixot kapjuk eredményül.  $\mathbf{EA} = \mathbf{A}$ , ill.  $\mathbf{AE} = \mathbf{A}$ .
- A mátrixok szorzása nem kommutatív:  $\mathbf{AB} \neq \mathbf{BA}$ . Gyakran az is előfordul, hogy az  $\mathbf{A}$  mátrix szorozható a  $\mathbf{B}$ -vel, de a tényezők felcserélésével már nem értelmezhető a művelet.
- Mátrixok szorzatának inverzére és transzponáltjára érvényesek a következő szabályok:

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$$

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1} \mathbf{A}^{-1}$$

Megfigyelhetjük, hogy a transzponáltakat illetve az inverz mátrixokat fordított sorrendben kell összeszorozni. Mivel a szorzás nem kommutatív, ezért fontos ügyelnünk a sorrendre.

## 4. Koordináta-rendszerek (*Coordinate system*)

[102] Feladatunk a különböző tárgyak geometriai sajátosságainak leírása. Ehhez valamilyen megállapodásra van szükségünk, ilyen a vonatkozási pont és a lépték. Ezeket a megállapodásokat összességében nevezzük koordináta-rendszernek. Ha kiválasztunk egy koordináta-rendszert, a tárgy bármely pontjának térbeli helyzetét néhány számérték segítségével meg tudjuk adni.

Azt, hogy melyik koordináta-rendszert érdemes alkalmazni az ábrázolandó alakzat határozza meg. Természetesen bármely koordináta-rendszerről át lehet térni egy másikra a megfelelő képletek segítségével. Ez a megfeleltetés kölcsönösen egyértelmű.

Bár a koordináta-rendszerek megválasztásához a lehetőségek száma végtelen, a gyakorlatban mégis csak néhány típust használunk.

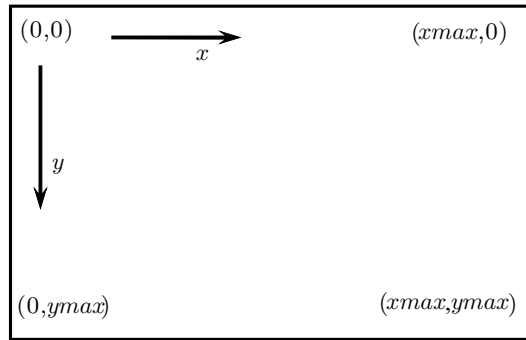
### 4.1. Descartes-féle koordináta-rendszer (*Cartesian coordinate system*)

A leggyakrabban alkalmazott koordináta-rendszer a Descartes-féle. A következőkben a kétdimenziós és háromdimenziós esetet ismertetjük.

#### 4.1.1. Kétdimenziós Descartes-féle koordináta-rendszer

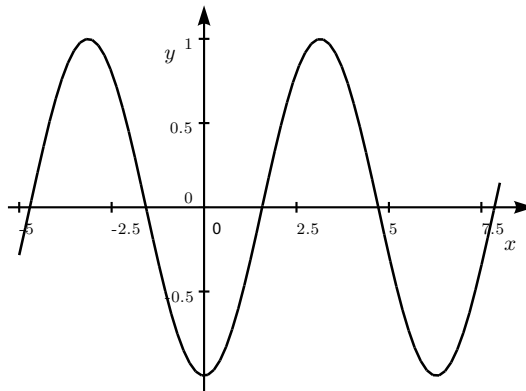
A kétdimenziós eset jól ismert, itt csak arra térünk ki, hogy a komputergrafikában az origó helyzete általában a bal felső sarok, s az  $y$  tengely lefelé növekszik.

Speciális esetekben, pl. a teknőcgrafikában előfordul, hogy a koordináta-rendszer kiindulópontja a képernyő közepén található, s az  $y$  tengelyirány is fölfelé növekszik. A komputergrafikában használt algoritmusok, így az általunk megadott raszteres algoritmusok is ebben a hagyományos rendszerben készülnek, mintha a matematika füzetünkben dolgoznánk. Ez nem okoz nehézséget, mert egy egyszerű ponttranszformációval áttérhetünk a 4.1 ábrán bemutatott, egyébként a monitorok kezelésének szokásos rendszerére. Azaz  $x$  tengelyre való tükrözésre, az origó közepre való eltolására és általában skálázásra (nagyításra) van szükség. Amikor erről megfeledezünk, akkor



4.1. ábra. Origó a bal felső sarokban

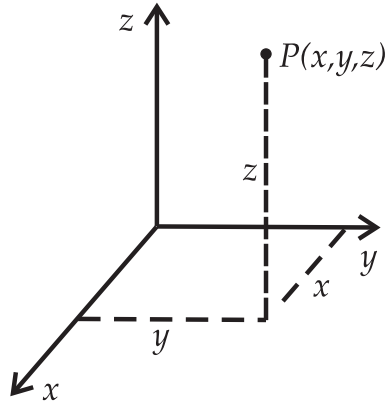
születnek olyan tipikus hibák, mint amikor a  $\cos(x)$  függvényt ábrázoljuk hibásan (lásd 4.2 ábra).



4.2. ábra. Hibás  $\cos(x)$  ábrázolás!

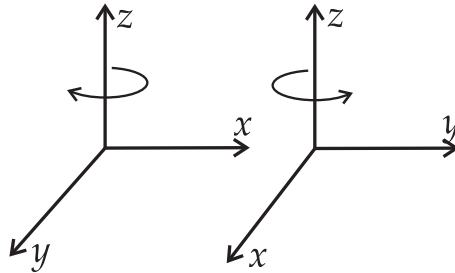
#### 4.1.2. Háromdimenziós Descartes-féle koordináta-rendszer

A Descartes-féle koordináta-rendszerben a három tengely egymásra merőleges, és egy pont helyzetét – koordinátáit – az  $x$ ,  $y$  és  $z$  tengelyektől mért távolsága határozza meg. A tengelyek metszéspontja az origó. Az origóból kiinduló, és a tengelyek irányába mutató, egymásra kölcsönösen merőleges egységvektorokat nevezzük alapvektoroknak vagy bázisvektoroknak. A  $P(x, y, z)$  pontba mutató helyvektor felírható a tengelyeken vett egységvektorok (bázisvektorok) lineáris kombinációjaként:  $\mathbf{p} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ .



4.3. ábra. Descartes-féle koordináta-rendszer

A tárgyakat általában jobbsodrású rendszerben ábrázoljuk. Szemléletesen ez azt jelenti, ha az  $x$ ,  $y$ ,  $z$  jobb kezünk hüvelyk, mutató, és középső ujjunkkal forgatással fedésbe hozható. Ellenkező esetben balsodrású koordináta-rendszerrel beszélünk. Más szemlélet szerint egy koordináta-rendszer akkor jobbsodrású, ha annak egy  $(\mathbf{i}, \mathbf{j}, \mathbf{k})$  ortogonális bázisával az  $(\mathbf{i} \times \mathbf{j}) \cdot \mathbf{k}$  vegyes szorzat pozitív, negatív esetben balsodrású a rendszer.



4.4. ábra. Bal- ill. jobbsodrású rendszer

A grafikai programok közül a RenderMan alapértelmezetten balsodrású, míg az OpenGL jobbsodrású koordináta-rendszert használ.

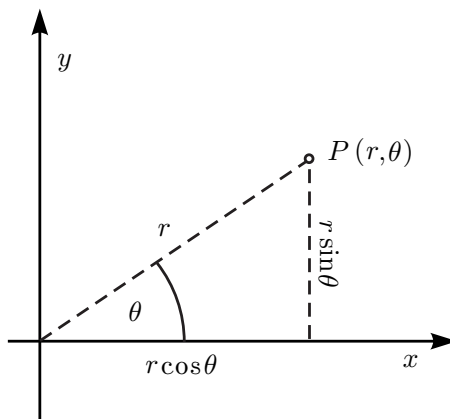
Ha az alapvektoroknál nem tesszük kötelezővé sem az egységnyi hosszúságot, sem azt, hogy derékszöveget zárjanak be egymással, akkor a koordináta-rendszer affin (ferdeszögű).

## 4.2. Görbevonalú koordináta-rendszer

[88] Gyakran nem lehet, vagy nem célszerű a Descartes-féle koordináta-rendszert használni. Előfordulhat, hogy a koordináta-rendszer az alkat minden pontjában más és más, és a koordinátatengelyek állása is pontonként változik az adott feladat jellegéhez illeszkedően. Ha például egy gömb alakú térrésszel kapcsolatos feladatot kell megoldanunk, akkor általában célszerű olyan pontonként változó koordináta-rendszert alkalmaznunk, amely "hozzásimul" a gömb alakjához. Ekkor ugyanis a gömb egyenlete ebben a koordináta-rendszerben egyszerűbb lesz, és ez megkönnyíti a számítást.

### 4.2.1. Polárkoordináta-rendszer (*Polar coordinate system*)

[94] A sík pontjait (síkbeli) polárkoordinátákkal is jellemezhetjük. Ezt a koordináta-rendszert az  $O$  kezdőpontja (az origó) és egy ebből kiinduló  $L$  félegyenes definiál, melynek irányát kezdőiránynak nevezzük. A  $P$  pont polárkoordinátái az  $r = OP$  távolság, valamint a kezdőirány és az  $OP$  irány által meghatározott  $\theta$  irányított szög. Mivel  $\theta$  mértéke  $360^\circ$  bármely egész többszörösével megváltoztatható, ezért minden ponthoz többféle koordináta-pár adható meg. Magának az  $O$  kezdőpontnak a  $\theta$  koordinátája tetszőleges lehet.



4.5. ábra. Polárkoordináta-rendszer

Polárkoordináta-rendszerben egy  $P$  pont helyét két adattal adhatjuk meg:

$(r, \theta)$ , ahol

- $r$  a sugár ( $0 \leq r$ ), azaz a pontnak az origótól való távolsága,
- $\theta$  a szög ( $0^\circ \leq \theta < 360^\circ$ ) pedig az  $L$  félegyenes és a sugár által bezárt szög.

Ha Descartes koordináta-rendszerben polárkoordinátákkal kell megadni adatokat, akkor általában origóként a  $(0, 0)$  pontot  $L$  félegyenesként pedig az  $x$  tengely pozitív részét (az origótól jobbra eső részt) választjuk. Ekkor az  $r, \theta$  polárkoordinátájú pont közösleges koordinátái:

$$\begin{aligned}x &= r \cos \theta \\y &= r \sin \theta\end{aligned}$$

Így például az  $x^2 + y^2 = 1$  egységkör poláregyenlete

$$(r \cos \theta)^2 + (r \sin \theta)^2 = 1,$$

vagyis egyszerűen

$$r = 1.$$

A polárkoordináták különösen alkalmasak az olyan mozgások és hasonlósági leképezések leírására, amelyeknek van invariáns pontjuk. Ekkor ugyanis ezt a pontot kezdőpontnak választva az  $(r, \theta)$  általános helyzetű pontot

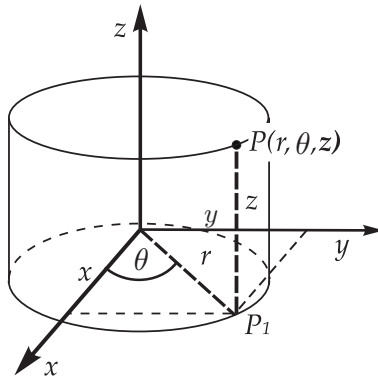
- az  $\alpha$  szöggel forgatás az  $(r, \theta + \alpha)$ ,
- félfordulat az  $(r, \theta + \pi)$ ,
- kezdőegyenesre tükrözés az  $(r, -\theta)$ ,
- $O\mu$  nyújtás  $(\mu r, \theta)$ ,
- egy  $O$  középpontú nyújtva forgatás a  $(\mu r, \theta + \alpha)$ ,

pontba transzformál.



#### 4.2.2. Hengerkoordináta-rendszer (*Cylindrical coordinates*)

[94] A hengerkoordináta-rendszer a polárkoordináta-rendszer térbeli általánosítása a magasság bevezetésével a  $z$  tengely irányában. A tér pontjait hengerkoordinátákkal is meg tudjuk adni. Egy alapsíkot (az  $x, y$  tengelyek által meghatározott sík), egy rá merőleges irányított egyenest (a  $z$  tengely), az alapsíkon belül pedig az origóból induló félegyenest (a  $x$  tengely pozitív félegyenese) veszünk fel. Az irányított egyenes mind a vele párhuzamos egyenesek, mind az alapsík irányítását is megszabja. Az irányított alapsíkban a megadott félegyenes egy polárkoordináta-rendszert határoz meg. A  $P$  pont polárkoordinátái az alapsíkra vetett  $P_1$  vetületének  $r, \theta$  polárkoordinátái, valamint az irányított  $P_1P$  szakasz előjeles  $z$  hossza.



4.6. ábra. Hengerkoordináta-rendszer

Ezért egy  $P$  pontot három koordinátája  $(r, \theta, z)$  definiál:

- $r$  a tengelytől mért merőleges távolság;
- $\theta$  a  $P$  pont és a tengely által meghatározott sík hajlásszöge a  $\theta = 0$  ( $x-z$ ) síkhoz.  $\theta$ -t azimuth szögnek is nevezik (lásd gömbkoordináták);
- $z$  a  $P$  pont és a pont merőleges vetületének,  $P_1$ -nek a távolsága;

Sajnálatos módon az irodalomban sok eltérő jelölést használnak, mivel mi a polárkoordinátáknál is  $\theta$ -val jelöltük a fenti szöveget, ezért használjuk továbbra is így. Ellenben pl. Arfken (lásd [87])  $(\rho, \phi, z)$  jelölést használja.

Több javaslatot is olvashatunk (lásd [103]) az egységesítésre, de elég ha figyelmesen olvassuk az irodalmat, hogy ne keverjük össze a jelöléseket.

[88] Az összes olyan pont, amelynek egyik hengerkoordinátája rögzített érték, ún. *koordinátafelületet* határoz meg. Így az  $r = r_0$  egyenletű koordinátafelületek a  $z$  tengely körüli végtelen hosszú körhengerek, a  $\theta = \theta_0$  koordinátafelületek a  $z$  tengelyre illeszkedő félsíkok, míg a  $z = z_0$  koordinátafelületek a  $z$  tengelyre merőleges síkok.

Ha azokat a pontokat keressük, amelyeknek hengerkoordinátája rögzített, akkor az ún. *koordinátagörbéket* kapjuk. Így  $r = r_0, \theta = \theta_0$  esetén  $z$  tengellyel párhuzamos egyeneseket;  $r = r_0, z = z_0$  esetén  $(x, y)$  síkkal párhuzamos,  $z$  tengely körüli körvonalakat; míg  $\theta = \theta_0, z = z_0$  esetén  $(x, y)$  síkkal párhuzamos, a  $z$  tengelytől induló félegyeneseket kapunk.

Ha az alapsíkbeli polárkoordináta-rendszerhez derékszögű  $xy$  koordináta-rendszert illesztünk, és irányított egyenesünket  $z$  tengelyül választjuk, akkor a közöséges koordinátákat az  $r, \theta, z$  polárkoordinátákból a következőképp számítjuk ki:

$$\begin{aligned}x &= r \cos \theta \\y &= r \sin \theta \\z &= z\end{aligned}$$

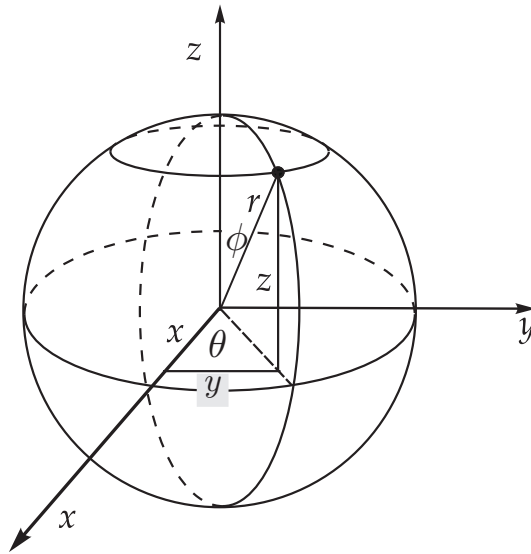
A Descartes-féle koordinátákból a hengerkoordinátákat a következő összefüggés alapján kapjuk meg:

$$\begin{aligned}r &= \sqrt{x^2 + y^2} \\ \theta &= \arctan\left(\frac{y}{x}\right) = \arcsin\left(\frac{y}{r}\right) \\ z &= z\end{aligned}$$

#### 4.2.3. Gömbi koordináta-rendszer (*Spherical coordinates*)

[94] A gömbi koordinátákat gömbi polárkoordinátáknak is nevezik. Most egy félsíkot (a  $z$  tengely, illetve az  $x$  pozitív fele által meghatározott félsík) és ennek határán egy origó kezdőpontú félegyenest ( $z$  tengely pozitív fele) ve-

szünk fel. A  $P$  pont első koordinátája az  $r = OP$  távolság. A második koordináta az a  $\theta$  irányított szög, amelyet az adott félsík, valamint a vele közös határu, a  $P$  pontot tartalmazó félsík határoz meg. Ennek az irányított szögnek a mérésénél azt a forgásirányt választjuk pozitívnak, amely az adott félegyenes irányából nézve pozitívnak látszik. Végül a harmadik koordináta az adott és az  $OP$  félegyenes  $\phi$  hajlásszöge. Eszerint  $\phi$  konvex szög;  $\theta$  minden pontra több értékű, mégpedig  $360^\circ$  bármely egész többszörösével megváltoztatható; az  $O$  kezdőpont  $\theta, \phi$  koordinátái pedig tetszőlegesek lehetnek.

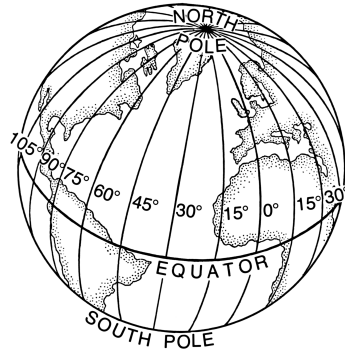


4.7. ábra. Gömbi koordináta-rendszer

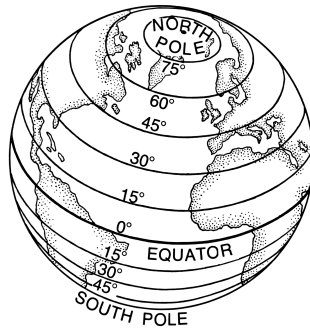
Egy  $P$  pontot három koordináta  $(r, \theta, \phi)$  határoz meg, ahol

- $r$  az origótól mért távolság, azaz a gömb sugara;
- $\theta$  a  $P$  pont és a  $\phi = 0$  tengely által meghatározott sík, valamint  $\theta = 0$  sík közötti hajlásszög az  $x - y$  koordinátasíkban,  $\theta$ -t azimut szögnek is nevezik;
- $\phi$  a pontot és az origót összekötő egyenes hajlásszöge a  $\phi = 0$  irányhoz, azaz a  $z$  tengellyel bezárt szög.

A  $\theta$ -t a földrajzi hálózatban hosszúsági foknak (longitude) nevezik, továbbá  $\phi = 90^\circ - \delta$ , ahol  $\delta$ -t pedig szélességi foknak (latitude) hívják. A következő magyarázó ábrán a hosszúsági majd a szélességi köröket látjuk. (forrás: Pearson Scott Foresman képei, <http://www.wikipedia.org>).



4.8. ábra. Hosszúsági körök



4.9. ábra. Szélességi körök

Azért töltünk ilyen sok időt a fogalmak tisztázással, mert a szakirodalomban sajnos többféle jelölést is használnak a gömbkoordinátákra. Pl. Arfken (lásd [87])  $(r, \phi, \theta)$  jelölést használja, azaz felcseréli a görög ábécé betűit a szögek jelölésében, ezért legyünk óvatosak az irodalom olvasásában.

Ha olyan jobbsodrású koordináta-rendszert veszünk fel, amelynek kezdőpontja  $O$ ,  $z$ -tengelye az adott félegyenes irányába mutat,  $x$ -tengelyének

pozitív fele pedig az adott félsíkban halad, akkor az  $r, \theta, \phi$  koordinátájú  $P$  pont közös koordinátái:

$$\begin{aligned}x &= r \sin \phi \cos \theta \\y &= r \sin \phi \sin \theta \\z &= r \cos \phi\end{aligned}$$

A Descartes-féle koordináták ismeretében a következőképp kapjuk meg a gömbi koordinátákat:

$$\begin{aligned}r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \arctan\left(\frac{y}{x}\right) \\ \phi &= \arccos\left(\frac{z}{r}\right)\end{aligned}$$

A gömbi polárkoordináták alkalmazása olyankor lehetnek hasznos, amikor gömböket tartalmazó problémával van dolgunk. Jól használhatók például tárgyak egy ponthoz viszonyított térbeli helyzetének és mozgásának leírásakor. Ugyancsak jól használhatók a földgömbön való, és ahhoz képesti helyzet megadásakor. Komputergrafikában a kamera vagy a fényforrás pozíciójának a megadására is elterjedt a gömbkoordináták használata, mivel így könnyedén tudjuk forgatni a kamerát vagy a fényforrást az objektum körül. Képzeljük el a kamerát egy gömb felületén, ahol a térbeli objektum a gömb középpontjában van. A kamera a gömb középpontjába néz. Ha körbe akarjuk forgatni az objektumot, akkor erre nincs szükség, hiszen a kamera mozgásával a szélességi és hosszúsági körök mentén ugyanazt a hatást érzük el. Ezen a módon a felhasználó számára sokkal jobban érthető és használható modellt alkotunk, mintha az objektumot forgatnánk a koordináta tengelyek körül.

## 5. Homogén koordináták

Komputergrafikában igen elterjedt a homogén koordináták használata. Nagyon sok irodalomban és leírásban csak formális, az egységes tárgyalási módot lehetővé tevő eszközként említik, mint például a ponttranszformációk esetében.

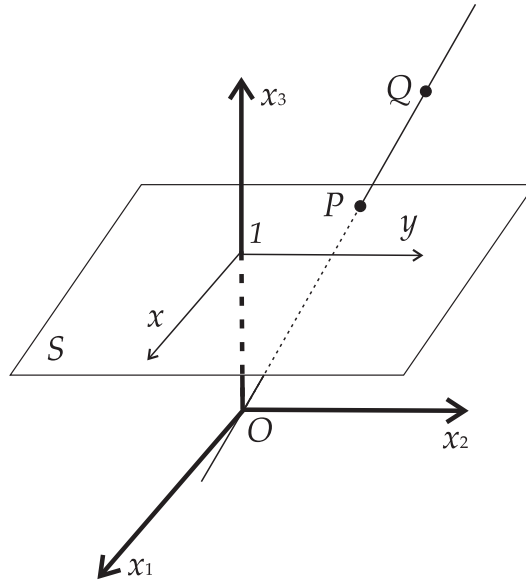
A homogén koordináták fogalma alapkonstrukció a projektív geometriában (lásd [29]). Számos komputergrafikai megoldás, algoritmus a projektív geometriai ismereteken alapszik. Ebben a könyvben csak fel tudjuk hívni a figyelmet ezen ismeretek fontosságára. Például a homogén koordináták használata elengedhetetlen amikor a térbeli objektumokat a képernyőn síkjában kívánjuk megjelelni, azaz centrális vetítést kell alkalmaznunk. Ha 3D megjelenítés kívánunk alkalmazni, akkor is két centrális vetítést alkalmazunk, szimulálva a két szemünk által elvégzett leképezést.

Több ezer éves probléma a párhuzamosság kérdése. A sík két egyenese vagy párhuzamos, vagy pedig egy közös pontjuk van. Ha nem szeretnénk a párhuzamosságot külön kezelni, bővítsük ki az egyenes pontjait egy **ideális pont**nak nevezett elemmel. Erre a végtelen távoli pontra teljesülnie kell a következőnek: két egyenes pontjainak az összességét akkor és csak akkor bővítjük ugyanazzal az új elemmel, ha párhuzamosak. A nem ideális pontokat megkülönböztetésül közönséges pontoknak nevezzük. Az ideális pontok egy egyenesen helyezkednek el, az ideális egyenesen. A térben szintén elvégezhető az ideális elemekkel történő bővítés. A tér valamennyi ideális pontja az ideális síkot határozza meg.

### 5.1. Síkbeli homogén koordináták

Az ideális pontokkal való bővítést a **homogén koordináták** bevezetésével oldjuk meg (lásd[48]). A sík pontjainak homogén koordinátás, három összetevős alakját egy háromdimenziós Descartes-féle koordináta-rendszerben szokás szemléltetni. Legyen adott egy  $z = 1$  sík, amely tehát párhuzamos az  $x - y$  koordináta síkkal. Definiáljunk ezen sík pontjai és a koordináta-rendszer kezdőpontján áthaladó egyenesek között egy kölcsönösen egyértelmű kapcsot

latot, úgy, hogy az egyenesekhez a síkkal alkotott metszéspontjukat (dőfés-pontjukat) rendeljük hozzá, a sík tetszőleges pontjához pedig az origóval összekötő egyenest. Az  $x - y$  koordinátasíkban fekvő egyeneseknek a  $z = 1$  sík végtelen távoli pontjai felelnek meg (lásd ábra).



5.1. ábra. Síkbeli homogén koordináták

Az origón áthaladó egyenesek egyértelműen reprezentálhatók irányvektorokkal, melyek hossza tetszőleges, tehát a  $\mathbf{v}(x, y, z)$  és  $\lambda\mathbf{v}(\lambda x, \lambda y, \lambda z)$  (ahol  $\lambda \neq 0$  valós szám) ugyanazt az egyenest, azaz ugyanazt a  $z = 1$  síkra illeszkedő pontot reprezentálja. Ilyen módon a  $z = 1$  sík pontjait olyan rendezett számhármassokkal reprezentáljuk, amelyek arányosság erejéig vannak meghatározva és mindhárom egyszerre nulla. Ezeket, a koordinátákat nevezik homogén koordinátáknak.

**5.1. definíció.** Síkbeli homogén koordináták: A sík pontjait olyan rendezett számhármassokkal reprezentáljuk, amelyek arányosság erejéig vannak meghatározva, és mind a három egyszerre nem lehet nulla.

A definíció értelmezése:

- rendezett számhármass:  $[x_1, x_2, x_3]$ ,

- arányosság: az  $[x_1, x_2, x_3]$  ugyanazt a pontot jelöli, mint a  $[\lambda x_1, \lambda x_2, \lambda x_3]$ , ahol  $\lambda$  egy 0-tól különböző valós szám, Pl:  $[1, -2, 2]$  ugyanaz a pontot jelöli, mint a  $[2, -4, 4]$ ,
- $[0, 0, 0]$  homogén koordinátájú pont nem létezik.

**Áttérés hagyományos Descartes koordinátákról homogén koordinátákra.** Legyen egy síkbeli pont hagyományos valós koordinátája  $[x, y]$ , a homogén koordinátás alak  $[x, y, 1]$  lesz. Tehát az  $x_1 = x, x_2 = y, x_3 = 1$  megfeleltetést használjuk. Mivel a homogén koordináták csak arányosság erejéig vannak meghatározva, ezért most már szorozhatjuk a koordinátákat, egy tetszőleges nem nulla számmal.

**Visszatérés homogén koordinátákról Descartes koordinátákra.** Ha csak affin transzformációkkal foglalkozunk, akkor a harmadik koordináta 1 lesz, ezért a visszatérésnél egyszerűen elhagyjuk. Általánosan ha egy pont homogén koordinátája  $[x_1, x_2, x_3]$ , és  $x_3$  nem nulla, akkor az első két koordinátát eloszthatjuk a definícióban foglalt arányossági tulajdonság miatt a harmadik koordinátával:

$$\left[ \frac{x_1}{x_3}, \frac{x_2}{x_3}, 1 \right].$$

Ebben az esetben láthatjuk, hogy valójában az  $x = \frac{x_1}{x_3}$  és az  $y = \frac{x_2}{x_3}$  megfeleltetést használtuk. Ha  $x_3 = 0$ , akkor nincs hagyományos valós megfelelője a pontnak, ez csak a nem affin transzformációknál fordul elő.

## 5.2. Két pontra illeszkedő egyenes egyenlete

Amennyiben szeretnénk meghatározni az  $S$  sík egy  $e$  egyenesét, ezt a legegyszerűbben a térbeli koordináta-rendszer  $O$  kezdőpontján és az  $e$  egyenesen átfektetett sík megadásával tehetjük meg. Ezt a síkot az  $\mathbf{e} \neq 0$  normálvektorával jellemezhetjük. A következőkben az egyenest meghatározó vektorok esetén ilyen normálvektorokra kell gondolnunk.

Vegyük az  $e$  egyenes két pontját  $A(a_1, a_2, 1)$ -t és  $B(b_1, b_2, 1)$ -t, melyek normált homogén koordinátás alakban vannak. Az ezen pontokba mutató



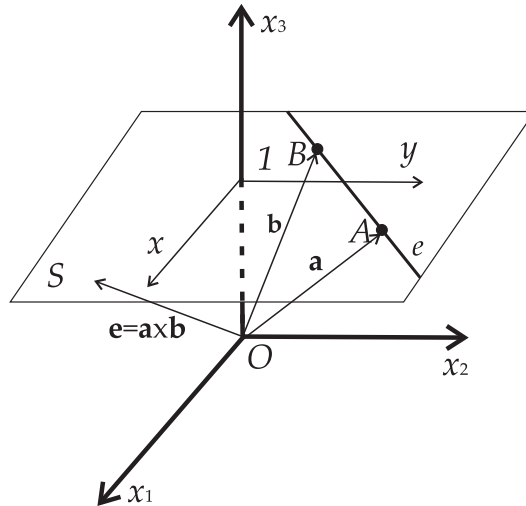
helyvektorok vektoriális szorzataként meg tudjuk határozni az  $e$  egyenes normálvektorát. Azaz:

$$\mathbf{e} = \mathbf{a} \times \mathbf{b}$$

Ennek alapján kiszámíthatók az  $e$  vektor koordinátái:

$$\mathbf{e} = \begin{pmatrix} a & b & c \end{pmatrix} = \begin{pmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \end{pmatrix} = \begin{pmatrix} a_2 - b_2 & b_1 - a_1 & a_1 b_2 - b_1 a_2 \end{pmatrix} \quad (5.1)$$

Mivel az  $\mathbf{a}$  és  $\mathbf{b}$  vektorok az  $S$  sík  $e$  egyenesének és a térbeli koordináta-rendszer  $O$  kezdőpontja által meghatározott síkjában vannak, a vektoriális szorzatuk pontosan merőleges lesz mind az  $S$  síkra, mind az  $e$  egyenesre.



5.2. ábra. Két pontra illeszkedő egyenes.

Legyen az  $e$  egyenes egyenlete  $ax + by + c = 0$  alakú. Szorozzuk meg az egyenlet mindkét oldalát  $x_3$ -mal, így a homogén koordináta definíciójából az  $ax_1 + bx_2 + cx_3 = 0$ -t kapjuk. Ennek alapján az  $e$  egyenes keresett egyenlete:

$$(a_2 - b_2)x_1 + (b_1 - a_1)x_2 + (a_1 b_2 - b_1 a_2)x_3 = 0$$

Tehát a sík egyeneseit (hasonlóan a sík pontjaihoz) rendezett számhároma-

sokkal reprezentáljuk, amelyek az arányosság erejéig vannak meghatározva, és mind a három egyszerre nem lehet nulla.

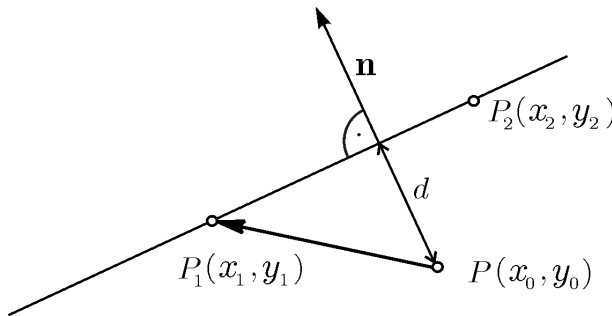
**Illeszkedés.** Az előzőekből következik, hogy egy  $\mathbf{p}$  vektor által meghatározott pont akkor és csak akkor illeszkedik az  $\mathbf{e}$  vektor által meghatározott egyenesre, ha  $\mathbf{p} \cdot \mathbf{e} = 0$ . Ez tulajdonképpen a két vektor merőlegességét jelenti.

### 5.3. Pont és egyenes távolsága

Egy pont homogén koordinátás vektora és egy  $e$  egyenes irányvektorának a skaláris szorzata a pont és az egyenes távolságával arányos. A keresett távolságot megkapjuk, ha a két vektor skaláris szorzatát elosztjuk az egyenes normálvektorának hosszával. Az  $\mathbf{e}(abc)$  vektorral mint homogén koordinátákkal reprezentált síkbeli egyenes egyenlete  $ax+by+c=0$ . Ebből az egyenes normálvektora  $\mathbf{n}(a,b)$ . Legyen a  $P$  pontba mutató helyvektor  $\mathbf{p}(x_0, y_0, 1)$ .

**Pont és egyenes távolsága a síkon.**

$$d = \frac{\mathbf{p} \cdot \mathbf{e}}{|\mathbf{n}|} = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}. \quad (5.2)$$



5.3. ábra. Pont és egyenes távolsága síkon

A fenti összefüggést megtaláljuk az alábbi linken Weisstein munkájában (lásd [84]):

<http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html>

Ha az egyenes két pontjával adott  $P_1(x_1, y_1)$  és  $P_2(x_2, y_2)$  akkor (5.1) szerint

$$\mathbf{e} = \begin{pmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix} = \begin{pmatrix} y_1 - y_2 & x_2 - x_1 & x_1 y_2 - x_2 y_1 \end{pmatrix}.$$

Ekkor (5.2) szerint

$$\begin{aligned} d &= \frac{|x_0(y_1 - y_2) + y_0(x_2 - x_1) + x_1 y_2 - x_2 y_1|}{\sqrt{(x_2 - x_1)^2 + (y_1 - y_2)^2}} = \\ &= \frac{|(x_2 - x_1)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_1 - y_2)^2}}. \end{aligned}$$

Ez utóbbi formula, azért szerencsés, mert megfelel a térbeli esetnek (lásd [85]), ahol minden vektornak van  $z$  koordinátája is.

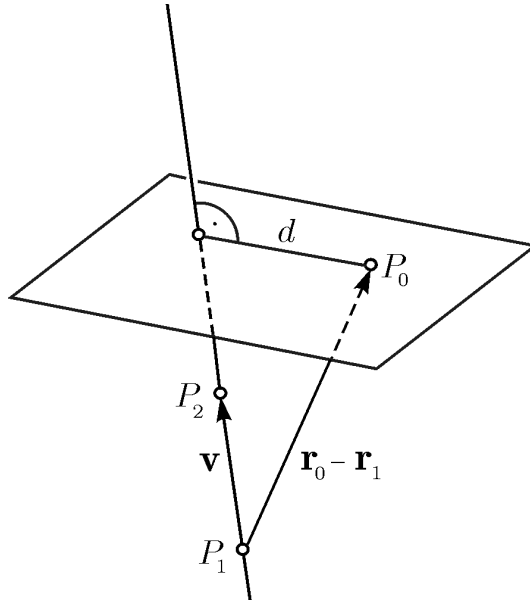
**Pont és egyenes távolsága térben.** A vektoriális szorzat definíciója alapján (lásd [76] 252. oldal):

$$d = \frac{|(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_1 - \mathbf{r}_0)|}{|\mathbf{r}_2 - \mathbf{r}_1|} = \frac{|(\mathbf{r}_0 - \mathbf{r}_1) \times (\mathbf{r}_0 - \mathbf{r}_2)|}{|\mathbf{r}_2 - \mathbf{r}_1|}.$$

#### 5.4. Két egyenes metszéspontjának meghatározása

Tekintsük az  $S$  sík  $e$  és  $f$  egyeneseit. Az egyenesek metszéspontjának koordinátáit szeretnénk meghatározni. Az  $e$  egyenest az  $A$  és  $B$  pont, az  $f$  egyenest a  $C$  és  $D$  pont határozza meg. Legyen a metszéspont  $P$ .

Mivel a  $P$  pont illeszkedik mind az  $e$ , mind az  $f$  egyenesre, így fennáll a  $\mathbf{p} \cdot \mathbf{e} = 0$ , illetve  $\mathbf{p} \cdot \mathbf{f} = 0$  egyenlőség. Az előző két egyenlet azt jelenti, hogy a  $\mathbf{p}$  vektor merőleges az  $\mathbf{e}$  illetve  $\mathbf{f}$  vektorokra. Ilyen irányú  $\mathbf{p}$  vektort a legegyszerűbben az  $\mathbf{e}$  és  $\mathbf{f}$  vektoriális szorzata segítségével állíthatunk elő,



5.4. ábra. Pont és egyenes távolsága, ahol  $\mathbf{v} = \mathbf{r}_2 - \mathbf{r}_1$

azaz  $\mathbf{p} = \mathbf{e} \times \mathbf{f}$ , ahol  $\mathbf{e}(a, b, c)$  és  $\mathbf{f}(e, f, g)$ :

$$P = \begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix} = \begin{pmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a & b & c \\ e & f & g \end{pmatrix} = \begin{pmatrix} bg - fc & ec - ag & af - eb \end{pmatrix}$$

A  $P$  pont Descartes koordinátái:

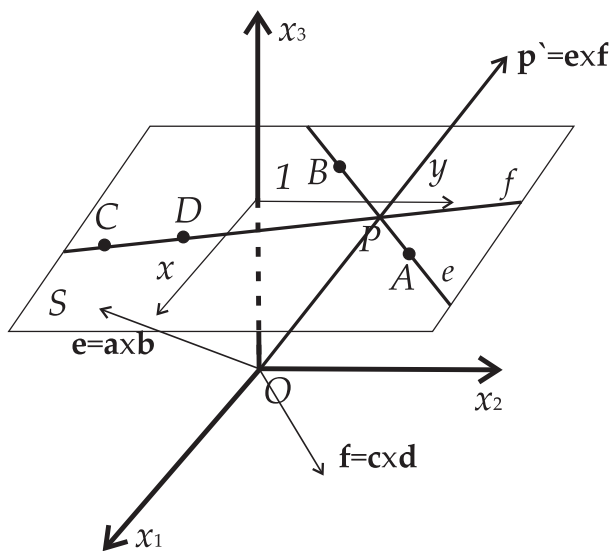
$$x = \frac{bg - fc}{af - eb}, y = \frac{ec - ag}{af - eb} .$$

## 5.5. Térbeli homogén koordináták

definíció: Térbeli homogén koordináták: A tér pontjait olyan rendezett számnégyesekkel reprezentáljuk, amelyek arányosság erejéig vannak meghatározva, és mind a négy egyszerre nem lehet nulla.

A definíció értelmezése:

– rendezett számnégyes:  $[x_1, x_2, x_3, x_4]$ ,



5.5. ábra. Két egyenes metszéspontja.

- arányosság: az  $[x_1, x_2, x_3, x_4]$  ugyanazt a pontot jelöli, mint a  $[\lambda x_1, \lambda x_2, \lambda x_3, \lambda x_4]$ , ahol  $\lambda$  egy 0-tól különböző valós szám.  
Pl:  $[1, -2, 2, 1]$  ugyanazt a pontot jelöli, mint a  $[2, -4, 4, 2]$ ,
- $[0, 0, 0, 0]$  homogén koordinátájú pont nem létezik.

**Áttérés hagyományos Descartes koordinátákról homogén koordinátákra.** Legyen egy térbeli pont hagyományos valós koordinátája  $[x, y, z]$ , a homogén koordináták alak  $[x, y, z, 1]$  lesz. Tehát az  $x_1 = x, x_2 = y, x_3 = z, x_4 = 1$  megfeleltetést használjuk. Mivel a homogén koordináták csak arányosság erejéig vannak meghatározva, ezért most már szorozhatjuk a koordinátákat, egy tetszőleges nem nulla valós számmal.

**Visszatérés homogén koordinátákról Descartes koordinátákra.** Ha csak affin transzformációkkal foglalkozunk, akkor a negyedik koordináta egy lesz, ezért a visszatérésnél egyszerűen elhagyjuk. Általánosan ha egy pont homogén koordinátája  $[x_1, x_2, x_3, x_4]$ , és  $x_4$  nem nulla, akkor az első három koordinátát eloszthatjuk a definícióban foglalt arányossági tulajdonság miatt

a negyedik koordinátával:

$$\left[ \frac{x_1}{x_4}, \frac{x_2}{x_4}, \frac{x_3}{x_4}, 1 \right].$$

Ebben az esetben láthatjuk, hogy valójában az  $x = \frac{x_1}{x_4}$  és az  $y = \frac{x_2}{x_4}$   $z = \frac{x_3}{x_4}$  megfeleltetést használtuk. Ha  $x_4 = 0$ , akkor nincs hagyományos valós megfelelője a pontnak, ez csak a nem affin transzformációknál fordul elő.

**Három ponton átmenő sík egyenletének megadása.** A síkot határozza meg három nem egy egyenesre 8 nem kollineáris) normált alakú pontja:  $A(a_1, a_2, a_3, 1)$ , a  $B(b_1, b_2, b_3, 1)$ , valamint a  $C(c_1, c_2, c_3, 1)$ . Egy sík általános egyenlete  $ax + by + cz + d = 0$ , melyből  $x_4$ -gyel való szorzás után következik az  $ax_1 + bx_2 + cx_3 + dx_4 = 0$  egyenlőség. Végezzük el az általánosan a két dimenzióban alkalmazott két ponton átmenő egyenes egyenletére alkalmazott eljárást, amiből azt kapjuk, hogy:

$$(a, b, c, d) = \begin{pmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{s} \\ a_1 & a_2 & a_3 & 1 \\ b_1 & b_2 & b_3 & 1 \\ c_1 & c_2 & c_3 & 1 \end{pmatrix} =$$

$$= \mathbf{i} \begin{vmatrix} a_2 & a_3 & 1 \\ b_2 & b_3 & 1 \\ c_2 & c_3 & 1 \end{vmatrix} - \mathbf{j} \begin{vmatrix} a_1 & a_3 & 1 \\ b_1 & b_3 & 1 \\ c_1 & c_3 & 1 \end{vmatrix} + \mathbf{k} \begin{vmatrix} a_1 & a_2 & 1 \\ b_1 & b_2 & 1 \\ c_1 & c_2 & 1 \end{vmatrix} - \mathbf{s} \begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix}$$

Tehát az  $a$ ,  $b$ ,  $c$  és  $d$  értékeket előállíthatjuk a megfelelő  $3 \times 3$ -as rész-mátrixok determinánsaként.

A homogén koordináták előnye az, hogy az ideális pontokhoz is tudunk megfelelő koordinátákat rendelni. Hátránya, hogy egyazon pont homogén koordinátákkal való meghatározása nem egyértelmű, mivel az így megadott pontok koordinátái csak arányosság erejével meghatározottak.

A módszer lényege, hogy a térbeli pontokat három koordináta helyett négy koordinátával azonosítjuk, azaz valamely  $P$  pont  $(x, y, z)$  koordinátahármasa helyett a  $(xw, yw, zw, w)$  koordinátanégyest használjuk, ahol  $w$  egy

tetszőleges, nullától különböző valós szám (skalár). Ebből az írásmódból bármikor visszatérhetünk a háromdimenziós koordinátákra, ha a vektor első három rendezőjét elosztjuk a negyedikkel. Látható, hogy egy számnegyeshöz pontosan egy pont tartozik, de egy ponthoz végtelen számnegyeshöz, melyek egymástól csak nullától különböző konstansszorzóban térnek el. A közönséges pont legegyszerűbb felírása az alábbi:  $[x_1, y_1, z_1, 1]$ , ebben az esetben a koordináta normált alakú.

Speciális esetet jelent, ha a negyedik koordináta 0, ebben az esetben ez végtelen távoli pontot jelöl, aminek nincs megfelelője az euklideszi térben. Az ideális pontok homogén alakjának összetevői közül az utolsót, a nullát elhagyva, a többiek az illető ideális pontra illeszkedő egyenesek irányvektorának összetevői. Az nem fordulhat elő, hogy mind a négy koordinátája egyszerre legyen nulla.

A transzformáció leírására szolgáló mátrixokat homogén koordinátákból építjük fel. A homogén koordináták segítségével a perspektivikus transzformációk is leírhatók. E koordináták bevezetésével minden transzformáció önálló matematikai egységként kezelhető, s egyben biztosítható az egymás után következő transzformációk összekapcsolása (konkatenációja).

Néhány nevezetes pont homogén koordinátája (tetszőleges, nullától különböző  $c$  valós számmal):

- az origó homogén alakja:  $[0, 0, 0, c]$
- az  $x$ ,  $y$ , illetve a  $z$  tengely végtelen távoli pontja:  $[c, 0, 0, 0]$ ,  $[0, c, 0, 0]$ ,  $[0, 0, c, 0]$ .

## 6. Ponttranszformációk *Linear transformation*

Ponttranszformáción olyan megfeleltetést értünk, mely az alakzat minden egyes pontjához egyértelműen hozzárendel egy pontot. Olyan eljárásoknál kell ezt alkalmazni, amelyekben a tárggyal összefüggésben annak valamilyen hasonmása keletkezik. Tipikus példa erre minden leképezési folyamat (például a fényképezés), ahol a tárgyhoz, annak egyes pontjaihoz egy kép pontjait rendeljük. Ide értjük azokat az eseteket is, amikor a tárgy deformációkat szenved. Az általunk tárgyalt legbővebb transzformáció a projektív transzformáció.

A térbeli ponttranszformációk általános alakja, ahol  $\mathbf{p}$  a transzformálandó,  $\mathbf{p}'$  a transzformált pontba mutató helyvektor,  $\mathbf{M}$  pedig a  $4 \times 4$ -es transzformációs mátrix:

$$\mathbf{p}' = \mathbf{M} \cdot \mathbf{p},$$
$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

ahol

$$|\mathbf{M}| \neq 0 \quad .$$

Először az úgynevezett elemi transzformációkat tárgyaljuk, ezekből lehet összetett transzformációkat létrehozni. Minden transzformáció esetén megadjuk az inverz transzformációt is. A leírásnál elsősorban Juhász Imre könyvére támaszkodunk (lásd [48], [96]).

### 6.1. Egybevágósági transzformáció (*Congruence transformation*)

Azokat a transzformációkat, ahol bármely szakasz képe ugyanolyan hosszúságú szakasz egybevágósági transzformációnak nevezzük. Ilyen tulajdonságú transzformáció az identitás, az elforgatás, az eltolás és a tükrözés. Az eltolást és elforgatást összefoglalóan mozgásnak szokás nevezni, ugyanis ilyen



esetben van olyan térbeli "mozgás" amivel az egybevágó alakzatok egymással "fedésbe" hozhatók. Az egybevágósági transzformációk közös jellemzője, hogy a reprezentáló mátrixok bal felső  $3 \times 3$ -as minor mátrixa ortonormált mátrix –azaz sorai és oszlopai páronként merőleges egységvektorok– és determinánsa mozgás esetén 1, tükrözésnél pedig -1.

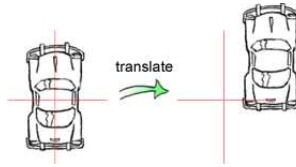
### 6.1.1. Identitás (*Identity*)

Az identitás esetén minden ponthoz önmagát rendeljük hozzá, tehát a pontok koordinátái nem változnak. Itt a tárgy mozdulatlan.

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

### 6.1.2. Eltolás (*Translation*)

Az eltolás esetén a  $P$  ponthoz megadott irányban és távolságban levő  $P'$  pontot rendeljük hozzá. Ez esetben az eltolást egy vektorral adhatjuk meg. Legyen a vektor koordinátája  $\mathbf{d}(dx, dy, dz)$ .



6.1. ábra. Síkbeli eltolás.[108]

A  $P'$  pont koordinátáját megkapjuk, ha a  $P$  pont megfelelő koordinátáihoz hozzáadjuk a  $\mathbf{d}$  vektor megfelelő koordinátáit:

$$\begin{aligned} \mathbf{p}' &= \mathbf{p} + \mathbf{d}, \\ x' &= x + dx \\ y' &= y + dy \\ z' &= z + dz. \end{aligned}$$

Ugyanez homogén koordinátás alakban és mátrix szorzással felírva:

$$\mathbf{p}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} dx \\ dy \\ dz \\ 0 \end{bmatrix},$$

$$\mathbf{p}' = \mathbf{M} \cdot \mathbf{p}$$

adódik az eltolás  $\mathbf{M}$  mátrixa illetve inverze:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{M}^{-1}(dx, dy, dz) = \mathbf{M}(-dx, -dy, -dz) = \begin{bmatrix} 1 & 0 & 0 & -dx \\ 0 & 1 & 0 & -dy \\ 0 & 0 & 1 & -dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

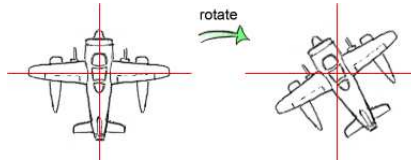
Az eltolás hatására a tárgy önmagával párhuzamosan a  $\mathbf{d}$  vektornak megfelelő irányba és távolságra tolódik el.

### 6.1.3. Forgatás (*Rotation*)

Térbeli forgatásnál szükség van egy  $e$  egyenesre, mely körül forgatunk, valamint egy  $\theta$  (theta) szögre, hogy milyen mértékkel. Az  $\theta$  egy irányított szög, ami azt jelenti, hogy pozitív szög esetén az óramutató járásával ellentétesen, míg negatív szög esetén az óramutató járásával megegyezően forgatunk. Egyenes körüli elforgatás esetén az egyenes pontjai fixpontok, azaz a forgatás során a képpont megegyezik az eredeti ponttal. Ha a  $P$  pontot az  $e$  egyenes nem tartalmazza, akkor a képe az a  $P'$  pont lesz, melyre teljesül, hogy  $P$  és  $P'$  pontnak az  $e$  egyenestől vett távolsága megegyezik, valamint  $PeP'$  szög nagysága és iránya a megadott  $\theta$  szög.

A legegyszerűbb, ha a forgatás tengelyének valamelyik koordináta-tengelyt választjuk. Amennyiben a  $z$  tengelyt választjuk, akkor a pontnak és a képpontnak a  $z$  tengelytől vett távolsága nem változik. Ezen térbeli elforgatás visszavezethető egy origó körüli síkbeli elforgatásra, ahol az elforgatás szöge megegyezik a térbeli elforgatás szögével. Síkbeli elforgatás esetén pedig:

$$\begin{aligned}x' &= x \cos \theta - y \sin \theta \\y' &= x \sin \theta + y \cos \theta\end{aligned}$$



6.2. ábra. Síkbeli elforgatás.[107]

Ennek megfelelően a  $z$  tengely körüli elforgatás mátrixa:

$$\mathbf{R}_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_z^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ha az elforgatás tengelyének az  $x$  vagy az  $y$  tengely valamelyikét választjuk, akkor a mátrix a következőképpen módosul:

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_x^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_y^{-1} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vegyük észre, hogy az  $y$  tengely körüli forgatás esetén a főátló felett a  $\sin(\theta)$  előjele pozitív ellentétben a másik két forgatással! A forgatás pozitív irányát úgy határozzuk meg, hogy a forgástengelynek kiválasztott koordináta tengely pozitív oldaláról nézünk az origóba. A forgás tengely pontonként fix, tehát a másik két tengely elforgatásánál vesszük figyelembe, hogy az óramutató járásával megegyező, vagy ellentétes irányba forgatunk. Az óramutató járásával ellentétes irányt (counter clock-wise) nevezzük pozitív illetve a megegyező irány (clock-wise)negatív forgatásnak. Tehát így értelmezhetünk negatív forgatási szöget is. A forgatás inverzét megkaphatjuk, ha a szög ellentétét használjuk, azaz a forgatás mátrixában a  $\sin \theta$  helyett  $\sin(-\theta)$ -t, míg  $\cos \theta$  helyett  $\cos(-\theta)$ -t írunk:

$$\begin{aligned} \mathbf{R}^{-1}(\theta) &= \mathbf{R}(-\theta), \\ \cos(-\theta) &= \cos(\theta), \quad \sin(-\theta) = -\sin(\theta), \end{aligned}$$

$$\begin{aligned} \mathbf{R}_z^{-1}(\theta) &= \mathbf{R}_z(-\theta) = \mathbf{R}_z = \\ &= \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Forgatási mátrix esetében az inverz mátrix megegyezik (otrogonális mátrixok) a transzponált mátrixszal:

$$\mathbf{R}^{-1} = \mathbf{R}^T \quad .$$

#### 6.1.4. Tükrözés (*Reflection*)

A térbeli tükrözés esetén szükség van egy  $s$  síkra, amelyre vonatkoztatva végezzük el a tükrözést. A tér egy  $P$  pontjához úgy rendeljük hozzá a képét, hogyha  $P$  illeszkedik az  $s$  síkra, akkor a képe önmaga; ha a  $P$  nem illeszkedik az  $s$  síkra, akkor azt a  $P'$  pontot rendeljük hozzá, amelyre fennáll, hogy a  $P$ - $P'$  szakasz felezőmerőleges síkja az  $s$  sík.

Elemi tükrözési transzformációk esetén a tükrözést a koordinátasíkokra hajtjuk végre. Ha az  $(x,y)$  síkra történik a tükrözés, akkor a pontnak az  $x$ , valamint  $y$  tengelytől való távolsága nem változik, viszont a  $z$  tengelytől való távolsága az ellentéte lesz az eredeti ponthoz tartozónak. Ennek megfelelően ennek a tükrözésnek a mátrixa:

$$\mathbf{M}_{(x,y)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}_{(x,y)}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Az  $(y,z)$ , valamint  $(x,z)$  síkokra vonatkozó tükrözésnél - ugyanilyen gondolatmenet alapján - az első, illetve a második oszlopban lesz -1.

A tükrözés inverzének mátrixa megegyezik a tükrözés mátrixával. Ez abból következik, hogyha egy pontot ugyanarra a síkra nézve kétszer tükrözünk, akkor a kétszeres tükrözést követően visszakapjuk az eredeti pontot.

#### 6.2. Hasonlósági transzformáció (*Similarity transformation*)

Az egybevágósági transzformációk kibővítésével a hasonlósági transzformációkat kapjuk. Egy ponttranszformációt hasonlóságnak nevezünk, ha bármely két pont képének a távolsága a pontok távolságával osztva mindig ugyanazt a nullától különböző hányadost adja. A képtávolságok és a megfelelő tárgytávolságok aránya adja a hasonlóság arányát. Mivel távolságokról van szó, ezért a hasonlóság aránya pozitív szám. Amennyiben a hasonlóság aránya  $\lambda$  (lambda) egynél kisebb, akkor kicsinyítésről, ha egynél nagyobb, akkor nagyításról beszélünk. Ha a  $\lambda$  értéke pontosan 1, akkor az azonosság.

Minden hasonlósági transzformáció az origóra történő kicsinyítés, vagy

nagyítás segítségével állítható elő. Ekkor egy  $P$  pont képének koordinátái a  $P$  pont koordinátáinak  $\lambda$ -szorosaként állnak elő:

$$\mathbf{M} = \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}^{-1} = \begin{bmatrix} \frac{1}{\lambda} & 0 & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A hasonlósági mátrix inverzét megkapjuk, ha a hasonlósági arány reciprokát használjuk.

### 6.3. Affin transzformációk (*Affine transformations*)

A transzformációk nagyobb halmazát az affin transzformációk adják. Az affinitás olyan transzformáció, amely párhuzamos egyeneseket párhuzamos egyenesekbe képez le. A párhuzamosságtartás egyik következménye, hogy paralelogrammának paralelogramma a képe, tehát pl. a négyzetrács képe egy paralelogrammarács (lásd [95]).

Az affinitás természetesen nem elfajuló transzformáció, és affin transzformációk egymásutánja is affinitást eredményez.

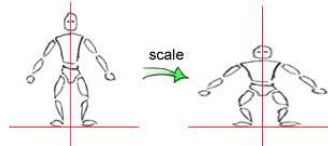
Természetesen a hasonlósági transzformációk, és ezen belül az egybevágóságok is az affin transzformációk halmazának egy részhalmaza, mivel azok egyeneseltartó transzformációk. A két legismertebb affinitás a skálázás és a nyírás.

Az euklideszi tér pontjainak homogén alakján ezeket a transzformációkat  $4 \times 4$ -es, nem szinguláris mátrixokkal való szorzással hajthatjuk végre. Az affin transzformációk mátrixának utolsó sorának első három eleme 0, negyedik eleme 1 (vagy bármilyen 0-tól különböző szám). Az ilyen mátrixszal szorozva minden közönséges pont képe közönséges pont, az ideális pontok képe ideális pont.

Általánosságban elmondható, hogy egy  $4 \times 4$ -es nem szinguláris mátrix (a mátrix determinánsa nem zérus), melynek negyedik sora  $(0,0,0,c)$ , és  $c \neq 0$  egy affin transzformációt ír le.

### 6.3.1. Skálázás (*Scaling*)

A skálázással az origóból a három koordinátatengely mentén megadott mértékű nyújtást, zsugorítást érhetünk el. Ennél a ponttranszformációnál a kép már nem egybevágó mása a tárgynak.



6.3. ábra. Síkbeli skálázás.[109]

A transzformáció mátrixa:

$$\mathbf{M} = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \nu & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}^{-1} = \begin{bmatrix} \frac{1}{\lambda} & 0 & 0 & 0 \\ 0 & \frac{1}{\mu} & 0 & 0 \\ 0 & 0 & \frac{1}{\nu} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ahol  $\lambda, \mu, \nu$  ("lambda, mú, nú") a nagyítások az egyes tengelyek mentén. Ha ezek egyenlők, akkor a tárgy arányos nagyítással (kicsinyítéssel) került leképezésre, egyébként torzulás lép fel.

Amennyiben egy közös skálázási faktort szeretnénk számolni, ebben az esetben elegendő az egységmátrix egyetlen elemét módosítani:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/s \end{bmatrix} \quad \mathbf{M}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix}$$

A közös skálázási faktor használata pontosan megfelel a kicsinyítésnek vagy nagyításnak.

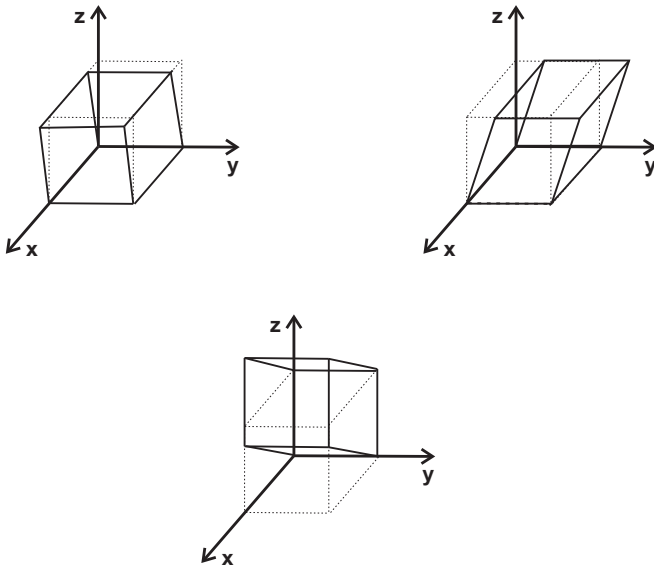
A skálázás inverzénél a  $\lambda, \mu, \nu$  reciprokaival számolunk.

### 6.3.2. Nyírás (*Shear*)

A térbeli nyírás a tér  $P$  pontjainak egy fix síkkal párhuzamos csúsztatása. Legyen a fix sík az origón áthaladó. A csúszás mértéke arányos a pontnak a fix síktól való  $d$  távolságával. A sík állása megadható egység hosszúságú  $\mathbf{n}$  normálvektorával, a csúszást pedig a  $\mathbf{t}$  irány egységvektorával, amely merőleges  $\mathbf{n}$ -re.

Egy  $\lambda$  mértékű nyírás:  $p' = p + \lambda dt = p + (\lambda \mathbf{n} \cdot \mathbf{p})\mathbf{t}$  alakba írható (lásd [48]). Ennek mátrixát a következőképpen írhatjuk fel:

$$\mathbf{M} = \begin{bmatrix} 1 + \lambda t_x n_x & \lambda t_x n_y & \lambda t_x n_z & 0 \\ \lambda t_y n_x & 1 + \lambda t_y n_y & \lambda t_y n_z & 0 \\ \lambda t_z n_x & \lambda t_z n_y & 1 + \lambda t_z n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



6.4. ábra. Nyírás.

A fenti ábrákon egy adott tengely irányában végeztük a nyírást. A legelsőn az  $x$  tengely irányában  $x$ - $y$  sík mentén, ebben az esetben a transzfor-



mációt leíró egyenletrendszer a következő (kotangens szokásos ctg jelölése helyett a cot-ot használtuk):

$$\begin{aligned}x' &= x + y \cot \theta \\y' &= y \\z' &= z\end{aligned}$$

Mátrix alakban  $a = \cot \theta$  jelölést bevezetve:

$$\mathbf{Sh}_x(a) = \begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

hasonlóan

$$\mathbf{Sh}_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{Sh}_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & a & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

adódik. A nyírás inverze:

$$\mathbf{Sh}_x^{-1}(a) = \mathbf{Sh}_x(-a) \begin{bmatrix} 1 & -a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Végezzük el a nyírást speciálisan az  $x$ , és  $y$  tengely irányában, a  $z$  tengely pedig legyen fix. Legyen a nyírási faktor  $a$  az  $x$  tengely irányában, illetve  $b$

az  $y$  irányában. Ebben az esetben a mátrix a következőre egyszerűsödik:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}^{-1} = \begin{bmatrix} 1 & 0 & -a & 0 \\ 0 & 1 & -b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A mátrix alapján leolvashatóak a következő egyenlőségek:

$$\begin{aligned} x' &= x + az \\ y' &= y + bz \\ z' &= z \end{aligned}$$

Ami azt jelenti, hogy a tér egy  $(x, y, z)$  pontja az  $(x + az, y + bz, z)$  pontba transzformálódik. Az  $x$  és  $z$  irányban elvégzett nyírás mátrixa:

$$\mathbf{M} = \begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}^{-1} = \begin{bmatrix} 1 & -a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Az  $y$  és  $z$  irányban elvégzett nyírás mátrixa:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ b & 1 & 0 & 0 \\ c & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -b & 1 & 0 & 0 \\ -c & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### 6.4. Projektív transzformáció (*Projective transformation*)

Megállapítható, hogy minden nem szinguláris  $4 \times 4$ -es mátrixnak megfeleltethető egy projektív transzformáció. A projektív transzformációnál csak az a megkötés, hogy az egyenes képe egyenes maradjon. Így elvesznek viszont olyan tulajdonságok, mint például a párhuzamosság, vagy a szögtartás. Tehát egy négyzet képe általános négyszög is lehet, kör képe pedig lehet para-

bola vagy hiperbola is.

## 6.5. Térbeli transzformációk szorzata (*Concatenation of transformations*)

Térbeli transzformációk egymás utáni végrehajtását térbeli transzformációk szorzásának illetve szorzatának nevezzük. Térbeli alakzatokon több transzformációt végrehajthatunk egymás után. Ha összetett transzformációt szeretnénk végrehajtani, akkor ennek elvégzését leegyszerűsíthetjük úgy, hogy az egyes transzformációkat leíró mátrixokat a megfelelő sorrendben összeszorozzuk, és a pontokat csak ezzel a szorzatmátrixszal kell megszorozni. Legyen  $\mathbf{M}_1$  az első,  $\mathbf{M}_2$  a második transzformáció mátrixa. Ekkor:

$$\mathbf{p}' = \mathbf{M}_1 \cdot \mathbf{p}$$

és

$$\mathbf{p}'' = \mathbf{M}_2 \cdot \mathbf{p}'$$

ami átírható

$$\mathbf{p}'' = \mathbf{M}_2 \cdot (\mathbf{M}_1 \cdot \mathbf{p}) = (\mathbf{M}_2 \cdot \mathbf{M}_1) \cdot \mathbf{p}$$

alakba. Ekkor a transzformációs mátrixokat össze lehet szorozni:

$$\mathbf{M}_3 = \mathbf{M}_2 \cdot \mathbf{M}_1$$

amiből

$$\mathbf{p}'' = \mathbf{M}_3 \cdot \mathbf{p}$$

A sorrend lényeges, lévén a mátrixszorzás nem kommutatív, így különböző sorrendben különböző eredményhez, tehát különböző transzformációhoz jutunk.

A pontranszformáció általános megadásánál balról szoroztunk a transzformációs mátrixszal. Ha jobbról szoroztunk volna, akkor egy teljesen ekvivalens rendszert tudtunk volna felépíteni, ilyen pl DirectX által használt rendszer is. Ebben az esetben a transzformációs mátrixok transzponáltja-

it kell használni, s az összetett transzformáció esetén egyenes sorrendben szorozzuk össze a transzformációs mátrixokat.

Minden összetett transzformáció felírható elemi transzformációk szorzataként. Egy tetszőleges mozgás felírható eltolásokból és forgatásokból; az általános egybevágósági transzformáció eltolásokból, tükrözésekből és forgatásokból; az általános hasonlósági transzformáció egybevágósági transzformációkból és origó középpontú hasonlóságokból, az általános affin transzformáció egybevágósági transzformációkból és skálázásokból állítható elő.

Könnyen belátható, hogy még a forgatás és az eltolás is összetett transzformáció. Mindkettő felírható tükrözés segítségével. Két párhuzamos egyenesre vonatkozó tükrözés szorzata eltolás. Az eltolás iránya merőleges az egyenesekre, nagysága a tengelyek távolságának kétszerese. Két, közös ponttal rendelkező egyenesre vonatkozó tükrözés egymásutánja a két egyenes metszéspontja körüli elforgatásnak felel meg, ahol az elforgatás szöge a két egyenes által bezárt szög kétszerese, iránya pedig a tükrözés sorrendjétől függ. Ebből következik, hogy a mozgás előáll két tükrözés, az egybevágóság pedig legfeljebb három tükrözés szorzataként.

A transzformációs mátrix négy fő részre osztható attól függően, hogy a koordináta-transzformáció során milyen jellegű változásokat eredményeznek az együtthatók:

$$\mathbf{p}' = \mathbf{M} \cdot \mathbf{p}$$

esetén

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}$$

- A bal felső  $3 \times 3$ -as részmátrix: az elforgatás, tükrözés, skálázás,
- A jobb felső  $3 \times 1$ -es részmátrix: az eltolás,
- A bal alsó  $1 \times 3$ -as részmátrix: projektív jelleg,
- A jobb alsó  $1 \times 1$ -es részmátrix: kicsinyítés, nagyítás (azonos mértékű skálázás).

Például szeretnénk egy tárgyat a  $z$  tengellyel párhuzamos egyenesen lévő  $(tx, ty, 0)$  koordinátájú pontban adott szöggel forgatni. Ezt a forgatást három elemi transzformáció szorzatára lehet felbontani:

1. Eltoljuk a tárgyat az origóba ( $\mathbf{T}_1$ ).
2. Elforgatjuk a megadott szöggel ( $\mathbf{R}$ ).
3. Visszatoljuk a tárgyat az eredeti pozíciójába ( $\mathbf{T}_2$ ).

Ennek megfelelően a transzformációs mátrixot a következőképpen kaphatjuk meg:

$$\begin{aligned} \mathbf{T}_2 \mathbf{R} \mathbf{T}_1 &= \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -tx \\ 0 & 1 & 0 & -ty \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & (-tx \cos \alpha + ty \sin \alpha + tx) \\ \sin \alpha & \cos \alpha & 0 & (-tx \sin \alpha - ty \cos \alpha + ty) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

## 6.6. Forgatás a három koordináta-tengely körül

Egy általános forgatás előállítható a három koordinátatengely körüli forgatással, viszont figyelembe kell venni, hogy a forgatások sorrendje nem felcserélhető. Pl. DirectX-ben a `D3DXMatrixRotationYawPitchRoll` függvénnyel lehet elvégezni a forgatást, ahol a sorrend:

1. Csavarás (roll), azaz  $z$  tengely körüli forgatás,
2. Billentés (pitch) következik, azaz  $x$  tengely körüli forgatás,
3. Fordulás (yaw) azaz  $y$  tengely körüli forgatás.

Ha balról szorzunk a transzformációs mátrixszal ( $\mathbf{P}' = \mathbf{M} \cdot \mathbf{P}$ ), akkor a összetett transzformáció esetén fordított sorrendben (lásd a *Térbeli transzformációk szorzata* fejezetet) kell összeszoroznunk a transzformációs mátrixokat:

$$\mathbf{R}_z = \begin{bmatrix} \cos z & -\sin z & 0 & 0 \\ \sin z & \cos z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos x & -\sin x & 0 \\ 0 & \sin x & \cos x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_y = \begin{bmatrix} \cos y & 0 & \sin y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin y & 0 & \cos y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\mathbf{R}_{zxy} = \mathbf{R}_y \cdot \mathbf{R}_x \cdot \mathbf{R}_z =$$

$$\begin{bmatrix} \cos y \cos z + \sin x \sin y \sin z & -\cos y \sin z + \cos z \sin x \sin y & \cos x \sin y & 0 \\ \cos x \sin z & \cos x \cos z & -\sin x & 0 \\ -\cos z \sin y + \cos y \sin x \sin z & \sin y \sin z + \cos y \cos z \sin x & \cos x \cos y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

ha jobbról szorzunk, akkor az előbbi mátrix transzponáltját kapjuk. Mivel a forgatások sorrendje kötött, ezért általánosan a tetszőleges tengely körüli forgatást használjuk.

## 6.7. Tetszőleges tengely körüli forgatás (*Rotation about an arbitrary axis*)

A tetszőleges tengely körüli forgatás gyakori feladat a robotikában, az animációk készítésében és a szimulációk esetében. Követve az tranzformációk egymás utáni végrehajtásáról tanultakat, egyszerűen adódik a megoldás. Vezessük vissza a problémát a koordináta-tengely körüli forgatásra. Tegyük fel, hogy a térbeli tengely adott két pontjával:  $P_0(x_0, y_0, z_0)$  ill.  $P_1(x_0, y_0, z_0)$ , továbbá a forgatás szöge legyen  $\theta$ .

Ebben az esetben a következő tranzformációkat kell sorban végrehajtunk:

1. Eltoljuk a  $P_0(x_0, y_0, z_0)$  tengelypontot az origóba.
2. Végrehajtjuk a megfelelő forgatásokat, úgy hogy a tranzformációk eredményeképpen forgástengely a  $z$  koordinátatengelyre illeszkedik.
3. Elvégezzük a  $\theta$  szöggel való forgatást a  $z$  tengely körül.
4. Végrehajtjuk a 2. lépésben kivitelezett forgatások inverzét.
5. Elvégezzük el az 1. lépésben lévő eltolás inverzét.

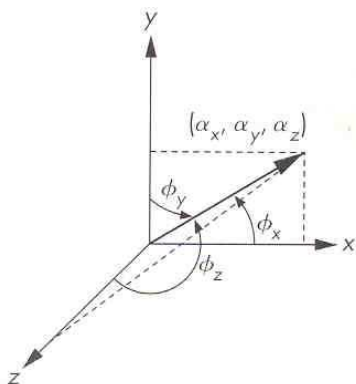
Az egyszerűség kedvéért előállítjuk az  $\mathbf{u} = P_1 - P_0$  vektort, amely a forgás tengellyel párhuzamos, majd normalizáljuk:

$$\mathbf{u} := \frac{\mathbf{u}}{|\mathbf{u}|} = (c_x, c_y, c_z) \quad .$$

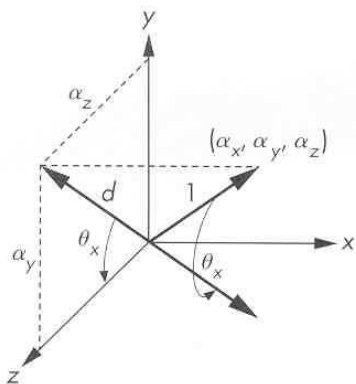
Az  $\mathbf{u}$  vektor komponenseit az origón átmenő tengely iránykoszinuszainak nevezzük (lásd 6.5 ábra), és teljesül a következő összefüggés:

$$c_x^2 + c_y^2 + c_z^2 = 1, \\ \cos \varphi_x = c_x, \quad \cos \varphi_y = c_y, \quad \cos \varphi_z = c_z.$$

A második lépésben két forgatásra lesz szükség. Az első egy  $x$  koordinátatengely körüli pozitív irányú forgatás  $\theta_x$  szöggel, úgy hogy a forgástengely az  $x - z$  koordinátasíkra fog illeszkedni (lásd 6.6 ábra).



6.5. ábra. Iránykoszinuszok



6.6. ábra.  $x$  tengely körüli forgatás

Az 6.6 ábráról leolvasható, hogy  $d = \sqrt{c_y^2 + c_z^2}$ , és a forgatás szögének szögfüggvényei:

$$\sin \theta_x = \frac{c_y}{d}, \quad \cos \theta_x = \frac{c_z}{d}$$

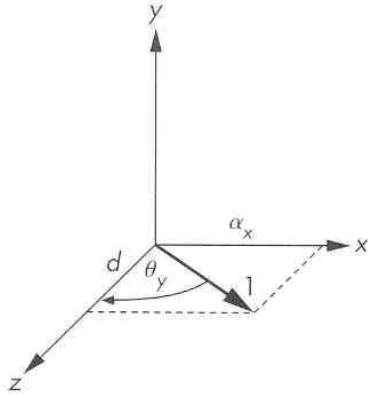
és ebből adódik a forgatás mátrixa:

$$\mathbf{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_z/d & -c_y/d & 0 \\ 0 & c_y/d & c_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

A második egy  $y$  koordinátatengely körüli negatív irányú forgatás  $\theta_y$



szöggel, úgy hogy a forgástengely a  $z$  koordinátatengelyre illeszkedjen.



6.7. ábra.  $z$  tengely körüli forgatás

Az 6.7 ábráról leolvashatóak a forgatás szögének szögfüggvényei:

$$\sin \theta_y = d, \quad \cos \theta_y = \alpha_x$$

és ebből adódik a forgatás mátrixa figyelembe véve a negatív irányt:

$$\mathbf{R}_y(\theta_y) = \begin{bmatrix} d & 0 & -\alpha_x & 0 \\ 0 & 1 & 0 & 0 \\ \alpha_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

A teljes transzformáció a transzformációs mátrixok fordított sorrendben történő összeszorzásából adódik

$$\mathbf{M} = \mathbf{T}(\mathbf{p}_0)\mathbf{R}_x(-\theta_x)\mathbf{R}_y(-\theta_y)\mathbf{R}_z(\theta)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)\mathbf{T}(-\mathbf{p}_0) \quad ,$$

ahol a belső öt mátrixot szokás külön is kiemelni:

$$\mathbf{R} = \mathbf{R}_x(-\theta_x)\mathbf{R}_y(-\theta_y)\mathbf{R}_z(\theta)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x) \quad , \quad (6.1)$$

azaz

$$\mathbf{M} = \mathbf{T}(\mathbf{p}_0)\mathbf{R}\mathbf{T}(-\mathbf{p}_0) \quad .$$

Az 6.1-ben lévő általános forgatási mátrixot Rodrigues formulának vagy Rodrigues-féle forgatási mátrixnak nevezzük, melynek a legtöbb irodalomban teljesen más levezetését tanulmányozhatjuk (lásd [105], vagy [11]). Belátható, hogy ha az öt forgatási mátrixot összeszorozzuk és elvégezzük a trigonometrikus függvényeken értelmezett azonos átalakításokat (lásd [98]), akkor az  $\mathbf{R}$  mátrix következő alakját kapjuk:

$$\mathbf{R} =$$

$$\begin{bmatrix} \cos \theta + c_x^2(1 - \cos \theta) & c_x c_y(1 - \cos \theta) - c_z \sin \theta & c_x c_z(1 - \cos \theta) + c_y \sin \theta & 0 \\ c_y c_x(1 - \cos \theta) + c_z \sin \theta & \cos \theta + c_y^2(1 - \cos \theta) & c_y c_z(1 - \cos \theta) - c_x \sin \theta & 0 \\ c_z c_x(1 - \cos \theta) - c_y \sin \theta & c_z c_y(1 - \cos \theta) + c_x \sin \theta & \cos \theta + c_z^2(1 - \cos \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Mivel a forgatás mátrixának az inverze a mátrix transzponáltja, ezért teljesül, hogy  $\mathbf{R}^{-1} = \mathbf{R}^T$ . A fentiekben megadott levezetés azért szerencsés, mert segítségével könnyedén áttérhetünk a tetszőleges síkra való tükrözés megoldására is.

## 6.8. Tetszőleges síkra való tükrözés (*Reflection about an arbitrary plane*)

Szintén gyakori feladat, amikor egy testet az egyik oldalapjának a síkjára szeretnénk tükrözni. Vezessük vissza a problémát a koordinátasíkra való tükrözésre. Figyelembe véve az előző fejezet levezetését adódik:

1. Eltoljuk el a tükrözési síkot egy  $P_0(x_0, y_0, z_0)$  pontjának segítségével az origóba.
2. Megfelelő forgatásokat hajtunk végre, úgy hogy a tükrözési sík normálvektora a  $z$  koordinátatengely pozitív irányába mutasson, ezáltal a tükrözési sík illeszkedni fog a  $z = 0$ , azaz az  $x - y$  koordináta síkra.
3. Az  $z = 0$  koordinátasíkra a tükrözést végzünk el.
4. Végrehajtjuk a 2. lépésben kivitelezett forgatások inverzét.

5. Elvégezzük az 1. lépésben leírt eltolás inverzét.

Tehát ha adott a sík  $P_0(x_0, y_0, z_0)$ ,  $P_1(x_1, y_1, z_1)$  és  $P_2(x_2, y_2, z_2)$ , három nem egy egyenesre eső (nem kollineáris) pontjával, akkor a sík normál vektora könnyedén kiszámolható az  $\mathbf{a} = P_1 - P_0$ ,  $\mathbf{b} = P_2 - P_0$  vektorok vektoriális szorzatából:

$$\mathbf{n} = \mathbf{a} \times \mathbf{b} \quad ,$$

amelyet normalizálunk

$$\mathbf{n} := \frac{\mathbf{n}}{|\mathbf{n}|} = (c_x, c_y, c_z) \quad .$$

ahol az  $\mathbf{n}$  vektor komponenseit iránykoszinuszoknak nevezzük, s az 6.5 ábra alapján igaz, hogy

$$c_x^2 + c_y^2 + c_z^2 = 1.$$

A 2. lépésben lévő forgatási mátrixok ugyanazok lesznek, mint amelyeket a tetszőleges tengely körüli forgatásnál alkalmaztunk. A teljes transzformáció a transzformációs mátrixok fordított sorrendben történő összeszorzásából adódnak

$$\mathbf{M} = \mathbf{T}(\mathbf{p}_0)\mathbf{R}_x(-\theta_x)\mathbf{R}_y(-\theta_y)\mathbf{R}_{reflect}^{(x,y)}\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)\mathbf{T}(-\mathbf{p}_0) \quad ,$$

ahol a belső öt mátrixot szokás külön is kiemelni:

$$\mathbf{R}_{reflect} = \mathbf{R}_x(-\theta_x)\mathbf{R}_y(-\theta_y)\mathbf{R}_{reflect}^{(x,y)}\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x) \quad , \quad (6.2)$$

azaz

$$\mathbf{M} = \mathbf{T}(\mathbf{p}_0)\mathbf{R}_{reflect}\mathbf{T}(-\mathbf{p}_0) \quad . \quad (6.3)$$

Az 6.2-ben lévő általános tükrözési mátrixnak nincs külön neve az irodalomban, levezetése megtalálható a [98] cikkben. Belátható, hogy ha az öt mátrixot összeszorozzuk és elvégezzük a trigonometrikus függvényeken

értelmezett azonos átalakításokat, akkor a következő alakot kapjuk:

$$\mathbf{R}_{reflect} = \begin{bmatrix} 1 - 2c_x c_x & -2c_y c_x & -2c_z c_x & 0 \\ -2c_y c_x & 1 - 2c_y c_y & -2c_y c_z & 0 \\ -2c_z c_x & -2c_y c_z & 1 - 2c_z c_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

miel a tükrözés inverze önmaga, ezért  $\mathbf{R}_{reflect}^{-1} = \mathbf{R}_{reflect}$ . Szokás még a 6.3-ben levő szorzást is elvégezni, mivel az eredmény nem lesz bonyolult:

$$\mathbf{R}_{reflect} = \begin{bmatrix} 1 - 2c_x^2 & -2c_x c_y & -2c_x c_z & -2c_x d \\ -2c_x c_y & 1 - 2c_y^2 & -2c_y c_z & -2c_y d \\ -2c_x c_z & -2c_y c_z & 1 - 2c_z^2 & -2c_z d \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

ahol  $d = -c_x x_0 - c_y y_0 - c_z z_0$ . Ugyanezt az eredmény használja a DirectX a D3DXMatrixReflect függvényben.

## 7. Koordináta-transzformációk (*Coordinatesystem transformation*)

Koordináta-transzformációról akkor beszélünk, ha a tárgy pont egy új koordináta-rendszerre vonatkozó koordinátáit határozzuk meg, a régiek ismeretében. Ilyenkor tehát a vizsgált tárgy változatlan, csupán a nézőpontunkat változtatjuk meg. A grafikus objektum változatlan marad.

A legtöbb alakzat koordináta-geometriai vizsgálata egyszerűbbé válik, ha az alakzathoz képest a koordináta-rendszert alkalmasan – rendszerint valamilyen speciális helyzetben – választjuk meg. Ehhez általában az szükséges, hogy egy meglévő koordináta-rendszerről áttérjünk egy új koordináta-rendszerre.

A koordináta-transzformációt a komputergrafikában szokásos problémán keresztül ismertetjük. A tárgyakat általában a világkoordináta-rendszerben (*world*) adjuk meg, s innen akarunk áttérni a kamera- vagy nézetikoordináta-rendszerre (*view*). Minkét rendszer Descartes-féle derékszögű koordináta-rendszer. Tegyük fel, hogy a nézetikoordináta-rendszer bázisvektorai és origója a világkoordináta-rendszerben rendre  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{n}$  és  $\mathbf{c}$ :

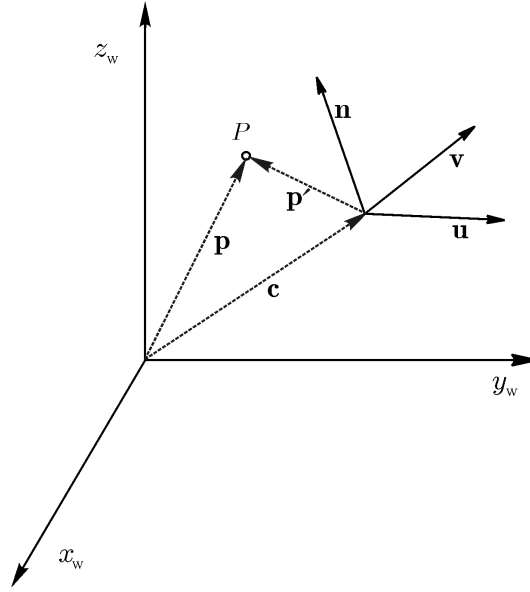
$$\mathbf{u} = [u_x, u_y, u_z], \quad \mathbf{v} = [v_x, v_y, v_z], \quad \mathbf{n} = [n_x, n_y, n_z], \quad \mathbf{c} = [c_x, c_y, c_z].$$

Először ismerjük a  $P$  pontnak az  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{n}$  nézeti (új) koordináta-rendszerbeli  $\mathbf{p}'(x_v, y_v, z_v)$  helyvektorát. Határozzuk meg a  $P$  pont világ (régi) koordináta-rendszerbeli  $\mathbf{p}(x_w, y_w, z_w)$  helyvektorát (lásd a 7.1 ábrán).

$$\mathbf{p} = \mathbf{c} + \mathbf{p}' = x_v \mathbf{u} + y_v \mathbf{v} + z_v \mathbf{n} + \mathbf{c},$$

aminek mátrix reprezentációja homogén koordinátákat felhasználva

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \mathbf{T}_{world} \cdot \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix},$$



7.1. ábra. Koordináta-transzformáció

$$\mathbf{T}_{world} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{n} & \mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_x & v_x & n_x & c_x \\ u_y & v_y & n_y & c_y \\ u_z & v_z & n_z & c_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Másodszor tekintsük a feladat inverzét, ami a komputergrafikában gyakoribb probléma. Ismerjük a  $P$  pontnak a világ (rég) koordináta-rendszerbeli  $\mathbf{p}(x_w, y_w, z_w)$  helyvektorát. Határozzuk meg a  $P$  pont  $\mathbf{u}, \mathbf{v}, \mathbf{n}$  nézeti (új) koordináta-rendszerbeli  $\mathbf{p}'$  helyvektorának  $(x_v, y_v, z_v)$  komponenseit.

$$\mathbf{p}' = \mathbf{p} - \mathbf{c},$$

$\mathbf{p}'$ -nek az új koordináta-rendszerbeli komponenseire

$$\begin{aligned} x_v &= \mathbf{p}' \cdot \mathbf{u} = (\mathbf{p} - \mathbf{c}) \cdot \mathbf{u} = \mathbf{p}' \cdot \mathbf{u} - \mathbf{c} \cdot \mathbf{u} \\ y_v &= \mathbf{p}' \cdot \mathbf{v} = (\mathbf{p} - \mathbf{c}) \cdot \mathbf{v} = \mathbf{p}' \cdot \mathbf{v} - \mathbf{c} \cdot \mathbf{v} \\ z_v &= \mathbf{p}' \cdot \mathbf{n} = (\mathbf{p} - \mathbf{c}) \cdot \mathbf{n} = \mathbf{p}' \cdot \mathbf{n} - \mathbf{c} \cdot \mathbf{n} \quad , \end{aligned}$$

aminek mátrix reprezentációja homogén koordinátákat felhasználva

$$\begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z & -\mathbf{c} \cdot \mathbf{u} \\ v_x & v_y & v_z & -\mathbf{c} \cdot \mathbf{v} \\ n_x & n_y & n_z & -\mathbf{c} \cdot \mathbf{n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}.$$

Az előzőekhez hasonlóan felírhatjuk, hogy

$$\begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix} = \mathbf{T}_{view} \cdot \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix},$$

ahol

$$\mathbf{T}_{view} = \mathbf{T}_{world}^{-1}.$$

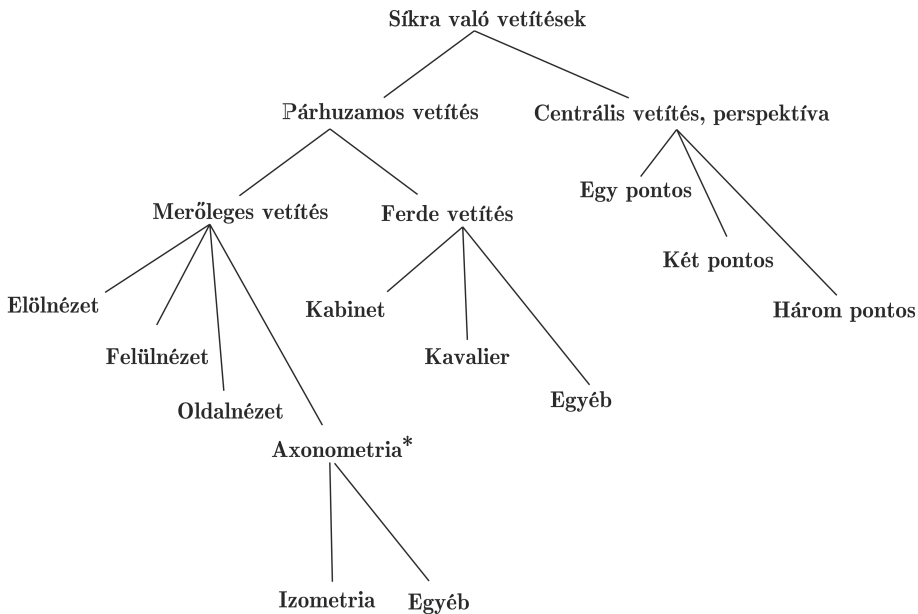
A  $\mathbf{T}_{world}$  mátrix inverze könnyen előállítható, hiszen ekkor a bal-felső minor-mátrix ortonormált mátrix (a sorvektorai egymásra merőleges egységvektorok), tehát annak inverze egyenlő a transzponáltjával, azaz

$$\mathbf{T}_{view} = \mathbf{T}_{world}^{-1} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

ahol a második mátrix egy eltolás ponttranszformációját írja le, nevezetesen a kamera pozíciójának az eltolását a világkoordináta-rendszer origójába. Az első mátrix két koordináta-tengely körüli forgatást és ha az új rendszernek a sodrása is (pl.jobbsodrásúból balsodrásúvá) megváltozik, akkor koordináta-síkra való tükrözést is tartalmaz.

## 8. Tér leképezése a síkra

Általában a vetítési ponttranszformáció alatt azt értjük, amikor  $n$  dimenziós koordináta-rendszer pontjait vetítjük  $n$ -nél kevesebb dimenziójú koordináta-rendszer pontjaiba. Leggyakrabban a 3 dimenziós tér pontjait vetítjük a 2 dimenziós síkra. A vetítéshez meg kell adnunk a *vetítés centrumát*, és amire vetítünk, azaz a *képsíkot*. A centrumból indított és a térbeli objektum pontjain át húzott *vetítő sugarak* képsíkkal alkotott dőfspontjai (metszéspontjai) határozzák meg az objektum képét. Csak olyan esetekkel foglalkozunk, amit síkbeli geometria projekciónak (vetítés) nevezünk, azaz a vetítő sugarak egyenesek és nem görbék, illetve a képsík sík és nem görbült felület. A térképészetben használnak más eseteket is. hasonló az IMAX vagy Omnimax filmszínház esete.



8.1. ábra. Vetítések osztályozása

A síkra való vetítések osztályozását megtalálhatjuk a 8.1 ábrán, ami Foley könyve [23] alapján készült. A \* megjegyzés azt jelenti, hogy figyelembe kell venni az axonometria estében Pohlke-tételét: Egy alakzat axonometrikus képe hasonló az alakzat párhuzamos projekciójához. Tehát az axonometria

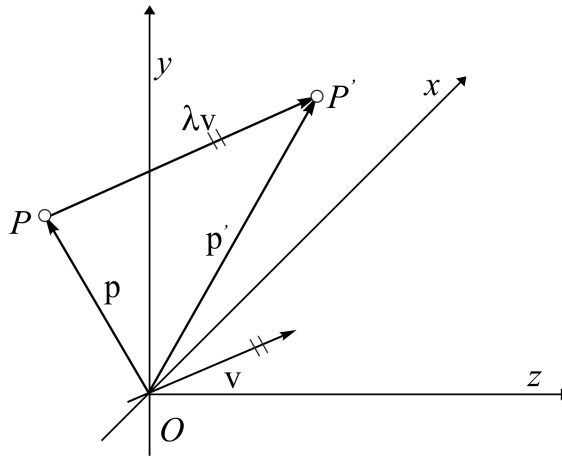


egy párhuzamos vetítés és egy hasonlósági transzformáció szorzata. A hasonlósági transzformációknak részhalmaza a helybenhagyás (a hasonlósági tényező 1), azaz létezik olyan axonometria, amely párhuzamos vetítés, de nem minden axonometria párhuzamos vetítés.

### 8.1. Párhuzamos vetítés

Ebben az esetben a vetítés centruma végtelen távoli pont, azaz homogén koordinátás alakban a negyedik koordinátája 0. A vetítésugarak amelyek illeszkednek erre a végtelen távoli pontra párhuzamosak lesznek egymással hagyományos (eukleidészi) értelemben. Ezért sok esetben csak a vetítés irányvektorát szokás megadni descartesi alakban:

$$\mathbf{v} = (vx, vy, vz)$$



8.2. ábra. Vetítések osztályozása

A képsíknak tekinthetjük az  $[x, y]$  koordináta síkot. A képsíkon egy térbeli  $P$  pont képét úgy kapjuk meg, hogy vesszük a vetítési iránnyal ( $\mathbf{v}$ -vel) párhuzamos egyenest a  $P$  ponton keresztül, majd ennek az egyenesnek és a képsíknak a metszéspontját határozzuk meg. Legyen  $P$  pont helyvektora  $\mathbf{p}(x, y, z)$ , illetve a  $P'$  képpont helyvektora  $\mathbf{p}'(x', y', z')$ , melynek kiszámítása

az 8.2 ábra alapján a következőképpen történik:

$$\mathbf{p}' = \mathbf{p} + \lambda \cdot \mathbf{v}$$

Kibontva ezt az egyenletet, a következő három egyenlőséget kapjuk:

$$\begin{aligned}x' &= x + \lambda \cdot vx \\y' &= y + \lambda \cdot vy \\z' &= z + \lambda \cdot vz\end{aligned}$$

Ezek után meg kell határoznunk  $\lambda$ -t. Az  $[x, y]$  koordináta sík egyenlete  $z = 0$ , azaz minden pontjának  $z$  koordinátája nulla. Ebből az következik:

$$z' = z + \lambda \cdot vz = 0$$

amiből  $\lambda$ -t kifejezhetjük

$$\lambda = -\frac{z}{vz}$$

Az eredményt visszahelyettesítve az első két egyenletbe a következőket kapjuk:

$$\begin{aligned}x' &= x - z \frac{vx}{vz} \\y' &= y - z \frac{vy}{vz} \\z' &= 0\end{aligned} \tag{8.1}$$

A fenti összefüggést mátrix alakban is megadjuk

$$\mathbf{p}' = \mathbf{M}\mathbf{p}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{vx}{vz} & 0 \\ 0 & 1 & -\frac{vy}{vz} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x - z \frac{vx}{vz} \\ y - z \frac{vy}{vz} \\ 0 \\ 1 \end{bmatrix}$$

Az  $\mathbf{M}$  mátrix harmadik sora csupa nulla, ami jelzi, hogy a párhuzamos vetítés dimenzió vesztes transzformáció. A párhuzamos vetítés speciális esete a merőleges vetítésről, ha vetítés iránya merőleges a képsíkra. Ebben az esetben

$$\mathbf{v} = (0, 0, vz)$$

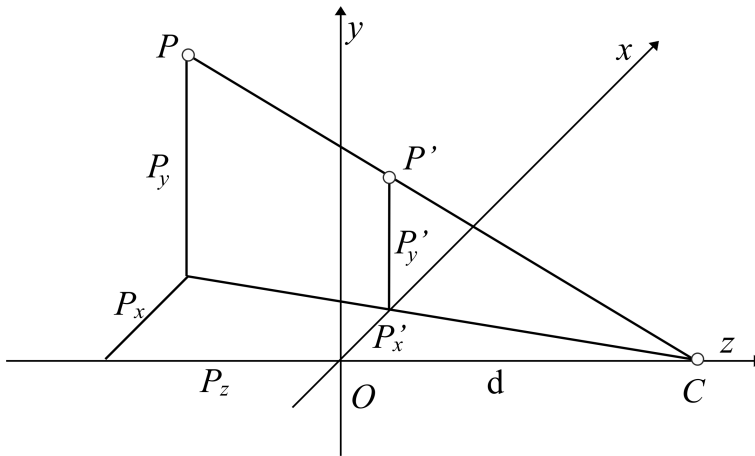
Az előbbi eredményt behelyettesítve a fenti egyenletrendszerbe, akkor a következőt kapjuk:

$$\begin{aligned}x' &= x \\y' &= y \\z' &= 0\end{aligned}$$

Tehát a legegyszerűbb vetítés az, ha elhagyjuk a térbeli  $z$  koordinátát.

## 8.2. Centrális vetítés

Ebben az esetben a vetítés centruma valós pont (a párhuzamos vetítéssel ellentétben nem végtelen távoli pont), azaz homogén koordinátás alakban a negyedik koordinátája nem 0. A vetítésugarak illeszkednek erre a centrumra és nem lesznek párhuzamosak. Centrális vetítésnél a vetítés centrumát rendszerint a  $z$  tengelyen helyezzük el. Két esetet fogunk tárgyalni.



8.3. ábra. Centrális vetítés első eset

**Első eset.** A centrum a  $z$  tengely pozitív felén található. Távolságát az origótól  $d$ -vel jelöljük. A képsíknak tekinthetjük az  $[x, y]$  koordináta ( $z = 0$ ) síkot. Egy  $P[P_x, P_y, P_z, 1]$  pont vetületét,  $P'$ -t a hasonló háromszögek közötti összefüggések alapján a következőképpen határozhatjuk meg:

$$\frac{P'_x}{d} = \frac{P_x}{d - P_z}, \quad \frac{P'_y}{d} = \frac{P_y}{d - P_z}$$

Szorozzuk meg mindkét oldalt  $d$ -vel:

$$P'_x = d \frac{P_x}{d - P_z}, \quad P'_y = d \frac{P_y}{d - P_z}$$

A  $P'_z = 0$ , mivel a képsík az  $[x, y]$  koordináta sík. A fenti összefüggést

mátrix alakban is megadjuk felhasználva a homogén koordinátás alakot:

$$P' = M_1 P$$

$$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{d} & 1 \end{bmatrix} \cdot \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ 0 \\ 1 - \frac{P_z}{d} \end{bmatrix} =$$

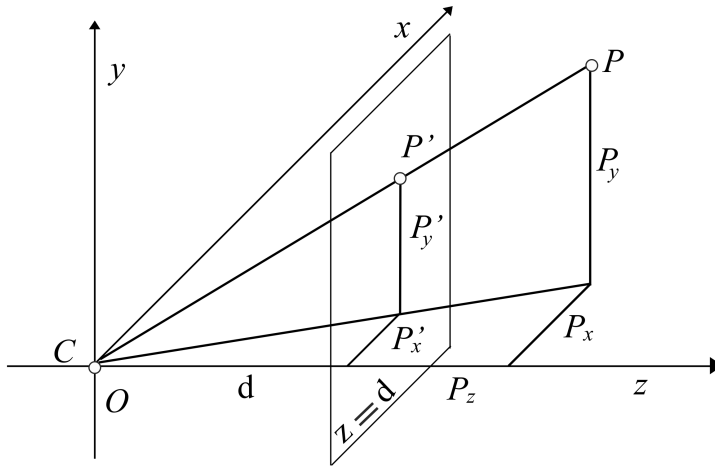
A kapott eredmény még nem használható, vissza kell térnünk a homogén koordinátás alakról a descarteszi alakra, azaz a negyedik koordinátával végigosztjuk az első három koordinátát:

$$\begin{bmatrix} P_x \\ P_y \\ 0 \\ 1 - \frac{P_z}{d} \end{bmatrix} = \begin{bmatrix} P_x \frac{d}{d-P_z} \\ P_y \frac{d}{d-P_z} \\ 0 \\ 1 \end{bmatrix}, \text{ ahol } P_z \neq d.$$

A fenti műveletet csak akkor tudjuk megtenni, ha  $P_z \neq d$ . Azon pontok mértani helye, ahol  $P_z = d$  a  $C$  centrumra illeszkedő és az  $[x, y]$  koordináta síkkal párhuzamos sík. Ezt a ( $z = d$ ) síkot eltűnési síknak nevezzük, ezen sík pontjai képe végtelen távoli pontok lesznek, azaz a végtelenbe kerülnek. Ha egy objektumot elmetsz az eltűnési sík, akkor az alakzat képe szétesik.

**Második eset.** A centrum az origóban található. Képsíknak tekinthetjük az  $[x, y]$  koordináta síkkal párhuzamos ( $z = d$ ) síkot. A képsík távolsága az origótól  $d$ -vel jelöljük. Egy  $P[P_x, P_y, P_z, 1]$  pont vetületét,  $P'$ -t a hasonló háromszögek közötti összefüggések alapján a következőképpen határozhatjuk meg:

$$\frac{P'_x}{d} = \frac{P_x}{P_z}, \quad \frac{P'_y}{d} = \frac{P_y}{P_z},$$



8.4. ábra. Centális vetítés második eset

Szorozzuk meg mindkét oldalt  $d$ -vel:

$$P'_x = d \frac{P_x}{P_z}, \quad P'_y = d \frac{P_y}{P_z}$$

A  $P'_z = d$ , mivel a  $P'$  pont illeszkedik a  $z = d$  képsíkra. A fenti összefüggést mátrix alakban is megadjuk felhasználva a homogén koordinátás alakot:

$$P' = \mathbf{M}_2 P$$

$$= \begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} \cdot \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \\ \frac{P_z}{d} \end{bmatrix} =$$

A kapott eredmény még nem használható, vissza kell térnünk a homogén koordinátás alakról a descarteszi alakra, azaz a negyedik koordinátával

végigosztjuk az első három koordinátát:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ \frac{P_z}{d} \end{bmatrix} = \begin{bmatrix} d\frac{P_x}{P_z} \\ d\frac{P_y}{P_z} \\ d \\ 1 \end{bmatrix}, \text{ ahol } P_z \neq 0.$$

A fenti műveletet csak akkor tudjuk megtenni, ha  $P_z \neq 0$ . Azon pontok mértani helye, ahol  $P_z = 0$  a  $C$  centrumra, azaz az origóra illeszkedő  $[x, y]$  koordináta sík, amely párhuzamos a képsíkkal. Ezt a ( $z = 0$ ) síkot eltűnési síknak nevezzük, ezen sík pontjai képe végtelen távoli pontok lesznek, azaz a végtelenbe kerülnek. Ha egy objektumot elmetesz az eltűnési sík, akkor az alakzat képe szétesik.

A komputergrafikában gyakori a második eset használata, mivel ekkor a kamerát képzelhetjük el az origóba.

Mind az  $\mathbf{M}_1$  és az  $\mathbf{M}_2$  mátrixok determinánsa nulla, azaz a pontranszformáció dimenzió vesztő transzformáció. Ez az is jelenti, hogy nem kölcsönösen egyértelmű a leképezés. Ha két térbeli pont képe egybeesik, akkor az ilyen pontokat fedőpontoknak nevezzük.

## 9. Görbék megadása

A görbéket általában *implicit módon* a következő alakban adhatjuk meg  $f(x, y) = 0$ . Például az origó középpontú  $r$  sugarú kör egyenlete  $f(x, y) = x^2 + y^2 - r^2$ . Szintén gyakori az *explicit* megadás is:  $y = f(x)$ . Az előbbi kör egyenletéből az egyik változóra kifejezve nyerhetjük a félkör egyenletét.  $y = \sqrt{r^2 - x^2}$ . További lehetőségünk a paraméteres egyenletrendszer alkalmazása:  $C(u) = (x(u), y(u))$ . Például:

$$\begin{aligned}x(u) &= r \cos(u) \\y(u) &= r \sin(u), \quad 0 \leq u \leq \frac{\pi}{2}\end{aligned}\tag{9.1}$$

egyenlet szintén a fenti kör egyenletét adja meg. Ez utóbbi alakot át is alakíthatjuk alkalmas helyettesítéssel:

$$t = \tan\left(\frac{u}{2}\right) \quad 0 \leq t \leq 1,$$

ami adja a következő alakot:

$$x(t) = \frac{1 - t^2}{1 + t^2} \quad y(t) = \frac{2t}{1 + t^2}.$$

Tehát megállapíthatjuk hogy a paraméteres alak többféle is lehet.

Gyakori példa a  $P_1$  és a  $P_2$  pontokat összekötő egyenes szakasz paraméteres egyenlete is, a melyet most vektor alakban adunk meg:

$$P(t) = P_1 + t(P_2 - P_1) \quad 0 \leq t \leq 1,$$

átalakítva:

$$P(t) = (1 - t)P_1 + tP_2 \quad 0 \leq t \leq 1,$$

Kifejtve a síkban két egyenlettel

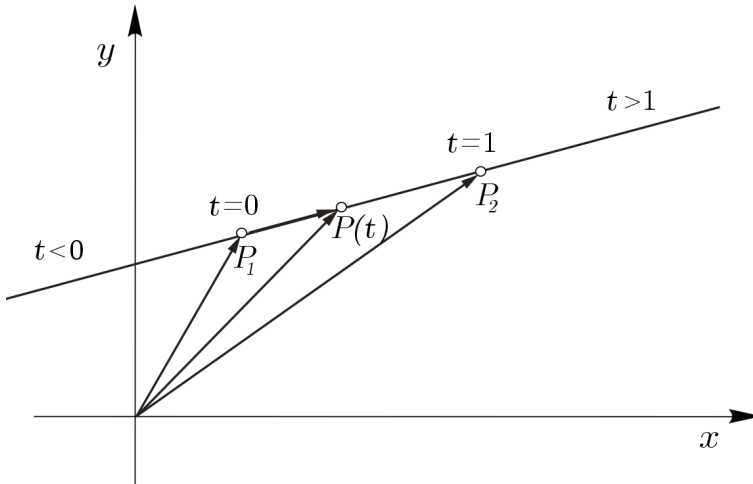
$$\begin{aligned}x(t) &= (1 - t)x_1 + tx_2 \\y(t) &= (1 - t)y_1 + ty_2\end{aligned}$$



a térben pedig egy kicsit átalakítva a szokásos módon három egyenlettel adhatjuk meg:

$$\begin{aligned}x(t) &= (1-t)x_1 + tx_2 \\y(t) &= (1-t)y_1 + ty_2 \\z(t) &= (1-t)z_1 + tz_2.\end{aligned}$$

Megjegyezzük, ha a  $t$  paramétert a teljes valós számhalmazon futtatjuk, akkor a két ponton átmenő végtelen egyenes paraméteres egyenletrendszerét kapjuk meg.



9.1. ábra. Egyenes paraméteres előállítás

## 9.1. Interpoláció

Adottak a sík vagy a tér  $\mathbf{p}_0, \dots, \mathbf{p}_n$  pontjai, ezeket kontrollpontoknak, az általuk meghatározott poligont kontrollpoligonnak nevezzük.

Keressük azt a legtöbbször  $k$ -ad fokú polinommal megadott  $\mathbf{s}(u)$  görbét, melyre találhatóak olyan  $u_0, u_1, \dots, u_n$  értékek, hogy

$$\begin{aligned}\mathbf{s}(u_0) &= \mathbf{p}_0 \\ \mathbf{s}(u_1) &= \mathbf{p}_1 \\ &\vdots \\ \mathbf{s}(u_n) &= \mathbf{p}_n\end{aligned}$$

Tehát a görbe áthalad (interpolálja) a megadott kontroll pontokat. A numerikus matematika is foglalkozik ezzel a problémával, s ad több megoldást is. Pl. ha a fenti egyenletrendszer jobboldalára  $n$ -edfokú interpolációs polinomot helyettesítünk, akkor egy lineáris egyenletrendszert kapunk, ahol az együtthatók lesznek az ismeretlenek. Az egyenletrendszer megoldható pl. Gauss-eliminációval. Emellett még a Lagrange- ill. Newton-interpoláció a legismertebbek a numerikus matematikában.

### 9.1.1. Hermite-görbe

Charles Hermite általánosabb fogalomknak tekintette a kontroll adatokat.

Keressük azt a  $\mathbf{s}(u)$  harmadfokú polinomot, ahol az input adat négy pont; vagy három pont és egy érintő; ill. két pont és két érintő lehet. Ebben a könyvben az utóbbi probléma megoldását fejtjük ki.

Tekintsük azt a harmadfokú Hermite-görbét, amely a kezdő és végpontjával ill. a kezdő és végpontban érintőjével adott. Tehát adottak a  $\mathbf{p}_0$  és  $\mathbf{p}_1$  pontok, valamint a  $\mathbf{t}_0$  és  $\mathbf{t}_1$  érintő vektorok.

Keressük az

$$\mathbf{s}(u) = \mathbf{a}_0 u^3 + \mathbf{a}_1 u^2 + \mathbf{a}_2 u + \mathbf{a}_3, \quad u \in [0,1]$$

harmadfokú polinommal megadott görbét amelyre

$$\mathbf{s}(0) = \mathbf{p}_0, \mathbf{s}(1) = \mathbf{p}_1, \dot{\mathbf{s}}(0) = \mathbf{t}_0, \dot{\mathbf{s}}(1) = \mathbf{t}_1$$

teljesül, ahol a felső pont a derivált jele. Tehát a kezdő- és a végpont, valamint a végpontokban húzott érintők ismertek. Ezek alapján az egyenletrendszer felírása és megoldása következik.

$$\begin{aligned}\mathbf{s}(0) &= \mathbf{p}_0 = \mathbf{a}_3 \\ \mathbf{s}(1) &= \mathbf{p}_1 = \mathbf{a}_0 + \mathbf{a}_1 + \mathbf{a}_2 + \mathbf{a}_3 \\ \dot{\mathbf{s}}(0) &= \mathbf{t}_0 = \mathbf{a}_2 \\ \dot{\mathbf{s}}(1) &= \mathbf{t}_1 = 3\mathbf{a}_0 + 2\mathbf{a}_1 + \mathbf{a}_2\end{aligned}$$

Az egyenletrendszer megoldása után

$$\begin{aligned}\mathbf{a}_0 &= -2(\mathbf{p}_1 - \mathbf{p}_0) + \mathbf{t}_0 + \mathbf{t}_1 \\ \mathbf{a}_1 &= 3(\mathbf{p}_1 - \mathbf{p}_0) - 2\mathbf{t}_0 - \mathbf{t}_1 \\ \mathbf{a}_2 &= \mathbf{t}_0 \\ \mathbf{a}_3 &= \mathbf{p}_0\end{aligned}$$

polinom-együtthatókat kapjuk. Ezeket visszahelyettesítve és átrendezve

$$\mathbf{s}(u) = (2u^3 - 3u^2 + 1)\mathbf{p}_0 + (-2u^3 + 3u^2)\mathbf{p}_1 + (u^3 - 2u^2 + u)\mathbf{t}_0 + (u^3 - u^2)\mathbf{t}_1.$$

Az egyenletben szereplő együttható polinomokat *Hermite-polinomoknak* nevezzük, és a következőképpen jelöljük:

$$\begin{aligned}H_0^3(u) &= 2u^3 - 3u^2 + 1, \\ H_1^3(u) &= -2u^3 + 3u^2, \\ H_2^3(u) &= u^3 - 2u^2 + u, \\ H_3^3(u) &= u^3 - u^2.\end{aligned}$$

Ekkor a görbe felírható a Hermite-alappolinomok segítségével:

$$\mathbf{s}(u) = \mathbf{p}_0 H_0^3(u) + \mathbf{p}_1 H_1^3(u) + \mathbf{t}_0 H_2^3(u) + \mathbf{t}_1 H_3^3(u).$$

Az előbbi összefüggés alapján az Hermite-görbét úgy tekinthetjük, mint a kontrolladatok súlyozott összege, ahol a súly- vagy bázisfüggvények az Hermite-alappolinomok.

Az egységesebb szemléletmód miatt felírhatjuk a görbét mátrix alakban is:

$$\mathbf{s}(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{t}_0 \\ \mathbf{t}_1 \end{bmatrix}$$

Megjegyzés: ha a végpontbeli érintőket egyre nagyobb mértékben növeljük, miközben irányukat nem változtatjuk, akkor váratlanul kialakulhat hurok a görbén, azaz átmetszheti önmagát.

## 9.2. Approximáció

Az approximáció esetében nem kívánjuk meg, hogy az előállított görbe érintse (áthaladjon) az összes kontrolladaton (kontrollponton), hanem csak közelítse azokat tetszőleges mértékben.

Ebben a könyvben olyan approximációt keresünk, ahol adott a sík vagy a tér  $\mathbf{b}_0, \dots, \mathbf{b}_n$  pontjaihoz keressük azt az  $\mathbf{s}(u)$  görbét, melyhez találhatóak olyan  $F_0, F_1, \dots, F_n$  súly- vagy bázisfüggvények, hogy

$$\mathbf{s}(u) = \sum_{i=0}^n \mathbf{b}_i F_i(u), \quad u \in [a, b].$$

## 9.3. Bézier-görbe

Az eljárást Pierre Bézier francia mérnök publikálta először 1962-ben, aki az autótérvezésben alkalmazta a közelítő görbéket. Paul de Casteljau 1959-ben már kifejlesztette azt az algoritmust, amelyet a mai napig is használunk a Bézier-görbék előállítására.

Keressünk olyan görbét, amely a megadott pontokat közelíti (approximálja) az előre megadott sorrendben és nem érinti (nem interpolálja) a pontokat, vagy legalábbis nem mindegyiket.

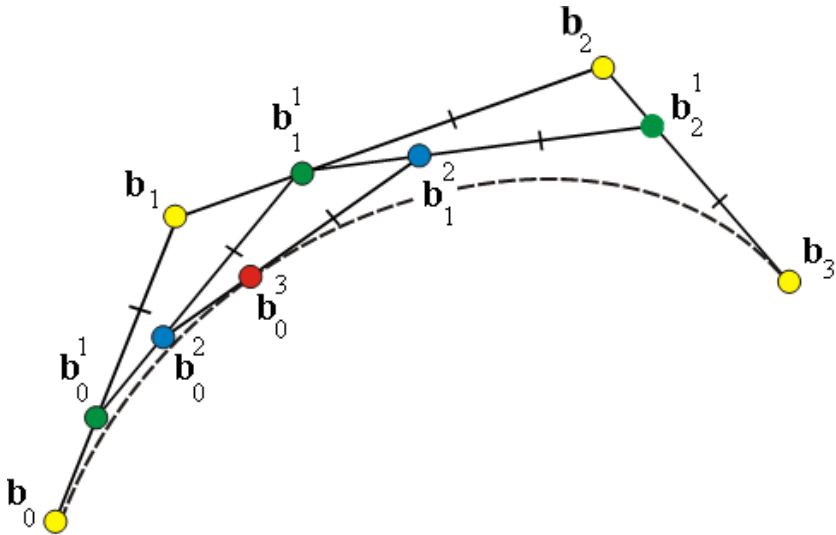
### 9.3.1. A de Casteljaú-algoritmus

Adottak a sík vagy a tér  $\mathbf{b}_0, \dots, \mathbf{b}_n$  pontjai és  $u \in [0,1]$  paramétertartomány. A megadott pontokat kontrollpontoknak, az általuk meghatározott poligont kontrollpoligonnak nevezzük. Legyen

$$\mathbf{b}_i^0 = \mathbf{b}_i \quad (i = 0, 1, \dots, n)$$

$$\mathbf{b}_i^r(u) = (1-u)\mathbf{b}_i^{r-1}(u) + u\mathbf{b}_{i+1}^{r-1}(u), \quad \begin{cases} r = 1, \dots, n \\ i = 0, 1, \dots, n-r. \end{cases}$$

A második egyenlet jól ismert, hiszen a szakasz osztópontjának a meghatározására szolgál. Most már elvégezhető a szerkesztés, ahol az így meghatározott  $\mathbf{b}_0^n(u)$  pont a Bézier-görbe  $u$  paraméterhez tartozó pontja lesz.



9.2. ábra.  $u = \frac{1}{3}$  paraméterhez tartozó  $\mathbf{b}_0^3(u)$  Bézier-görbepont szerkesztése de Casteljaú-algoritmussal.

### 9.3.2. A Bézier-görbe előállítás Bernstei-polinommal

A Bézier-görbe polinominális előállításához segítségül hívjuk a *Bernstei polinom*-ot:

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i},$$

ahol a binominális együtthatók a következőképpen adóttak

$$\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!} & \text{ha } 0 \leq i \leq n \\ 0 & \text{egyébként.} \end{cases}$$

Ezekután az

$$\mathbf{s}(u) = \sum_{i=0}^n \mathbf{b}_i B_i^n(u), \quad u \in [0,1]$$

a Bézier-görbe polinominális alakja. Láthatjuk, hogy a Hermite-görbéhez hasonlóan a kontrolladatokat, jelen esetben a kontrollpontok súlyozott összegéről van szó.

### 9.3.3. A Bézier-görbe néhány tulajdonsága

A Bézier-görbe kontrollpontjai affin transzformációjával szemben invariáns. Ez következik a de Casteljau-féle előállításból. Ezen tulajdonságot kihasználva, a görbe affin transzformációja (Pl.:eltolás, elforgatás, tükrözés, skálázás, párhuzamos vetítés) esetén elég a kontrollpontokra végrehajtani a transzformációt, mivel a transzformált pontok által meghatározott Bézier-görbe megegyezik az eredeti görbe transzformáltjával.

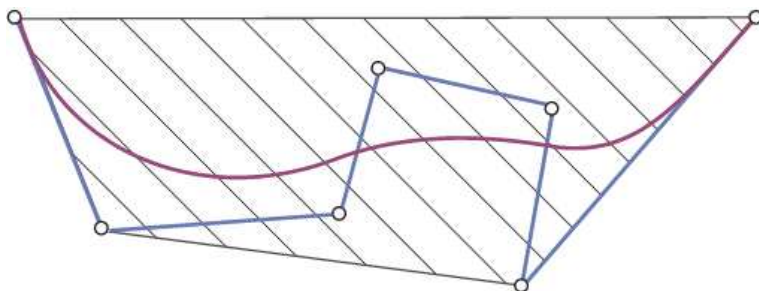
Ha  $u \in [0,1]$ , akkor a Bézier görbe kontrollpontjainak konvex burkán belül van.

A Bézier-görbe az első és utolsó kontrollponton áthalad.

A Bézier-görbe „szimmetrikus”, azaz a  $\mathbf{b}_0, \dots, \mathbf{b}_n$  és a  $\mathbf{b}_n, \dots, \mathbf{b}_0$  pontok ugyanazt a görbét állítják elő.

Ha  $u \in [0,1]$  akkor a Bézier-görbe kezdő- és végérintője:

$$\dot{\mathbf{b}}(0) = n(\mathbf{b}_1 - \mathbf{b}_0), \quad \dot{\mathbf{b}}(1) = n(\mathbf{b}_n - \mathbf{b}_{n-1}).$$



9.3. ábra. A Bézier-görbe kontrollpontjainak konvex burkán belül marad

Tehát a kezdő és a végpontban az érintők tartó egyenese a kontrollpoligon oldalai.

A görbe globálisan változtatható, azaz, ha megváltoztatjuk egy kontrollpontjának a helyzetét, akkor az egész görbe alakváltozáson megy keresztül. Bebizonyítható a Bernstein-polinomok tulajdonsága alapján, hogy a  $\mathbf{b}_i$  kontrollpontnak a  $u = \frac{i}{n}$  paraméterértéknél van legnagyobb hatása a görbe alakjára. Ez utóbbi tulajdonságot nevezik úgy, hogy a görbe pszeudolokálisan változtatható. Tehát a Bézier-görbe alakváltozása jól prognosztizálható.

A polinominális előállításból jól látható, hogy  $\mathbf{b}_0, \dots, \mathbf{b}_n$  pontok, azaz  $n + 1$  db pont esetén  $n$ -edfokú approximáló görbét kapunk, azaz a kontrollpontok számának növekedésével arányosan nő a poligon fokszáma is.

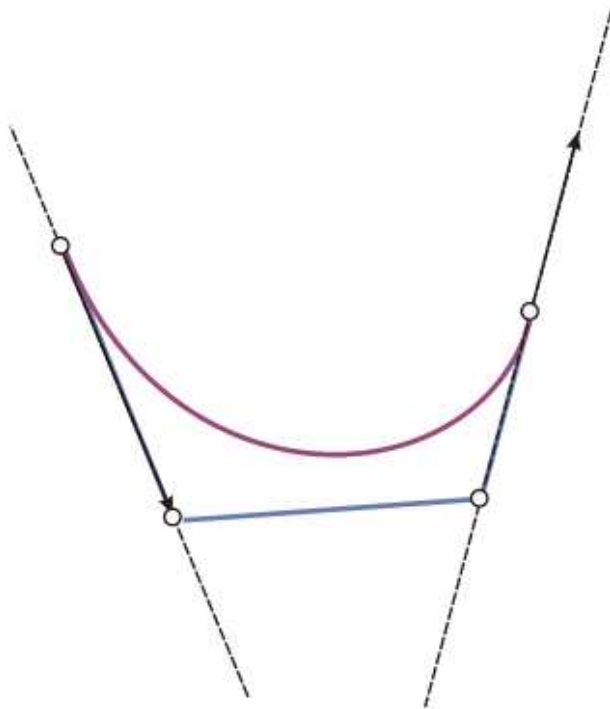
### 9.3.4. Harmadfokú Bézier-görbék

Adjuk meg az  $n = 3$  esetén  $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$  kontrollpontokat közelítő Bézier-görbe polinominális alakját a harmadfokú Bernstein-polinomok segítségével:

$$s(u) = \sum_{i=0}^3 \mathbf{b}_i B_i^3(u) = \mathbf{b}_0 B_0^3(u) + \mathbf{b}_1 B_1^3(u) + \mathbf{b}_2 B_2^3(u) + \mathbf{b}_3 B_3^3(u), \quad u \in [0,1].$$

Ahol a harmadfokú Bernstein-polinomok

$$B_i^3(u) = \binom{3}{i} u^i (1-u)^{3-i}, \quad i = 0, \dots, 3,$$



9.4. ábra. A Bézier-görbe és érintői  $n = 3$  esetén az érintővektorok harmadát felmérve.

azaz

$$B_0^3(u) = \binom{3}{0} u^0 (1-u)^3 = (1-u)^3$$

$$B_1^3(u) = \binom{3}{1} u^1 (1-u)^2 = 3u(1-u)^2$$

$$B_2^3(u) = \binom{3}{2} u^2 (1-u)^1 = 3u^2(1-u)$$

$$B_3^3(u) = \binom{3}{3} u^3 (1-u)^0 = u^3,$$

tehát

$$\begin{aligned} \mathbf{s}(u) &= \mathbf{b}_0(1-u)^3 + \mathbf{b}_1 3u(1-u)^2 + \mathbf{b}_2 3u^2(1-u) + \mathbf{b}_3 u^3 = \\ &= \mathbf{b}_0(-u^3 + 3u^2 - 3u + 1) + \mathbf{b}_1(3u^3 - 6u^2 + 3u) + \end{aligned}$$



$$+\mathbf{b}_2(-3u^3 + 3u^2) + \mathbf{b}_3u^3.$$

A görbe érintője:

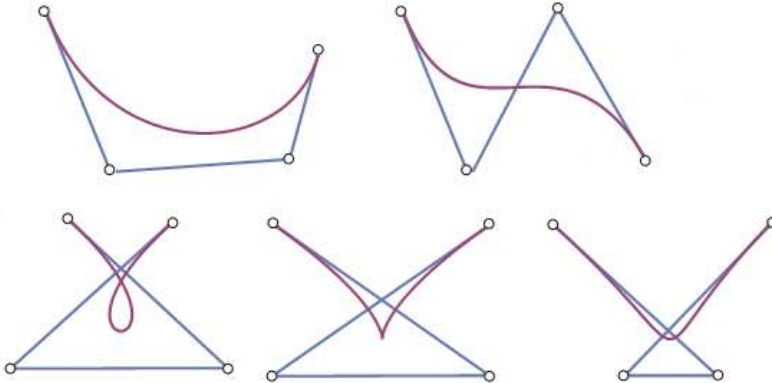
$$\dot{\mathbf{b}}(0) = 3(\mathbf{b}_1 - \mathbf{b}_0), \dot{\mathbf{b}}(1) = 3(\mathbf{b}_3 - \mathbf{b}_2)$$

Ezután könnyedén megtalálhatjuk az Hermite-görbe és a Bézier-görbe közötti kapcsolatot. Ha az Hermite-görbe a  $\mathbf{p}_0$  és  $\mathbf{p}_1$  pontokkal, valamint a  $\mathbf{t}_0$  és  $\mathbf{t}_1$  érintő vektorokkal van meghatározva, akkor a kezdő és végpontbeli érintőknek meg kell egyezniük a Bézier-görbe érintőivel, azaz  $\mathbf{t}_0 = \dot{\mathbf{b}}(0)$ ,  $\mathbf{t}_1 = \dot{\mathbf{b}}(1)$ , továbbá a kezdő és végpontok is egybeesnek.

Tehát az Hermite-görbével megegyező Bézier-görbe kontrollpontjai:

$$\mathbf{b}_0 = \mathbf{p}_0, \mathbf{b}_1 = \mathbf{p}_0 + \frac{1}{3}\mathbf{t}_0, \mathbf{b}_2 = \mathbf{p}_1 - \frac{1}{3}\mathbf{t}_1, \mathbf{b}_3 = \mathbf{p}_1.$$

A harmadfokú Bézier-görbék (s természetesen az Hermite-görbék is) változatos képet mutathatnak. Ha a harmadfokú Bézier-görbe kontrollpontjai



9.5. ábra. Harmadfokú Bézier-görbék

nincsenek egy síkban, akkor a Bézier-görbe térgörbe lesz, azaz nem találunk olyan síkot amelyre a görbe minden pontja illeszkedne. A három kontrollpont esetén a másodfokú Bézier-görbe már egy jól ismert síkgörbe, parabola lesz.

### 9.3.5. Kapcsolódó Bézier-görbék

Az előző fejezetben láttuk, hogyan milyen kapcsolat van az Hermite-görbe és a Bézier-görbe között. Ha több mint két pontot szeretnénk összekötni interpoláló görbével, akkor kézenfekvő megoldás a kapcsolódó harmadfokú Bézier-görbék alkalmazása, amelyek természetesen Hermite-görbék is lehetnek. Kapcsolódó görbévek használata igen gyakori a modellezésben. Ilyen esetekben a csatlakozásnál megadjuk a folytonosság mértékét. Ha az  $\mathbf{a}(u), u \in [0,1]$  és  $\mathbf{b}(u), u \in [0,1]$  két csatlakozó görbe amely négy-négy kontrollponttal adott:  $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$  és  $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ .

A kapcsolódásnál megkövetelhetünk nulladrendű  $C^0$ , elsőrendű  $C^1$ , illetve másodrendű  $C^2$  folytonosságot. Általánosságban azt mondhatjuk, hogy két csatlakozógörbe kapcsolódása  $C^n$  folytonos, ha az egyik görbe deriváltjai a végpontjában megegyeznek a másik görbe deriváltjaival a kezdőpontban, az  $n$ . deriváltig bezárólag. A matematikai folytonosság vagy parametrikus folytonosság mellett létezik geometriai folytonosság is. Pl.  $G^0$  megegyezik  $C^0$ -lal, és  $G^1$  folytonosan kapcsolódik két görbe, ha az érintők a kapcsolódási pontban egy irányba néznek, de a hosszuk nem feltétlenül egyezik meg, azaz egyik érintő skalárszorosa a másiknak:  $\dot{\mathbf{a}}(1) = \omega \dot{\mathbf{b}}(0), \omega > 0$

A nulladrendű folytonossághoz elegendő, ha a csatlakozáskor keletkező görbe megrajzolható anélkül, hogy a ceruzánkat felemelnénk. A mi esetünkben ez akkor teljesül, ha az első görbe végpontja megegyezik a második görbe kezdőpontjával, azaz:  $\mathbf{a}(1) = \mathbf{b}(0)$  és mivel ezen görbepontok megegyeznek a megfelelő kontrollpontokkal, hiszen a Bézier-görbe a végpontokat interpolálja, ezért  $\mathbf{a}_3 = \mathbf{b}_0$  kell, hogy teljesüljön.

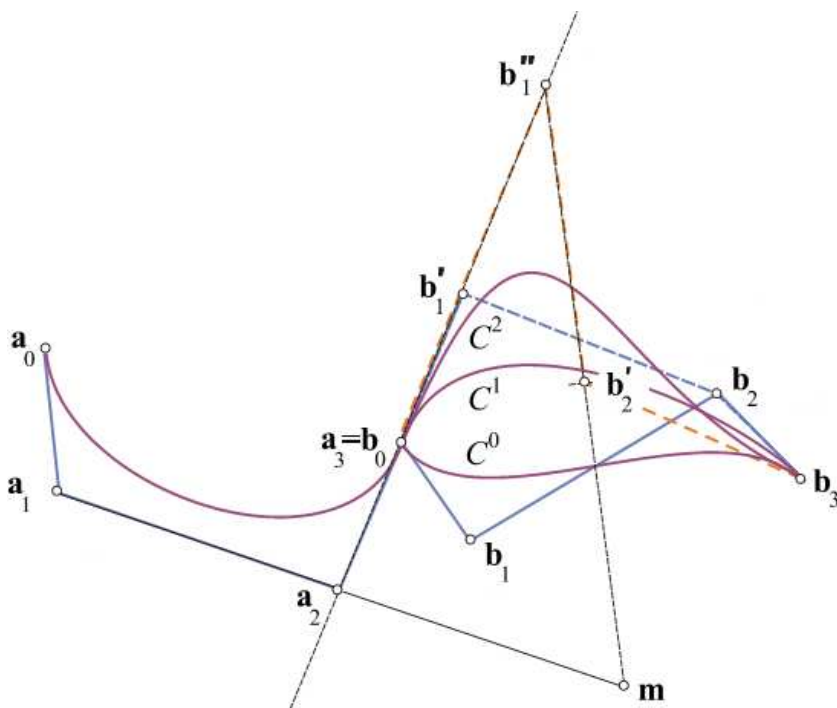
Az elsőrendű,  $C^1$  folytonossághoz az érintőknek kell megegyezniük, azaz  $\dot{\mathbf{a}}(1) = \dot{\mathbf{b}}(0)$  kell, hogy teljesüljön. Ez a kontrollpontokra az  $(\mathbf{a}_3 - \mathbf{a}_2) = (\mathbf{b}_1 - \mathbf{b}_0)$  feltételt jelenti, azaz, amellet, hogy a két szegmens kezdő- és végpontja megegyezik, az  $\mathbf{a}_2, \mathbf{a}_3 = \mathbf{b}_0, \mathbf{b}_1$  pontoknak egy egyenesre kell illeszkedniük és az  $\mathbf{a}_3$  pontnak feleznie kell az  $\mathbf{a}_2\mathbf{b}_1$  szakaszt.

A másodrendben folytonos kapcsolódáshoz a fenti feltételeken kívül a második deriváltaknak is meg kell egyezniük, azaz  $\ddot{\mathbf{a}}(1) = \ddot{\mathbf{b}}(0)$  Ez a kontrollpontokra nézve a következőt jelenti:

$$((\mathbf{a}_3 - \mathbf{a}_2) - (\mathbf{a}_2 - \mathbf{a}_1)) = ((\mathbf{b}_2 - \mathbf{b}_1) - (\mathbf{b}_1 - \mathbf{b}_0)),$$

ami geometriai szempontból azt jelenti, hogy az  $\mathbf{a}_1\mathbf{a}_2$  egyenes és a  $\mathbf{b}_1\mathbf{b}_2$  egyenes  $\mathbf{m}$  metszéspontjára teljesül, hogy  $\mathbf{a}_2$  felezi az  $\mathbf{a}_1\mathbf{m}$  szakaszt,  $\mathbf{b}_1$  pedig felezi az  $\mathbf{mb}_2$  szakaszt.

A gyakorlatban  $C^2$  folytonosságú kapcsolat elégséges, pl. animáció esetén a mozgó kamera által készített felvétel akkor lesz valóság-hű, ha a második derivált is megegyezik. Gondoljunk arra, hogy az út/idő függvény második deriváltja a gyorsulást adja, tehát ha a kapcsolódási pontban megváltozik a gyorsulás, akkor szaggatott felvételt kapunk.

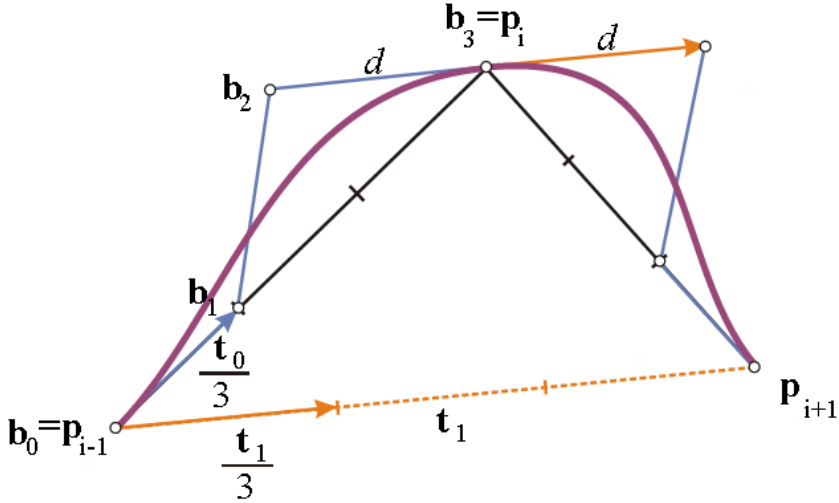


9.6. ábra.  $C^0$ ,  $C^1$  és  $C^2$  folytonosan kapcsolódó Bézier-görbék

### 9.3.6. Cardinal spline

A **cardinal spline** egy a kontrollpontokat interpoláló, azaz az előre megadott pontokon adott sorrendben áthaladó görbe, tulajdonképpen elsőrend-

ben folytonosan csatlakozó harmadfokú (kubikus) Hermite-görbék sorozata. Mivel az előző fejezetben már kimutattuk az Hermite- és a Bézier-görbe közötti kapcsolatot, ezért a cardinal spline megadhatjuk harmadfokú  $C^1$  folytonosan kapcsolódó Bézier-görbék sorozataként is. Járjunk el az utóbbi módon!



9.7. ábra. 3 pontot interpoláló  $C^1$  folytonosan csatlakozó Bézier-görbék

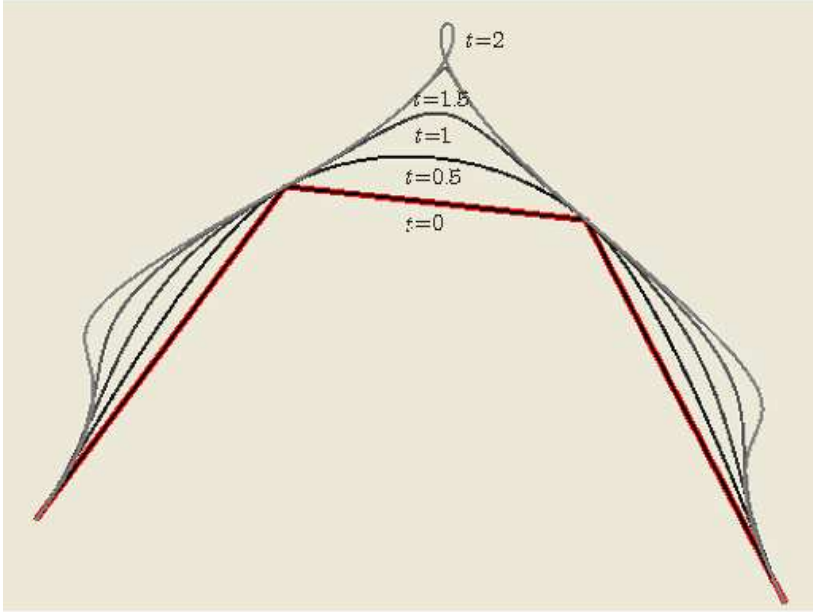
Az  $i$ . Bézier szegmens kezdő és végpontja a szomszédos  $\mathbf{p}_i$  és  $\mathbf{p}_{i+1}$  pontok. A görbe deriváltja egy közbülső  $\mathbf{p}_i$  pontban párhuzamos a  $\mathbf{p}_{i-1}$   $\mathbf{p}_{i+1}$  egyenessel, azaz

$$\dot{\mathbf{s}}_i(u) = t(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}), t \geq 0$$

Ne felejtjük el, hogy a harmadrendű Bézier-görbe négy kontrollpontjával, az Hermite-görbét pedig a kezdő- és a végpontjával, valamint a kezdő- és a végpontban húzott érintővel adjuk meg. Az  $i$ . Bézier-görbe szegmens kezdő és végpontja  $\mathbf{b}_0 = \mathbf{p}_i$ ,  $\mathbf{b}_3 = \mathbf{p}_{i+1}$ . Mivel a görbe érintője:

$$\dot{\mathbf{b}}(0) = 3(\mathbf{b}_1 - \mathbf{b}_0), \quad \dot{\mathbf{b}}(1) = 3(\mathbf{b}_3 - \mathbf{b}_2).$$

ezért minden kontrollpont minden szegmens esetén most már meghatározható. Az érintők meghatározásánál használhatunk egy  $t \geq 0$  tenziós értéket, amely az érintők nagyságát befolyásolja. Ha  $t = 0.5$  akkor a Catmul-Rom spline-t, ha  $t = 0$  akkor a kontrollpontokat összekötő poligont kapjuk meg.



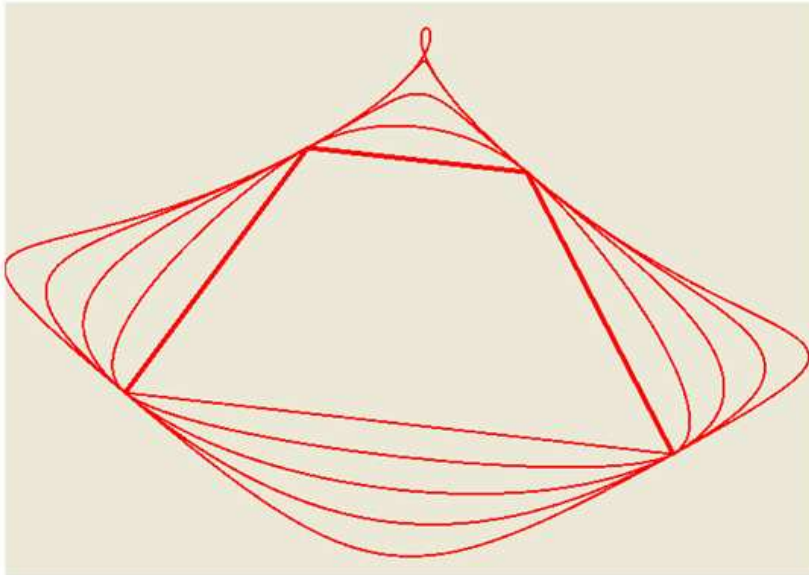
9.8. ábra.  $t$  tenziós érték hatása a görbe alakjára

Zárt görbe estén az érintő a kezdő és a végpontban is az

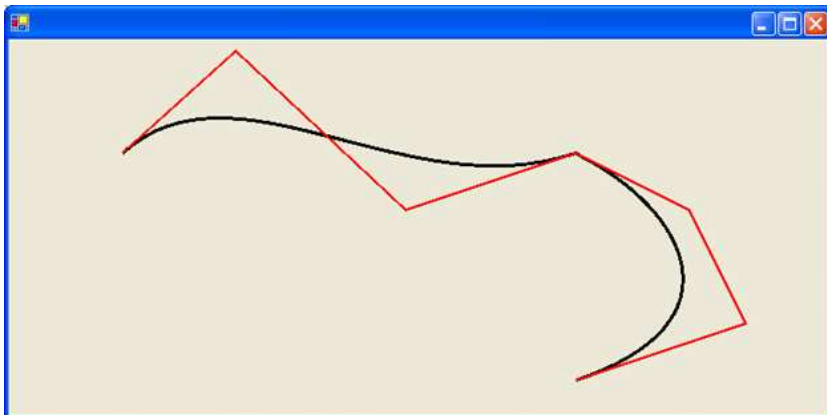
$$\dot{\mathbf{s}}_i(u) = t(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}), t \geq 0$$

általános képlettel határozható meg.

Visual Studióban GDI+ segítségével lehetőségünk van cardinal spline előállítására **DrawCurve** metódussal. Mivel a GDI+ kihasználja, hogy a cardinal spline csatlakozó Bézier-görbékéből áll, ezért szükség van egy Bézier-görbét előállító metódusra is. A **DrawBezier** négy kontroll pontra illetve közelítő görbét, míg a **DrawBeziers** pedig  $C^0$  folytonosan kapcsolódó Bézier-görbéket rajzol.



9.9. ábra. Zárt görbék különböző  $t$  értékeknél



9.10. ábra. 7 pont esetén a **DrawBeziers** metódus futási eredménye

## 10. B-spline görbe és felület

Ebben a fejezetben áttekintjük a számítógépes modellezés talán az egyik legnépszerűbb görbéjét, a B-spline-t. Irodalomként felhasználjuk egyrészt G. Farin[21] és Juhász Imre [48], [41] munkáit, másrészt jól használhatónak tartom még F. Yamaguchi [86], Rogers és Adams [71], és Piegl és Tiller [63] könyvét. Ez utóbbit sokat alapl műnek tekintik a témában. A B-spline görbék alapjául szolgáló úgynevezett B-spline alapfüggvények már régóta ismertek. N. I. Lobacsevszkij már a XIX. században vizsgálta egy speciális esetét (speciális csomóérték sorozat fölött definiálta, lásd Rényi [69], 165. o.), J. Schoenberg 1946-ban statisztikai adatok simítására használta, az ő cikkétől [74] származtatjuk a spline approximáció modern elméletének a kezdetét. A B-spline görbék tervezése a hetvenes évek elején válik használhatóvá. Ehhez hozzájárult a B-spline alapfüggvények rekurzív tulajdonságát felfedező C. de Boor [10], M. Cox [17] és L. Mansfield, továbbá W. Gordon és R. Riesenfeld [26] munkásságának köszönhetjük, hogy a csak approximációs szempontból vizsgált függvényeket paraméteres görbék előállítására, görbék tervezésére is használják. Jelentős munkásságot fejtett ki ebben a témakörben még W. Boehm [9] is.

A B-spline görbe a Bézier görbéhez hasonlóan approximáló görbe. A tulajdonságok tárgyalásánál látni fogjuk, hogy a B-spline esetén jóval több szabadsági fokkal rendelkezünk a görbe alakjának változtatása tekintetében. A B-spline görbe használatának előnye még, hogy kevesebb kontrollpontra van szükség, mint a Bézier görbe esetében, azaz kevesebb adatot kell tárolni a számítógépen, és mint látni fogjuk a Bézier görbe egy speciális esete a B-spline görbének.

### 10.1. Normalizált B-spline alapfüggvény

**10.1. definíció.** Tekintsük az  $u_i \leq u_{i+1}$ ,  $u_i \in \mathbb{R}$  ( $i = 0, \dots, n + k$ ) skalárokat. Az

$$N_i^1(u) = \begin{cases} 1, & \text{ha } u_i \leq u < u_{i+1}; \\ 0 & \text{egyébként.} \end{cases}$$

$$N_i^k(u) = \frac{u - u_i}{u_{i+k-1} - u_i} N_i^{k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} N_{i+1}^{k-1}(u) \quad (10.1)$$

rekurzióval definiált függvényt normalizált B-spline alapfüggvénynek, az  $u_i$  skalárokat pedig csomóértékeknek (knot values) vagy csomóvektornak (knot vector) nevezzük. Az előforduló  $\frac{0}{0}$ -t definíció szerint 0-nak tekintjük.

Az  $N_i^k(u)$  normalizált B-spline alapfüggvény egy legfeljebb  $k - 1$ -edfokú polinom (a rendje  $k$ ). A  $\frac{0}{0}$  akkor fordulhat elő, ha a szomszédos csomóértékek egybeesnek.

Az  $N_i^k(u)$  normalizált B-spline alapfüggvény néhány fontos tulajdonsága, melyet fokszám szerinti teljes indukcióval bizonyíthatunk:

- $N_i^k(u) = 0$ , ha  $u \notin [u_i, u_{i+k}]$ , local support
- $N_i^k(u) \geq 0$ ,  $\forall u$
- $\sum_{i=0}^n N_i^k(u) \equiv 1$ ,  $\forall k$  és  $u \in [u_{k-1}, u_{n+1}]$ , ezen tulajdonság teljesülése miatt nevezzük hívják normálnak, ahol az alapfüggvények a  $k - 1$ -ed fokú polinomtér normált bázisát alkotják.
- $N_i^k(u)$   $k - 2$ -ször folytonosan differenciálható

10.2. *megjegyzés.* Egyenlő közötti csomóértékek esetén ( $u_0 = 0, u_{i+1} = u_i + c, c \in \mathbb{R}$ ) legyen  $u_i = i$ , ekkor a rekurzív definíció átalakul,

$$N_i^k(u) = \frac{u - i}{k - 1} N_i^{k-1}(u) + \frac{i + k - u}{k - 1} N_{i+1}^{k-1}(u),$$

és  $N_i^k$ -t uniform B-spline bázisnak nevezzük. Ha a B-spline nem egyenlőközű csomóértékek fölött van megadva, akkor non-uniformnak, ha a csomóértékek növekedése, – azaz a differenciája – állandó vagy 0, akkor open uniform B-spline bázisnak nevezzük.

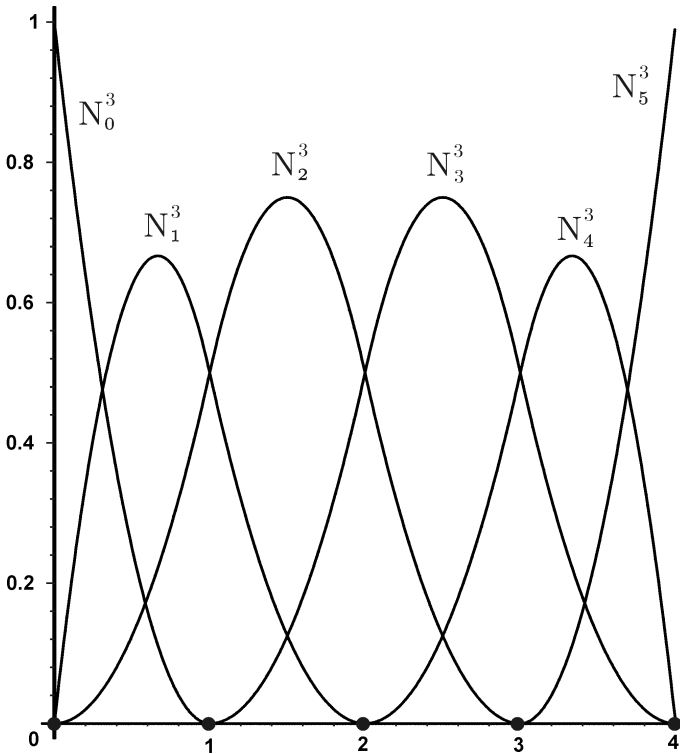
10.3. *megjegyzés.* Tekintsünk egy példát a B-spline alapfüggvények előállítására.



sára, open uniform csomóérték megadással:

$$u_i = \begin{cases} 0, & \text{ha } 0 \leq i < k \\ i - k + 1 & \text{ha } k \leq i \leq n \\ n - k + 2 & \text{ha } n < i \leq n + k \end{cases} \quad (10.2)$$

Pl: Legyen  $n = 5$ ,  $k = 3$ , ekkor a csomóértékek  $u_0, \dots, u_8 = 0, 0, 0, 1, 2, 3, 4, 4, 4$ .



10.1. ábra. B-spline alapfüggvények open uniform esetben

A csomóértékek ilyen megadása esetén (a kezdő és a végpontban  $k$ -szoros a csomóérték) biztosítjuk, hogy a B-spline görbe interpolálja a végpontokban a kontrollpontokat, amely egyébként nem feltétlenül következik be. Legyen  $u \in [0, n - k + 2)$ , ekkor már csak  $k$ -tól és a kontrollpontoktól függ a B-spline görbe alakja

## 10.2. B-spline görbe

A Bézier görbéhez hasonlóan határozzuk meg a B-spline görbét is. A Bézier görbe fokszáma és a kontrollpontok száma függ egymástól, itt a szituáció különböző.

**10.4. definíció.** Azt a görbét, amely

$$\mathbf{r}(u) = \sum_{i=0}^n \mathbf{d}_i N_i^k(u), 2 \leq k \leq n+1,$$

alakban felírható,  $k-1$ -ed fokú (vagy  $k$ -adrendű) B-spline görbének nevezzük, ahol  $N_i^k(u)$  a  $k-1$ -ed fokú normalizált B-spline alapfüggvény

$$\mathbf{U} = (u_0, u_1, \dots, u_n, u_{n+1}, \dots, u_{n+k})$$

csomóvektorral. A  $\mathbf{d}_i$  ( $i = 0, \dots, n$ ) pontokat kontrollpontoknak vagy de Boor-pontoknak, az általuk meghatározott poligont pedig kontroll- vagy de Boor-poligonnak nevezzük.

### 10.2.1. Tulajdonságok:

- A B-spline görbe lokálisan változtatható, azaz valamely kontrollpont helyének a megváltoztatása nem eredményezi (szemben a Bézier görbével) a teljes görbe alakjának megváltozását.

Ez abból következik, hogy ha az  $\mathbf{r}(u) = \sum_{i=0}^n \mathbf{d}_i N_i^k(u)$ , a B-spline görbe csomóvektora  $\mathbf{U} = (u_0, u_1, \dots, u_n, u_{n+1}, \dots, u_{n+k})$ , akkor  $\mathbf{d}_i$  kontrollpontnak csak  $u \in [u_i, u_{i+k}]$  estén van befolyása a görbe alakjára.

- A B-spline görbe (szemben a Bézier görbével) tartalmazhat egyenes szakaszt akkor is, ha nem minden kontrollpontja kollineáris.
- Egy  $k-1$ -ed fokú B-spline görbe bármely pontja a görbe legfeljebb  $k$  darab kontrollpontjának konvex burkában van. Ez a konvex burkok tulajdonság jóval szigorúbb, mint a Bézier görbe esetén, mivel itt a görbe a konvex burkok uniójában halad.

- $k$ -szoros kontrollpont: ha  $\mathbf{d}_j = \dots = \mathbf{d}_{j+k-1}$ , akkor  $\mathbf{d}_j$  a görbén van. tehát a (10.2) szerinti megadás biztosítja, hogy a görbe interpolálja a kezdő és a végpontot.
- $k$ -szoros csomóérték esetén,  $u_{j+1} = \dots = u_{j+k} = \bar{u}$ , valamely  $j \in [0, \dots, n]$ , akkor a görbe áthalad  $\mathbf{d}_j$  kontrollponton
- Érvényes az affin invariancia tulajdonság. Ha a görbét affin transzformációnak akarjuk alávetni, akkor elég a kontrollpontokat transzformálni, és nem kell a görbe összes pontját.
- $k$ -ad rendű nyílt B-spline görbe esetén, ha  $k = n - 1$  és a csomóvektor  $\mathbf{U}(u_0 = u_1 = \dots = u_{k-1}, u_k, u_{k+1}, \dots, u_n, u_{n+1} = u_{n+2} = \dots = u_{n+k})$  akkor a B-spline görbe amellet hogy interpolálja a kontrollpoligon kezdő és a végpontját még Bézier görbére is redukálódik.
- Ha  $\mathbf{d}_j = \dots = \mathbf{d}_{j+k}$  ( $(k - 1)$  db) pontok kollineárisak, és nem mind esik egy pontba, akkor a bspline görbe érinti a poligont. Ebből következik, hogy  $k = 3$  esetén a B-spline görbe érinti a kontrollpoligon minden szegmensét.
- A görbe legalább annyiszor metsz egy hipersíkot (2D-ben egyenest), mint a hipersík a kontrollpoligont. (Variation-diminishing property)
- Zárt (vagy periodikus) B-spline görbe a következő két módon definiálható a  $\mathbf{d}_i$  ( $i = 0, \dots, n$ ) kontrollpontokra. Az egyik lehetőség, hogy a kontrollpontokat ciklikusan ismételjük, azaz

$$\mathbf{d}_j = \mathbf{d}_{j \bmod (n+1)} \quad (j = 0, 1, \dots, n + k - 1),$$

Ebben az esetben a görbe megadható a következő módon:

$$\mathbf{r}(u) = \sum_{j=0}^{n+k-1} \mathbf{d}_j N_j^k(u), \quad u \in [u_{k-2}, u_{k+2}].$$

A másik lehetőség, hogy

$$\mathbf{d}_0 = \mathbf{d}_{n+1},$$

és periódikusan kiterjesztjük a csomóértékeket

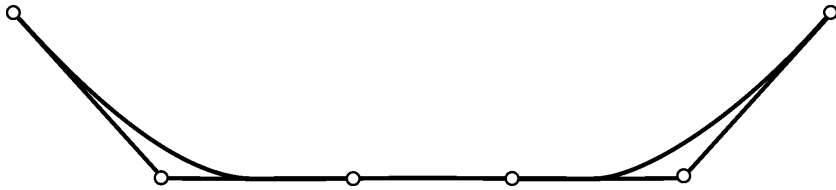
$$u_{n+1} \doteq u_0, u_{n+2} \doteq u_{n+1} + (u_1 - u_0) \doteq u_1, \dots$$

Ez a következő csomóvektorhoz vezet:

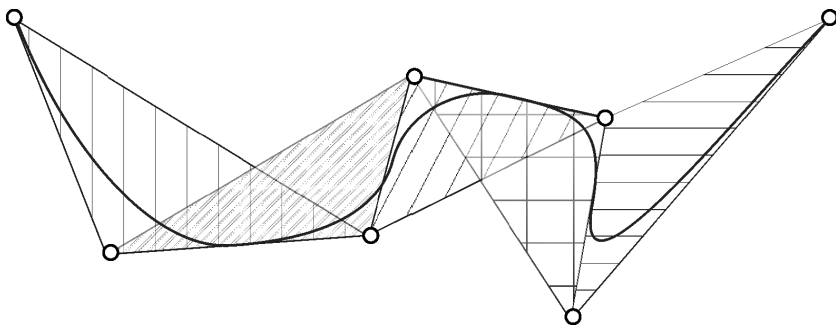
$$\mathbf{U} = (u_0, u_1, \dots, u_n, u_{n+1} \doteq u_0, u_{n+2} \doteq u_1, \dots, u_{n+k} \doteq u_{k-1}). \quad (10.3)$$

Ebben az esetben a görbe így írható fel:

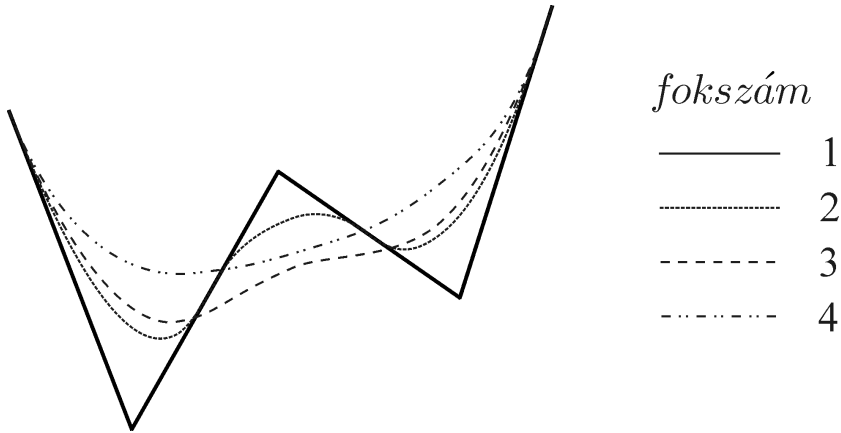
$$\mathbf{r}(u) = \sum_{j=0}^n \mathbf{d}_j N_j^k(u), \quad u \in [u_0, u_{n+1}].$$



10.2. ábra. Egyenes szakaszt tartalmazó harmadfokú B-spline görbe



10.3. ábra. Egy másodfokú ( $k = 3$ ) B-spline görbe íveinek konvex burka



10.4. ábra. Uniform B-spline görbék

### 10.3. B-spline görbék kapcsolódása

Kapcsolódó görbévek használata igen gyakori a modellezésben. Ilyen esetekben a csatlakozásnál megadjuk a folytonosság mértékét. Ha  $\mathbf{r}_0(u)$ ,  $u \in [u_0, u_1]$  és  $\mathbf{r}_1(u)$ ,  $u \in [u_1, u_2]$  két csatlakozó görbe legalább  $n$ -szer folytonosan differenciálható és az  $u = u_1$  paraméternél igaz, hogy

$$\frac{d^j}{du^j} \mathbf{r}_0(u_1) = \frac{d^j}{du^j} \mathbf{r}_1(u_1) \quad (j = 0, \dots, n),$$

akkor azt mondjuk, hogy  $n$ -ed renben folytonosan kapcsolódnak egymáshoz ( $n$ -edrenben érintkeznek), ezt röviden  $C^n$ -folytonosságnak nevezzük. Az előbbi definíciót gyakran matematikai folytonosságnak nevezzük, emellett beszélhetünk geometriai folytonosságról is. Ha

$$\dot{\mathbf{r}}_0(u_1) = \omega_1 \dot{\mathbf{r}}_1(u_1), \quad \omega_1 > 0,$$

akkor  $G^1$  folytonosságról (érintő folytonosság) beszélünk, tehát csak az érintők iránya kell, hogy megegyezzen a nagysága nem. Ha

$$\ddot{\mathbf{r}}_0(u_1) = \omega_1^2 \ddot{\mathbf{r}}_1(u_1) + \omega_2 \dot{\mathbf{r}}_1(u_1),$$

akkor  $G^2$  folytonosságról (görbület folytonosság) beszélünk, tehát az csatlakozási pontban a görbület megegyezik. Ha

$$\ddot{\mathbf{r}}_0(u_1) = \omega_1^3 \ddot{\mathbf{r}}_1(u_1) + \omega_2 \ddot{\mathbf{r}}_1(u_1) + \omega_3 \dot{\mathbf{r}}_1(u_1),$$

akkor  $G^3$  folytonosságról (torzió folytonosság) beszélünk, tehát mindkét görbe esetében a csatlakozási pontban a második deriváltak segítségével meghatározható torzió megegyezik.

B-spline görbék egymáshoz kapcsolódó ívekből áll. A kapcsolódási pontok a csomóértékekhez tartozó pontokban vannak, tehát ezekben a pontokban vizsgálhatjuk a kapcsolódás folytonosságát. Az

$$\mathbf{r}(u) = \sum_{j=0}^n \mathbf{d}_j N_j^k(u)$$

görbe

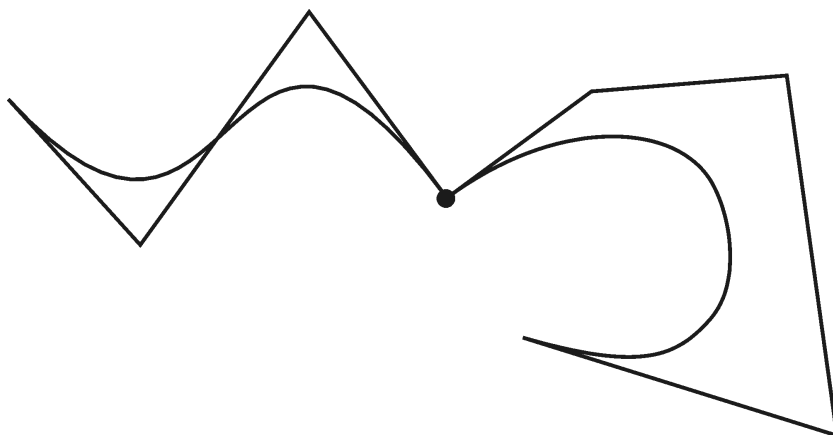
$$\mathbf{r}_i(u) = \sum_{j=i-k+1}^i \mathbf{d}_j N_j^k(u),$$

$$\mathbf{r}_{i-1}(u) = \sum_{j=i-k}^{i-1} \mathbf{d}_j N_j^k(u)$$

ívei  $k - 2$  -ed endben folytonosan kapcsolódnak egymáshoz az  $u = u_i$  csomóértéknél, ha  $k > 1$  es  $u_i$  multiplicitása 1. A csomóérték multiplicitásának növelése csökkenti a kapcsolódás folytonosságát, azaz, ha az  $u_i$  csomóérték multiplicitása  $m$ , akkor a folytonos kapcsolódás rendje  $k - 1 - m$ . Tehát a csomóérték multiplicitásának növelése olyan mértékben húzhatja a megfelelő kontrollpont felé a görbét, hogy akár ott csúcspont is kialakulhat.

#### 10.4. B-spline görbe előállítás

A görbe előállítására – az előbbi fejezet alapján – már tudunk készíteni programot. Mivel a polinominális alakban megadott rekurzív függvény számítása rosszul kondicionált, ezért javallott a Bézier görbénél tanult lineáris



10.5. ábra. Csúcs pont kialakulása a csomóérték multiplicitásának növelésével

műveleteket tartalmazó de Casteljau algoritmushoz hasonló Cox de Boor algoritmus használata.

#### 10.4.1. Cox-de Boor algoritmus

de Boor javasolt egy algoritmust arra, hogy B-spline görbe pontot állítsunk elő rekurzívan, lineáris műveletek segítségével. A (10.1) egyenletben definiált rekurzív formula átalakításával kezdjük:

$$\begin{aligned}
 r(u) &= \sum_{i=0}^n \mathbf{d}_i N_i^k(u) = \\
 &= \sum_{i=0}^n \mathbf{d}_i \frac{u - u_i}{u_{i+k-1} - u_i} N_i^{k-1}(u) + \sum_{i=0}^n \mathbf{d}_i \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} N_{i+1}^{k-1}(u).
 \end{aligned}$$

Indextranszformációt hajtunk végre a második tagon,  $i = i - 1$ , amely a következőt eredményezi

$$r(u) = \sum_{i=0}^{n+1} \frac{\mathbf{d}_i (u - u_i) + \mathbf{d}_{i-1} (u_{i+k-1} - u)}{u_{i+k-1} - u_i} N_i^{k-1}(u) =: \sum_{i=0}^n \mathbf{d}_i^{[1]} N_i^{k-1}(u)$$

ahol,  $\mathbf{d}_{-1} = 0$  és  $\mathbf{d}_{n+1} = 0$ . Az újraindexelést folytatva kapjuk

$$r(u) = \sum_{i=0}^{n+j} \mathbf{d}_i^{[j]}(u) N_i^{k-j}(u), \quad j = 1, \dots, k-1,$$

ha

$$\mathbf{d}_j^{[0]}(t_s) = \mathbf{d}_i \quad (i = 0, \dots, n)$$

akkor

$$\mathbf{d}_i^{[j]}(u) = (1 - \lambda) \mathbf{d}_{i-1}^{[j-1]}(u) + \lambda \mathbf{d}_i^{[j-1]}(u), \quad j > 0$$

konvex lineáris kombináció, ahol

$$\lambda = \frac{u - u_i}{t_{i+k-j} - u_i}.$$

Ha az algoritmust folytatjuk addig, hogy  $j = k - 1$  feltétel teljesül, akkor megkapjuk az  $N_r^1$  bázisfüggvényt, azaz  $u \in [u_r, u_{r+1}]$  esetén a függvényértéket

$$r(u) = \mathbf{d}_r^{k-1}(u).$$

Ezt a felosztási algoritmust *Cox-de Boor algoritmusnak* nevezzük. Ha az  $u_i$  és  $u_{i+k-1}$  ( $u_i < u_{i+k-1}$ ) multiplicitása  $k - 1$ , azaz

$$u_i = u_{i+1} = \dots = u_{i+k-2} = u_{i+k-1} = u_{i+k} = \dots = u_{i+2k-3},$$

akkor speciális esetként a de Casteljau-algoritmust kapjuk, következésképpen a B-spline görbe speciális eseteként a Bézier görbét. Bizonyítást a [48]-ban találjuk meg.

Figyelembe véve azt a tényt, hogy adott  $u \in [u_r, u_{r+1}]$  paraméter esetén minden  $N_k^i$  eltűnik kivéve azon bázisfüggvényt, ahol az indexekre igaz, hogy  $r - (k - 1) \leq i \leq r$ . A Cox- de Boor algoritmus a következő sémával adható meg:



$$\begin{array}{rcccc}
\mathbf{d}_{r-k+1} & = & \mathbf{d}_{r-k+1}^0 & & \\
\mathbf{d}_{r-k+2} & = & \mathbf{d}_{r-k+2}^0 & \mathbf{d}_{r-k+2}^1 & \mathbf{d}_{r-k+3}^2 & \dots \\
& & & & & \dots \\
& & & & & \mathbf{d}_{r-1}^{k-2} \\
& \vdots & & & \dots & \mathbf{d}_r^{k-1} = r(u). \\
& & & & & \mathbf{d}_r^{k-2} \\
& & & & \dots & \\
\mathbf{d}_{r-2} & = & \mathbf{d}_{r-2}^0 & & \mathbf{d}_{r-1}^2 & \\
& & & \mathbf{d}_{r-1}^1 & & \\
\mathbf{d}_{r-1} & = & \mathbf{d}_{r-1}^0 & & \mathbf{d}_r^2 & \\
& & & \mathbf{d}_r^1 & & \\
\mathbf{d}_r & = & \mathbf{d}_r^0 & & & 
\end{array}$$

Ez alapján a szerkesztés is könnyen elvégezhető, csak itt a szakaszok osztási aránya függ a csomóértékektől.

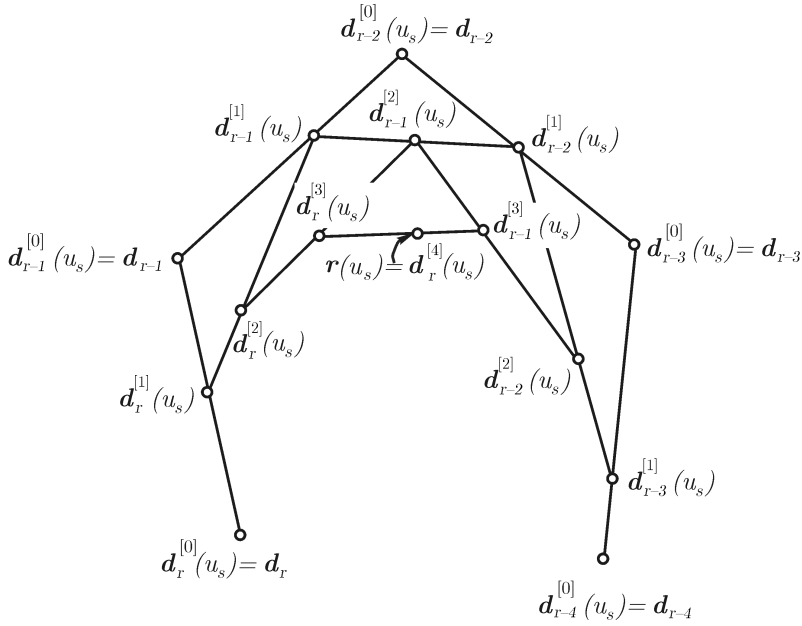
Mivel programozási szempontból nagyon fontos a Cox-de Boor algoritmus, ezért részletesen megadjuk az algoritmust:

**Adatok:**

- $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_n$ , ahol  $n \geq k - 1$ ,  $(n + 1)$  de Boor pont  $k$ , a B-spline görbe rendje (fokszám =  $k - 1$ )
- Zárt görbe esetén, a (10.3) szerint megadott csomóértékeket érdemes kiterjeszteni jobbra és balra is, azaz bevezetni az

$$\begin{aligned}
u_{-1} &= u_0 + (u_{n+1} - u_n), & u_{-2} &= u_{-1} + (u_n - u_{n-1}), \dots, \\
u_{n+2} &= u_{n+1} + (u_1 - u_0), & u_{n+3} &= u_{n+2} + (u_2 - u_1), \dots,
\end{aligned}$$

értékeket



10.6. ábra. B-spline görbepont szerkesztése de Boor algoritmussal,  $k = 5$

– Nyílt görbe esetén, a csomóértékek megadása legyen

$$\mathbf{U}(u_0 = u_1 = \dots = u_{k-1}, u_k, u_{k+1}, \dots, u_n, u_{n+1} = u_{n+2} = \dots = u_{n+k})$$

### A számítás menete:

Határozzuk meg a  $u = u_s$  paraméter értékhez tartozó  $r(u_s)$  görbe pontot.

1. Keressünk olyan  $i$  értéket, amelyre igaz  $u_i \leq u_s < u_{i+1}$ .
2. Legyen  $\lambda_j^1 = (u_s - u_j) / (u_{j+k-1} - u_j)$ ,  
for  $j = (i - k + 2)$  to  $i$  do

$$\mathbf{d}_j^{[1]} = (1 - \lambda_j^1)\mathbf{d}_{j-1} + \lambda_j^1\mathbf{d}_j$$

3. Ismételten alkalmazzuk a formulát:

for  $m = 2$  to  $(k - 1)$  do

for  $j = (i - k + m + 1)$  to  $i$  do

$$\begin{aligned}\lambda_j^m &= (u_s - u_j) / (u_{j+k-m} - u_j), \\ \mathbf{d}_j^{[m]} &= (1 - \lambda_j^m) \mathbf{d}_{j-1}^{[m-1]} + \lambda_j^m \mathbf{d}_j^{[m-1]}.\end{aligned}$$

Ha uniform módon adjuk meg a csomóértékeket és a lépésköz 1, akkor

$$\lambda_j^m = \frac{u_s - u_j}{k - m}.$$

4. Számítsuk ki a görbepontot  $r(u_s) = \mathbf{d}_i^{[k-1]}$ .

## 10.5. Interpoláció B-spline görbével

A B-spline görbe approximáló görbe, de használhatjuk interpolációra is. Farin [21], Piegl [63] munkái alapján is elemeztem ezt a problémát [52]-ban és [32]-ban. Keresünk egy olyan adott fokszámú B-spline görbét, azaz határozzuk meg kontrollpontjait és csomóértékeit, amely az adott paraméterértéknél az adott ponton áthalad. Ha  $Q_r$ ,  $r = 0, \dots, j$  pontok illeszkednek egy  $k$ -ad fokú nem racionális B-spline görbére, akkor kielégítik a következő  $(j + 1)$  egyenletből és  $(n + 1)$  ismeretlenből álló egyenletrendszert

$$Q_r = Q(\bar{u}_r) = \sum_{i=0}^n \mathbf{d}_i N_{i,k}(\bar{u}_r), \quad (10.4)$$

ahol  $2 \leq k \leq n + 1 \leq j$ , és az  $(\bar{u}_r)$  paramétereket hozzárendeljük minden  $Q_r$ -hez és választunk egy megfelelő  $U = u_0, \dots, u_m$  csomóvektort. A  $\mathbf{d}_i$  kontrollpontok lesznek az egyenletrendszer keresett  $n + 1$  db ismeretlenje. Az egyenletrendszernek egy együttható mátrixa van és természetesen  $s$  megoldáshalmaza, ha  $\mathbf{d}_i$ -kontrollpontoknak  $s$  koordinátája van, azaz  $s$  a  $Q_r$  pontok koordinátáinak a száma, tipikusan 2,3, vagy 4.

A (10.4) egyenletrendszert kifejtve a következőt kapjuk

$$\begin{aligned}Q(\bar{u}_0) &= N_{0,k}(\bar{u}_0) \mathbf{d}_0 + N_{1,k}(\bar{u}_0) \mathbf{d}_1 + \dots + N_{n,k}(\bar{u}_0) \mathbf{d}_n \\ Q(\bar{u}_1) &= N_{0,k}(\bar{u}_1) \mathbf{d}_0 + N_{1,k}(\bar{u}_1) \mathbf{d}_1 + \dots + N_{n,k}(\bar{u}_1) \mathbf{d}_n\end{aligned}$$

$$\begin{aligned} & \vdots \\ Q(\bar{u}_j) &= N_{0,k}(\bar{u}_j) \mathbf{d}_0 + N_{1,k}(\bar{u}_j) \mathbf{d}_1 + \dots + N_{n,k}(\bar{u}_j) \mathbf{d}_n \end{aligned}$$

amelyet átírhatunk (a szokásos módon) mátrix alakba

$$[Q] = [N] [\mathbf{d}].$$

Ha  $2 \leq k \leq n + 1 = j$ , azaz  $(n + 1)$  egyenletből és  $(n + 1)$  ismeretlenből áll az egyenletrendszer, akkor  $[N]$  mátrix kvadratikus, és az eredmény mátrixszorzással nyerhető, azaz,

$$[\mathbf{d}] = [N]^{-1} [Q]. \quad (10.5)$$

A probléma megoldása tehát  $\bar{u}_r$  és  $U$  meghatározására korlátozódik, mivel megválasztásuk befolyásolja a görbe alakját. Tegyük fel, hogy a paramétertartomány  $u \in [0,1]$ . A paraméterek kiválasztására sok módszer ismert, ekvidisztáns (egyenlőközű), ívhossz szerinti, centripetális mód. Számítási igényeket és a kapott görbe alakját is figyelembe véve a centripetális módszer a javasolt. Legyen

$$d = \sum_{r=1}^n \sqrt{|Q_r - Q_{r-1}|},$$

tudva, hogy  $\bar{u}_0 = 0$  és  $\bar{u}_n = 1$ ,

$$\bar{u}_r = \bar{u}_{r-1} + \frac{\sqrt{|Q_r - Q_{r-1}|}}{d} \quad r = 1, \dots, n - 1.$$

Ez a módszer jobb eredményt ad, amikor a pontok hirtelen élesen változnak, tehát éles "kanyart" várunk a görbétől. Ajánlott még a centripetális módszer kombinálása az átlagolás technikájával. Azaz

$$u_0 = \dots = u_k = 0 \quad u_{m-k} = \dots = u_m = 1$$

$$u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{u}_i \quad j = 1, \dots, n - p.$$

Ez a kombináció vezet az (10.4) egyenletrendszerhez, és biztosítja a következő előnyös tulajdonságot:  $N_{i,k}(\bar{u}_r) = 0$  ha  $|i - r| \geq k$ . Az egyenletrendszert megoldhatjuk (10.5) szerint is, de javasolt a Gauss elimináció (vagy a Gauss-Jordan módszer) használata. Érdekes az LU dekompozíció technikájával alsó és felső háromszögalakú komponensekre bontani az együtthatómátrixot (lásd [68]).

Bár az eredmény görbe mindenhol  $C^{k-2}$ -folytonos, előfordulhat, hogy nem lesz elég szép sima és tartalmazhat váratlan hullámokat és kilengéseket. A nem várt kilengések elkerülése végett keressünk kevesebb  $\mathbf{d}_i$  kontrollpontot mint  $Q_r$  input pontot, azaz  $2 \leq k \leq n + 1 < r$ . Mivel ebben az esetben  $[N]$  már nem kvadratikus, ezért a (10.4) egyenletrendszer átírva kompakt mátrix alakba a következőt eredményezi:

$$[Q] = [N] [\mathbf{d}]$$

$$[N]^T [Q] = [N]^T [N] [\mathbf{d}]$$

$$[\mathbf{d}] = \left[ [N]^T [N] \right]^{-1} [N]^T [Q]$$

Nem minden esetben kapunk megoldást, mivel a feladat túlhatározott, de ilyenkor numerikus úton kezelhető a probléma.

## 10.6. Racionális görbék

A racionális Bézier görbékhez hasonlóan lehet bevezetni a racionális B-spline görbét is. Térbeli parabolák vetületeként (pl. a  $z = 1$  síkra) állíthatunk elő kúpszeleteket, azaz, kúpszeletek térbeli másodrendű Bézier görbék centrális vetületei lesznek. Ezen megközelítés általánosítása a racionális Bézier görbe. (Hoschek[42] és Juhász [41] alapján.)

### 10.6.1. Racionális Bézier görbe

A Bernstein polinomokat a következő binomiális formulából származtathatjuk

$$1 = [(1-t) + t]^n = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i.$$

A

$$B_i^n(t) := \binom{n}{i} (1-t)^{n-i} t^i, \quad i = 0, 1, \dots, n$$

polinomokat  $n$ -edfokú Bernstein polinomoknak nevezzük.

**10.5. definíció.** Az

$$r(u) = \frac{\sum_{i=0}^n w_i \mathbf{d}_i B_i^n(u)}{\sum_{j=0}^n w_j B_j^n(u)}, \quad u \in [0,1] \quad (10.6)$$

kifejezéssel definiált görbét  $n$ -edrendű racionális Bézier görbének nevezzük. A  $\mathbf{d}_i$  ( $i = 0, 1, \dots, n$ ) pontokat a görbe kontrollpontjainak, az általuk definiált poligont kontrollpoligonnak, a  $w_i$  skalárokat pedig súlyoknak nevezzük.

A definíció szemléletesen ez azt jelenti, hogy a négydimenziós térben egy  $x, y, z, w$  koordinátarendszerben adott egy  $n$ -edrendű Bézier görbe  $\begin{bmatrix} w_i \mathbf{d}_i \\ w_i \end{bmatrix}$  kontrollpontjaival. Ezt a görbét levetítjük az origóból a  $w = 1$  hipersíkra (3 dimenziós).

Látható, hogy  $w_i = konst.$  esetén nem racionális Bézier görbét kapunk. Negatív súlyok esetén szingularitások léphetnek fel, azaz (10.6)-ban a nevező eltűnik, azaz  $\sum_{j=0}^n w_j B_j^n(u) = 0$ , amely akkor fordul elő, ha van olyan pont, amelynek vetülete végtelentávoli pont. A súlyok csak arányosság erejéig vannak meghatározva, azaz, a  $w_i$  -k tetszőleges  $0 < \lambda \in \mathbb{R}$  számmal szorozhatóak. A (10.6) definíció átalakítva kapjuk

$$r(u) = \sum_{i=0}^n \mathbf{d}_i \frac{w_i B_i^n(u)}{\sum_{j=0}^n w_j B_j^n(u)},$$

amelyből látható, hogy az alapfüggvény:

$$\frac{w_i B_i^n(u)}{\sum_{j=0}^n w_j B_j^n(u)}.$$

Ebből következik, hogy a racionális Bézier görbe a kontrollpontok olyan lineáris kombinációja, ahol az együtthatók nem negatívak és összegük 1. Érvényes tehát az affin invariancia tulajdonság is. A  $w_i$  súlyokat alakparamétereknek nevezzük, mivel ha növeljük egy  $w_i$  értékét akkor a görbe a megfelelő  $\mathbf{d}_i$  kontrollpont felé fog mozogni.

### 10.6.2. Racionális B-spline görbe, NURBS

A jelenlegi tervezési rendszerekben a nem uniform racionális B-spline görbék, amelynek szokásos rövidítése NURBS (nonuniform rational B-spline), biztosítják a legtöbb szabadságot, és a legtöbb alakváltoztatási lehetőséget a tervezők számára. Az előző pont alapján teljesen analóg módon megadhatjuk a racionális Bézier görbéhez hasonlóan a racionális B-spline görbét is. Szemléletesen tárgyalva azt mondhatjuk, hogy ha a négydimenziós térben egy  $x, y, z, w$  koordináta-rendszerben adott egy B-spline görbe  $\begin{bmatrix} w_i \mathbf{d}_i \\ w_i \end{bmatrix}$  kontrollpontjaival, és ezt a görbét centrálisan levetítjük az origóból a  $w = 1$  hipersíkra (3 dimenziós), akkor vetületként racionális B-spline görbét kapunk. Dimenziószámokat változtatva kapunk egyéb eseteket, pl eggyel csökkentve síkbeli racionális B-spline görbéhez jutunk.

**10.6. definíció.** Az

$$s(u) = \frac{\sum_{i=0}^n w_i \mathbf{d}_i N_i^k(u)}{\sum_{j=0}^n w_j N_j^k(u)}$$

kifejezéssel definiált görbét  $k - 1$ -ed fokú racionális B-spline görbének nevezzük. A  $\mathbf{d}_i$  ( $i = 0, 1, \dots, n$ ) pontokat a görbe kontrollpontjainak, az általuk definiált poligont kontrollpoligonnak, a  $w_i$  skalárokat pedig súlyoknak nevezzük. Az  $N_i^k(u)$  a  $k - 1$ -ed fokú normalizált B-spline alapfüggvény, amelyhez meg kell adnunk az  $\mathbf{U} = (u_0, u_1, \dots, u_n, u_{n+1}, \dots, u_{n+k})$  csomóvektort is, amelyben a csomóértékekre mint skalárookra igaz, hogy  $u_i \leq u_{i+1}$ ,  $u_i \in \mathbb{R}$  ( $i = 0, \dots, n + k$ ).

A  $w_i$  súlyokat a szingularitások elkerülése végett pozitívnak szokták választani, mivel ebben az esetben nem csak a nem racionális B-spline görbék kedvező tulajdonságait tartja meg (local support, konvex burok, folytonosan differenciálhatóság), hanem még újabb alakváltoztatási lehetőséghez is jutunk. A racionális Bézier görbéhez hasonlóan a  $w_i$  súlyokat alakparamétereknek nevezzük, mivel ha növeljük egy  $w_i$  értékét akkor a görbe a megfelelő  $\mathbf{d}_i$  kontrollpont felé fog mozogni. Ez a tulajdonság hasonló ahhoz, amelyet a csomóérték multiplicitással érünk el, de nem egyezik meg azzal. Megjegyezzük, hogy egyszerű átalakításokkal megkaphatjuk a racionális Cox- de Boor algoritmust is (lásd [41]).

Legyenek megadva a kontrollpontok (de Boor pontok) homogén koordináták alakban

$$\mathbf{D}_j = \begin{cases} (k_j \lambda_j, v_j \lambda_j, w_j \lambda_j, \lambda_j)^T = (\lambda_j \mathbf{d}_j, \lambda_j)^T, & \lambda \neq 0, \mathbf{d}_j \in E^3 \\ (u_j, v_j, w_j, 0)^T = (\bar{\mathbf{d}}_j, 0)^T, & \lambda = 0, \bar{\mathbf{d}}_j \in E^3 \end{cases},$$

ahol a  $\bar{\mathbf{d}}_j$  ideális pont, olyan pont képe, amely végtelenbe kerül a centrális vetítés során. Homogén koordináták alakban a  $k$ -ad rendű racionális B-spline görbe

$$S(u) = \sum_{j=0}^n \mathbf{D}_j N_j^k(u),$$

amely nem homogén alakban  $(x, y, z) \in \mathbb{R}^3$  esetén

$$x = \frac{\sum_{j=0}^n u_j \lambda_j N_j^k(u)}{\sum_{j=0}^n \lambda_j N_j^k(u)}, \quad y = \frac{\sum_{j=0}^n v_j \lambda_j N_j^k(u)}{\sum_{j=0}^n \lambda_j N_j^k(u)}, \quad z = \frac{\sum_{j=0}^n w_j \lambda_j N_j^k(u)}{\sum_{j=0}^n \lambda_j N_j^k(u)}.$$

## 10.7. B-spline felület

Ebben a pontban a B-spline felületet definiáljuk, mivel további kutatásaink szempontjából – a geometria tervezésnél a legnépszerűbb – NURBS felületekkel törődünk. Kindulásként tekintsük a következő felületek osztályát

$$\mathbf{s}(u, v) = M(u)\mathbf{g}(v), \quad u \in [u_0, u_1], \quad v \in [v_0, v_1],$$



ahol  $\mathbf{g}(v)$  egy tetszőleges görbe,  $M(u)$  pedig egy  $4 \times 4$ -es mátrix. Az így kapott felület nem más, mint az  $u$  paramétertől függő egyparaméteres görbesereg által sűrölt felület. A  $\mathbf{g}(v)$  görbe térbeli mozgása során egy felületet ír le. Megengedett, hogy a görbe alakja is változzon a mozgás során. Pontosán az utóbbi tulajdonsággal származtatható Bézier és B-spline felület is.

Tegyük fel, hogy az  $\mathbf{a}_i$  kontrollpontjával adott B-spline görbét mozgatjuk, úgy, hogy feltételezzük, hogy a B-spline görbe kontrollpontjai ugyancsak B-spline görbén mozognak. Jelöljük  $\mathbf{b}_{ij}$ -vel az  $\mathbf{a}_i$  kontrollpontok pályáját meghatározó B-spline görbe kontrollpontjait ( $j = 0, 1, \dots, m$ ), azaz  $\mathbf{b}_{i0} = \mathbf{a}_i$ , ( $i = 0, 1, \dots, n$ ). Ebben az esetben a mozgó B-spline görbe

$$\mathbf{a}(u) = \sum_{i=0}^n \mathbf{a}_i N_i^{k_1}(u),$$

amelynek kontrollpontjai az

$$\mathbf{a}_i(v) = \sum_{j=0}^m \mathbf{b}_{ij} N_j^{k_2}(v)$$

B-spline görbén mozognak. A két alakot kombinálva a következő felületet kapjuk

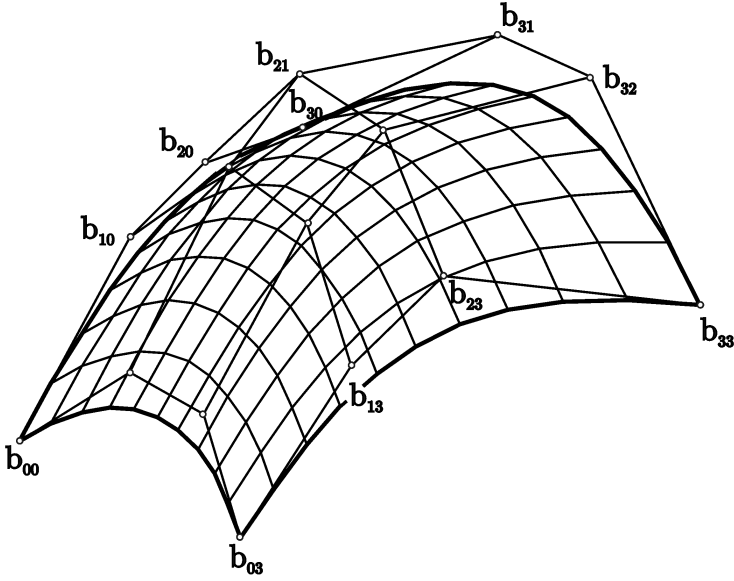
$$\mathbf{s}(u, v) = \sum_{i=0}^n \left( \sum_{j=0}^m N_j^{k_2}(v) \right) N_i^{k_1}(u) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{b}_{ij} N_i^{k_1}(u) N_j^{k_2}(v).$$

Ezt a felületet B-spline felületnek, vagy tenzor szorzattal előállított B-spline felületnek nevezzük. A  $\mathbf{b}_{ij}$  pontokat a felület kontrollpontjainak, az általuk meghatározott hálót pedig kontrollhálónak nevezzük. Megjegyezzük, hogy az előbbihez hasonlóan más görbék tenzor szorzataként is származtathatunk felületet. Az

$$\mathbf{s}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{b}_{ij} F_i(u) G_j(v)$$

felület tenzor szorzat felület (tensor product surface), ahol a  $F_i(u)$  és a  $G_j(v)$  különböző bázisfüggvények, például Bézier felület esetén Bernstein

polinomok, de lehetnek például Lagrange-, racionális Bézier- vagy racionális B-spline alapfüggvények is.

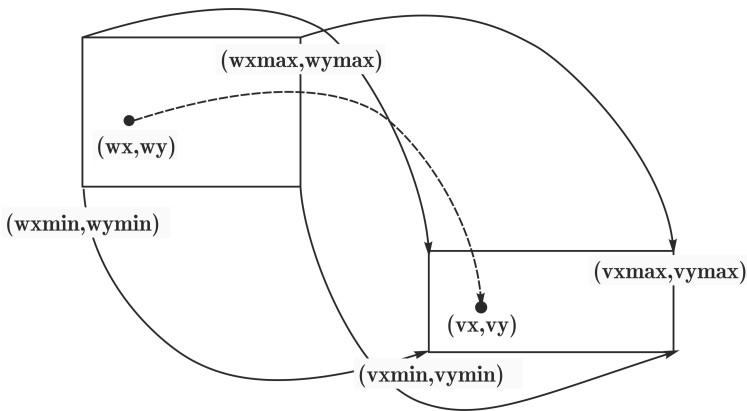


10.7. ábra. B-spline felület és kontrollháló

## 11. Függelék

### 11.1. Window-viewport transzformációk

A legtöbb grafikus problémát általában Descartes koordináta-rendszerben oldunk meg, tehát derékszögű úgynevezett világkoordináta-rendszerben adjuk meg az algoritmust. A világkoordináta-rendszer speciális, ezért gyakran transzformálni kell a periféria koordináta-rendszerbe. Általában a világ koordináta-rendszer derékszögű ablakát (Window) transzformáljuk az output rendszer képtartományába (Viewport). Gyors algoritmusra van szükségünk, mivel sok pont kezelésével nagy mennyiségű ismétlődő számítást kell elvégeznünk.



11.1. ábra. Window-viewport transzformáció

A geometriai input és output adatokat koordináta-rendszerek közötti leképezéseknek vetjük alá. A következő koordináta-rendszereket különböztetjük meg:

- világkoordináta-rendszer (World Coordinates, WC) a felhasználó ebben adja meg illetve ebben kapja vissza a geometria információkat.
- normalizált eszközkoordináta-rendszer (Normalized Device Coordinates, NDC), a különböző grafikus munkahelyek számára egységes koordináta-rendszer. ■

- eszközkoordináta-rendszere (Device Coordinates, DC), a munkahely hardverének megfelelő specifikus koordináta-rendszer.

A 11.1 ábra alapján könnyen előállíthatunk egy olyan függvényt, amely világkoordináta-rendszerben lévő pont leképezi a normalizált eszközkoordináta-rendszerbe. A függvénynek rendelkeznie kell a következő tulajdonságokkal: a világ koordináta-rendszer derékszögű ablakának (Window) határvonalpontjait az output rendszer képtartományának (Viewport) határvonalpontjaiba transzformálja.

Tehát a következő megfeleltetés igaz:

$$(wx \min, wy \min, wx \max, wy \max) \rightarrow (vx \min, vy \min, vx \max, vy \max)$$

Egyenestartó leképezés, vagyis egyenes képe is egyenes lesz. Az x és y tengelyek a két rendszerben páronként párhuzamosak egymással. Ebből következik, hogy a leképezés lineáris és a 19. ábra alapján a következő összefüggés vezethető le:

$$wx = wx \min + a(wx \max - wx \min), \text{ ahol } (0 \leq a \leq 1)$$

és igaz, hogy

$$vx = vx \min + a(vx \max - vx \min) \tag{11.1}$$

ebből adódik, hogy

$$a = \frac{wx - wx \min}{wx \max - wx \min}$$

visszahelyettesítve 11.1-be:

$$\begin{aligned} vx &= vx \min + \frac{wx - wx \min}{wx \max - wx \min} (vx \max - vx \min) = \\ &= vx \frac{vx \max - vx \min}{wx \max - wx \min} + vx \min - vx \min \frac{vx \max - vx \min}{wx \max - wx \min} \end{aligned}$$

Új jelöléseket bevezetve:  $dvx := vx \max - vx \min$ ,  $dwx := wx \max - wx \min$  és hasonlóan számítható  $dvy$  és  $dwy$ .

$$vx = \frac{dvx}{dwx} \cdot wx + \left( vx \min - wx \min \frac{dvx}{dwx} \right) \quad (11.2)$$

ami lineáris leképezés.

A másik koordinátára hasonló kifejezést kapunk:

$$vy = \frac{dvy}{dwy} \cdot wy + \left( vy \min - wy \min \frac{dvy}{dwy} \right)$$

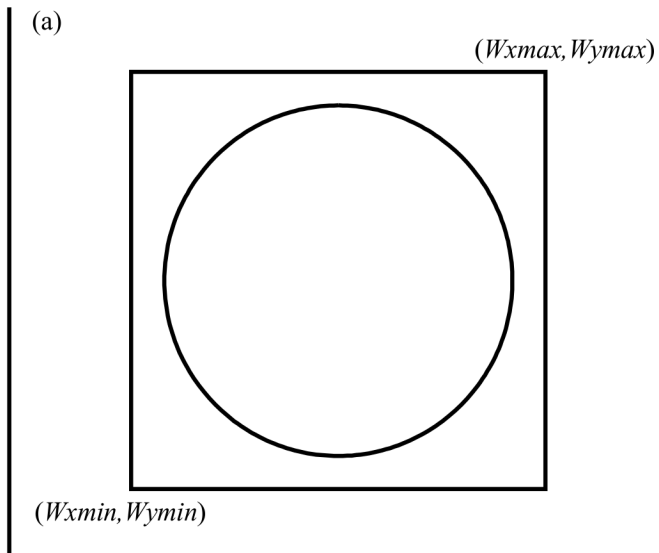
Tehát végeredményként azt kaptuk, hogy

$$\begin{aligned} vx &= A \cdot wx + B \\ vy &= C \cdot wy + D, \end{aligned}$$

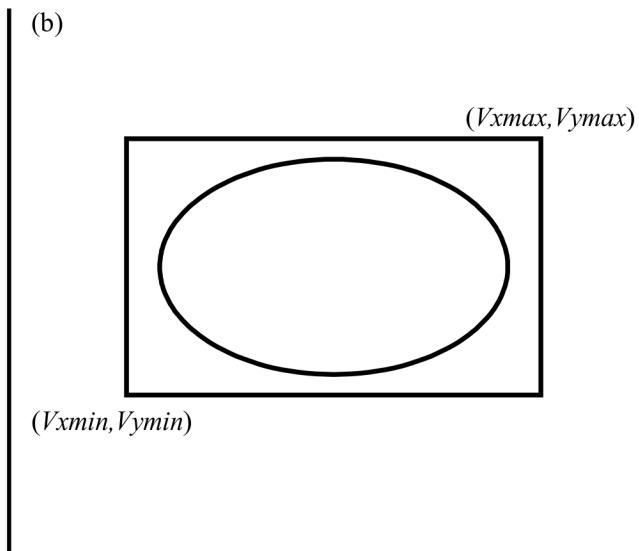
Mivel  $A, B, C, D$  a leképezés során függetlenek a  $(wx, wy)$  transzformálandó ponttól ezért ezeket az értékeket csak egyszer kell kiszámítani. Ha  $a < 0$  vagy  $a > 1$  akkor a transzformálandó pont kívül esik az ablakon ezért nem kell vele foglalkozni.

### 11.1.1. Uniform eszköz transzformáció

A 11.2 és 11.3 ábrák egy Window-Viewport transzformációt illusztrálnak, ahol a torzítási tényezők (aspect ratio) nem egyeznek meg. Torzítási tényezőn a magasság és a szélesség arányát értjük. Ilyen esetekben a kör képe ellipszis lesz. A leképezési függvényben szereplő  $(dvx/dwx)$  és  $(dvy/dwy)$  nem egyenlő. Ez elfogadható, de a programozónak nagyon ügyelnie kell, hogy ne feledkezzen meg a torzulásról. Az *uniform transzformációknál* követelmény az előbbi hányadosok egyenlősége. is nevezhetjük Az uniform transzformáció valójában hasonlósági transzformáció is, mivel hasonló alakzatok keletkeznek.



11.2. ábra. Világ koordináták



11.3. ábra. Normalizált eszköz koordináták

### 11.1.2. Window-viewport transzformáció 2. változat

Megoldhatjuk a problémát síkbeli – 3x3-as – transzformációs mátrixok szorzásával is.

1. Az ablak bal alsó sarkát az origóba toljuk.
2. Skálázunk az ablak és a képmező (viewport) oldalárányaival, és az y tengelyre tükrözünk.
3. Az ablak bal felső sarkát a képmező bal felső sarkába toljuk. (Az y tengely mentén többet kell tolnunk a tükrözés miatt.)

Az első lépésben az eltolás vektora

$$\mathbf{d} = (-window.Left, -window.Top)$$

$$\mathbf{M1} = \begin{bmatrix} 1 & 0 & -window.Left \\ 0 & 1 & -window.Top \\ 0 & 0 & 1 \end{bmatrix},$$

A második lépésben az skálázás mátrixa, amely tartalmazza az y tengelyre való tükrözést is:

$$\mathbf{M2} = \begin{bmatrix} \frac{viewport.Width}{window.Width} & 0 & 0 \\ 0 & -\frac{viewport.Height}{window.Height} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

A harmadik lépésben visszakell tolnunk a már skálázott ablakot a képmezőbe, akkor eltolás vektora

$$\mathbf{d} = (-viewport.Left, -viewport.Top + viewport.Height)$$

$$\mathbf{M3} = \begin{bmatrix} 1 & 0 & -viewport.Left \\ 0 & 1 & -viewport.Top + viewport.Height \\ 0 & 0 & 1 \end{bmatrix}$$

Figyelnünk kell arra, hogy a transzformációkat fordított sorrendben kell elvégezni, erre a program mellékletben ügyeltünk. Ha összeszorozzuk a mát-

rixokat, akkor a [11.2](#)-ban közölt eredményeket kapjuk.



## 11.2. Programkódok

### Az identitás mátrixa:

```
// NxN-es egységmátrixot ad vissza
public static Matrix Identity(int dimensions)
{
    Matrix ret = new Matrix(dimensions, dimensions);
    for (int i = 1; i <= dimensions; ++i)
    {
        ret[i, i] = 1;
    }
    return ret;
}
```

### Az eltolás mátrixa:

```
// Háromdimenziós eltolási mátrix
public static Matrix Translate3D(double dx, double dy, double dz)
{
    return new Matrix(4, 4,
        1, 0, 0, dx,
        0, 1, 0, dy,
        0, 0, 1, dz,
        0, 0, 0, 1);
}
```

### Az elforgatás mátrixa:

```
// 3D-s forgatási mátrix az X tengely körül
public static Matrix Rotate3Dx(double alpha)
{
    return new Matrix(4, 4,
        1, 0, 0, 0,
        0, Math.Cos(alpha), -Math.Sin(alpha), 0,
        0, Math.Sin(alpha), Math.Cos(alpha), 0,
        0, 0, 0, 1);
}
```

```
// 3D-s forgatási mátrix az Y tengely körül
public static Matrix Rotate3Dy(double alpha)
{
    return new Matrix(4, 4,
        Math.Cos(alpha), 0, Math.Sin(alpha), 0,
        0, 1, 0, 0,
        -Math.Sin(alpha), 0, Math.Cos(alpha), 0,
        0, 0, 0, 1);
}
```

```

}

// 3D-s forgatási mátrix a Z tengely körül
public static Matrix Rotate3Dz(double alpha)
{
    return new Matrix(4, 4,
        Math.Cos(alpha), -Math.Sin(alpha), 0, 0,
        Math.Sin(alpha),  Math.Cos(alpha), 0, 0,
        0, 0, 1, 0,
        0, 0, 0, 1);
}

```

### Rodrigues-féle forgatás mátrixa:

```

// Tetszőleges tengely körüli forgatás
public static Matrix Rodrigues(double phi, Point3D P1, Point3D P2)
{
    double h = Math.Sqrt((P2.X - P1.X) * (P2.X - P1.X) +
        (P2.Y - P1.Y) * (P2.Y - P1.Y) +
        (P2.Z - P1.Z) * (P2.Z - P1.Z));

    // A normált vektor koordinátái:
    double cx, cy, cz, d, CosPhi, SinPhi;
    cx = (x2 - x1) / h;
    cy = (y2 - y1) / h;
    cz = (z2 - z1) / h;
    d = Math.Sqrt(cz * cz + cy * cy);
    CosPhi = Math.Cos(phi);
    SinPhi = Math.Sin(phi);
    Matrix cg = new Matrix(4, 4,

        (-CosPhi + 1) * cx * cx + CosPhi,
        (-CosPhi + 1) * cx * cy - SinPhi * cz,
        (-CosPhi + 1) * cx * cz + SinPhi * cy, 0,

        (-CosPhi + 1) * cx * cy + SinPhi * cz,
        (-CosPhi + 1) * cy * cy + CosPhi,
        (-CosPhi + 1) * cy * cz - cx * SinPhi, 0,

        (-CosPhi + 1) * cx * cz - SinPhi * cy,
        (-CosPhi + 1) * cy * cz + cx * SinPhi,
        (-CosPhi + 1) * cz * cz + CosPhi, 0,

        0, 0, 0, 1);
    return cg;
}

```

## A koordináta síkra való tükrözés mátrixa:

```
// x-y síkra való tükrözés
public static Matrix ReflectXY()
{
    return new Matrix(4, 4,
        1, 0, 0, 0,
        0, 1, 0, 0,
        0, 0, -1, 0,
        0, 0, 0, 1);
}
```

## Tetszőleges síkra való tükrözés mátrixa, 1.változat:

```
\\Tetszőleges síkra való tükrözés 1.változat
public static Matrix Reflection1( Point3D A, Point3D B, Point3D C)
{
    Point3D v1 = new Point3D(B.X - A.X, B.Y - A.Y, B.Z - A.Z),
    v2 = new Point3D(C.X - A.X, C.Y - A.Y, C.Z - A.Z),
    normv = new Point3D(0, 0, 0);
    // vektoriális szorzat normv=v1xv2
    normv.X = v2.Y * v1.Z - v2.Z * v1.Y;
    normv.Y = v2.Z * v1.X - v2.X * v1.Z;
    normv.Z = v2.X * v1.Y - v2.Y * v1.X;
    //normalizálás :/normv|=1
    double h = Math.Sqrt(normv.X * normv.X + normv.Y * normv.Y +
        normv.Z * normv.Z);
    //normált vektor koordinátáit segédváltozóban tároljuk
    double cx, ay, az;
    cx = normv.X / h;
    cy = normv.Y / h;
    cz = normv.Z / h;
    Matrix cg = new Matrix(4, 4,
        1 - 2 * cx * cx, -2 * cy * cx, -2 * cz * cx, 0,
        -2 * cy * cx, 1 - 2 * cy * cy, -2 * cy * cz, 0,
        -2 * cz * cx, -2 * cy * cz, -2 * cz * cz + 1, 0,
        0, 0, 0, 1);
    return cg;
}
```

## Tetszőleges síkra való tükrözés mátrixa, 2.változat:

```
\\Tetszőleges síkra való tükrözés 2.változat
public static Matrix Reflection2( Point3D A, Point3D B, Point3D C)
{
    Point3D v1 = new Point3D(B.X - A.X, B.Y - A.Y, B.Z - A.Z),
```

```

        v2 = new Point3D(C.X - A.X, C.Y - A.Y, C.Z - A.Z),
        normv = new Point3D(0, 0, 0);
// vektoriális szorzat normv=v1xv2
normv.X = v2.Y * v1.Z - v2.Z * v1.Y;
normv.Y = v2.Z * v1.X - v2.X * v1.Z;
normv.Z = v2.X * v1.Y - v2.Y * v1.X;
//normalizálás :|normv|=1
double h = Math.Sqrt(normv.X * normv.X + normv.Y * normv.Y +
        normv.Z * normv.Z);
//normált vektor koordinátáit segédváltozóiban tároljuk
double cx, cy, cz;
cx = normv.X / h;
cy = normv.Y / h;
cz = normv.Z / h;
//———— különbség cz 1.változattól —————
// a sík fix pontja legyen cz A pont
double x0 = A.X, y0 = A.Y, z0 = A.Z,
        d = -cx * x0 - cy * y0 - cz * z0;
Matrix cg = new Matrix(4, 4,
        1 - 2 * cx * cx, -2 * cy * cx, -2 * cz * cx, -2 * cx * d,
        -2 * cy * cx, 1 - 2 * cy * cy, -2 * cy * cz, -2 * cy * d,
        -2 * cz * cx, -2 * cy * cz, -2 * cz * cz + 1, -2 * cz * d,
        0, 0, 0, 1);
return cg;
//———— különbség vége —————
}

```

### A skálázás mátrixa:

```

// Skálázás
public static Matrix Scale3D(double kx, double ky, double kz)
{
    return new Matrix(4, 4,
        kx, 0, 0, 0,
        0, ky, 0, 0,
        0, 0, kz, 0,
        0, 0, 0, 1);
}

```

### A hasonlóság mátrixa 1.változat:

```

// Kicsinyítés vagy nagyítás
public static Matrix Scale3D(double k)
{
    return new Matrix(4, 4,
        k, 0, 0, 0,
        0, k, 0, 0,

```

```

        0, 0, k, 0,
        0, 0, 0, 1);
}

```

## A hasonlóság mátrixa 2.változat:

```

// Kicsinyítés vagy nagyítás
public static Matrix Scale3D(double k)
{
    return new Matrix(4, 4,
        1, 0, 0, 0 ,
        0, 1, 0, 0 ,
        0, 0, 1, 0 ,
        0, 0, 0, 1/k);
}

```

## A vetítés mátrixa:

```

//Merőleges vetítés mátrixa
public static Matrix OrtogonalProj()
{
    return new Matrix(4, 4,
        1, 0, 0, 0,
        0, 1, 0, 0,
        0, 0, 0, 0,
        0, 0, 0, 1);
}
//Párhuzamos vetítés mátrixa
public static Matrix ParalellProj(double vx,double vy,double vz)
{
    return new Matrix(4, 4,
        1, 0, -vx/vz, 0,
        0, 1, -vy/vz, 0,
        0, 0, 0, 0,
        0, 0, 0, 1);
}
//Centrális vetítés 1.változat
// A nézőpont a Z tengely pozitív felén van
// és negatív irányba néz, a képsík az XY sík.
public static Matrix CentralProj1(double eyeDistance)
{
    return new Matrix(4, 4,
        1, 0, 0, 0,
        0, 1, 0, 0,
        0, 0, 0, 0,
        0, 0, -1 / eyeDistance, 1);
}

```

```

//Centrális vetítés 2.változat
// A nézőpont az origóban van
// és pozitív irányba néz, a képsík az z=eyeDistance sík.
public static Matrix CentralProj2(double eyeDistance)
{
    return new Matrix(4, 4,
        1, 0, 0, 0,
        0, 1, 0, 0,
        0, 0, 1, 0,
        0, 0, 1 / eyeDistance, 0);
}

```

### Speciális nézeti rendszer:

```

// Viewing system
// Áttérés világ koordináákról kamera koordinátákra
// Kamera: gömbi koordinátákkal van megadva;
// a világkoordináta rendszer origójába néz;
// a függőleges irány a világ koordináta-rendszer
// z tengelye jelöli ki;
// balsodrású a kamera koordináta rendszer.
public static Matrix SpecViewingSystem(double alpha, double beta,
                                        double r)
{
    double sina = Math.Sin(alpha * Math.PI / 180);
    double cosa = Math.Cos(alpha * Math.PI / 180);
    double sinb = Math.Sin(beta * Math.PI / 180);
    double cosb = Math.Cos(beta * Math.PI / 180);

    return new Matrix(4, 4,
        -sina, cosa, 0, 0,
        -cosa * cosb, -sina * cosb, sinb, 0,
        -cosa * sinb, -sina * sinb, -cosb, r,
        0, 0, 0, 1);
}

```

### Általános nézeti rendszer:

```

// General Viewing system u,v,n vektorokkal
public static Matrix GenViewingSystem(Point3D C, Point3D F,
                                        Point3D Up, string sodras)
{
    // Áttérés világ koordináákról kamera koordinátákra
    // Kamera: Descartes koordinátákkal, van megadva;
    // a z tengely iránya : n, F-C;
    // a y tengely iránya: v=Up-skalarisszorzat(Up,n)*n;
    // a x tengely iránya: u=vxn->jobsodrású

```

```

// -u=nxv->balsodrású koordináta-rendszer.
Point3D n = new Point3D(F.X - C.X, F.Y - C.Y, F.Z - C.Z),
        v = new Point3D(0, 0, 0),
        u = new Point3D(0, 0, 0);

///n=1
n = Normalizalas(n);
v.X = Up.X - SkalariSzorzat(Up, n) * n.X;
v.Y = Up.Y - SkalariSzorzat(Up, n) * n.Y;
v.Z = Up.Z - SkalariSzorzat(Up, n) * n.Z;
v = Normalizalas(v);
u = VektorialisSzorzat(v, n);
u = Normalizalas(u);
if (sodras == "bal")
{
    u.X = -u.X; u.Y = -u.Y; u.Z = -u.Z;
}
Matrix Tc = new Matrix(4, 4,
                    u.X, u.Y, u.Z, 0,
                    v.X, v.Y, v.Z, 0,
                    n.X, n.Y, n.Z, 0,
                    0, 0, 0, 1);
Matrix T = new Matrix(4, 4);
T = Matrix.Translate3D(-C.X, -C.Y, -C.Z);
Matrix cg = new Matrix(4, 4);
cg = Tc * T;
return cg;
}

```

## Window-viewport transzformáció mátrixa:

```

// Window to viewport transzformációs mátrix
public static Matrix WindowToViewport(RectangleF window,
                                     RectangleF viewport)
{
    // 1. Az ablak bal alsó sarkát az origóba toljuk.
    // 2. Skálázunk az ablak és a viewport oldalarányaival,
    //    és az Y tengelyre tükrözünk.
    // 3. A kép bal felső sarkát a viewport bal felső sarkába toljuk.
    //    (Az Y tengely mentén többet kell tolnunk a tükrözés miatt.)

    return Matrix.Translate2D(viewport.Left,
                             viewport.Top + viewport.Height) *
           Matrix.Scale2D(viewport.Width / window.Width,
                          -viewport.Height / window.Height) *
           Matrix.Translate2D(-window.Left, -window.Top);
}

```

## Kavalier-axonometria mátrixa:

```
// A Kavalier-axonometria mátrixa.  
// Az X és Y tengelyek egységvektorainak képe önmaga,  
// a Z tengely egységvektorának képe length hosszú, és  
// angle szöveget zár be az X tengellyel  
public static Matrix KavalierAxonometry(double length, double angle)  
{  
    return new Matrix(3, 4,  
                      1, 0, -length * Math.Cos(angle), 0,  
                      0, 1, -length * Math.Sin(angle), 0,  
                      0, 0, 0, 1);  
}
```

## A mátrix osztály:

```
class Matrix  
{  
    // a sorok és oszlopok száma  
    private int rows, cols;  
    public int Rows  
    {  
        get  
        {  
            return rows;  
        }  
    }  
    public int Cols  
    {  
        get  
        {  
            return cols;  
        }  
    }  
    // Az elemeket 1-től indexelünk, hogy közelebb legyünk  
    // a matematikai jelöléshez.  
    // Az elemek kívülről nem módosíthatók  
    private double[,] elements;  
    public double this[int i, int j]  
    {  
        get  
        {  
            return elements[i - 1, j - 1];  
        }  
        private set
```



```

        {
            elements[i - 1, j - 1] = value;
        }
    }

    // Az osztály metódusainak a felsorolása , lásd lentebb.
    //...
}
//Metódusok
//Mátrix létrehozása, az elemek implicite 0-k lesznek
//Privát a konstruktor, mivel kívülről nincs értelme hívni
//(a hívó nem tudná utólag kitölteni az elemeket)
private Matrix(int _rows, int _cols)
{
    if (_rows <= 0 || _cols <= 0)
    {
        throw new ArgumentException("Legyen_pozitív_a_dimenzió!");
    }
    rows = _rows; cols = _cols;
    elements = new double[_rows, _cols];
}

//Mátrix létrehozása a dimenziók és az elemek megadásával
public Matrix(int _rows, int _cols, params double[] values)
    : this(_rows, _cols)
{
    // ellenőrizzük, hogy a megadott elemek száma megfelelő-e
    if (values.Length != rows * cols)
    {
        throw new ArgumentException("Nem_megfelelő_az_elemek_száma!");
    }
    // a values tömbből sorfolytonosan töltjük fel a mátrix elemeit
    int curr = 0;
    for (int i = 0; i < rows; ++i)
        for (int j = 0; j < cols; ++j)
        {
            elements[i, j] = values[curr];
            ++curr;
        }
}

// A mátrixszorzás operátora
public static Matrix operator *(Matrix left, Matrix right)
{
    // Megnézzük, lehet-e egyáltalán szorozni a két mátrixot
    if (left.cols != right.rows)

```

```

    {
        throw new ArgumentException("Nem_összeszorozható_matrixok!");
    }
    Matrix res = new Matrix(left.rows, right.cols);
    // Kihasználva azt, hogy a privát konstruktorunk 0-val
    // töltötte fel a res mátrixot, elég a ciklus
    // belsejében += operátort használni.
    for (int i = 1; i <= res.rows; ++i)
        for (int j = 1; j <= res.cols; ++j)
            for (int k = 1; k <= left.cols; ++k)
                {
                    res[i, j] += left[i, k] * right[k, j];
                }
    return res;
}

// Kétdimenziós eltolási mátrix
public static Matrix Translate2D(double x, double y)
{
    return new Matrix(3, 3,
        1, 0, x,
        0, 1, y,
        0, 0, 1);
}

// Kétdimenziós skálázási mátrix
public static Matrix Scale2D(double x, double y)
{
    return new Matrix(3, 3,
        x, 0, 0,
        0, y, 0,
        0, 0, 1);
}

// NxN-es egységmátrixot ad vissza
public static Matrix Identity(int dimensions)
{
    Matrix ret = new Matrix(dimensions, dimensions);
    for (int i = 1; i <= dimensions; ++i)
        {
            ret[i, i] = 1;
        }
    return ret;
}

```

```
\\A mátrix transzponáltja
public Matrix Transpose()
{
    Matrix ret = new Matrix(cols , rows);
    for (int i = 1; i <= rows; ++i)
        for (int j = 1; j <= cols; ++j)
        {
            ret[j, i] = this[i, j];
        }
    return ret;
}
```

## Irodalomjegyzék

- [1] ALDER, M., TOGNERI, R., LAI, E., ATTIKIOUZEL, Y., Kohonen's algorithm for the numerical parametrisation of manifolds, *Pattern Recognition Letters* 11, pp. 313-319, 1990.
- [2] AUMANN, G., Approximate development of skew ruled surfaces. *Comput. & Graph.* 13 361-366, 1989.
- [3] AUMANN, G., Interpolation with developable Bézier patches. *Computer Aided Geometric. Design.* 8 409-420, 1991.
- [4] BARHAK, J., FISCHER, A., Parametrization and reconstruction from 3D scattered points based on neural network and PDE techniques, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 7, No.1, 2001.
- [5] BARSKY, B., *Computer Graphics and Geometric Modelling Using Beta-splines*, Springer-Verlag, Berlin, 1988.
- [6] BLUM, H., A transformation for extracting new descriptions of shape, *IEEE Proceedings of the Symposium on Models for the Speech and Vision Form*, Boston, pp. 362-380, 1964.
- [7] BODDULURI, R., RAVANI, B., Design of developable surfaces using duality between plane and point geometries, *Computer-Aided Design*, 10 ,pp. 621-632, 1993.
- [8] BODDULURI, R., RAVANI, B., Geometric Design and Fabrication of Developable Surfaces, *ASME Adv. Design Autom.* 2, pp. 243-250, 1992.
- [9] BOEHM, W., FARIN, G., AND KAHMANN, J., A survey of curve and surface methods in CAGD, *Computer Aided Geometric Design*, 1, pp.1-60, 1984.
- [10] de BOOR, C., On calculating with B-splines, *J. Approx. Theory*, 6:50-62, 1972.

- [11] BELONGIE, SERGE, Rodrigues' Rotation Formula, From MathWorld—A Wolfram Web Resource, created by Eric W. Weisstein. <http://mathworld.wolfram.com/RodriguesRotationFormula.html>
- [12] BORGULYA, I., Neurális hálók és fuzzy-rendszerek, Dialóg Campus Kiadó, Budapest-Pécs, 1998.
- [13] BRANHILL, R. E., Representation and Approximation of Surfaces, in: Rice, J.R. (ed): Mathematical Software III., Academic Press, New York, 1977.
- [14] BUDÓ, Á., Kísérleti fizika, I kötet, Tankönyvkiadó, Budapest, pp 116-118, 1981.
- [15] CASTILLO, E., Functional Networks, Neural Processing Letters 7, pp. 151-159, 1998.
- [16] COONS, S.A., Surface Patches and B-spline Curves, Computer Aided Geometric Design, Academic Press, 1974.
- [17] COX, M., The numerical evaluation of B-splines, DNAC 4, National Physical Laboratory, 1971.
- [18] COXETER, H. S. M., Projektív geometria, Gondolat Könyvkiadó, Budapest, 1986., Eredeti: Projective Geometry, Second Edition, University of Toronto Press, Toronto, 1974.
- [19] DATTA, A., PARUI, S.K., CHAUDHURI, B.B., Skeletonization by topology-adaptive self-organizing neural network, Pattern Recognition 34, Elsevier Science, pp. 617-629, 2001.
- [20] ECK, M., HADENFELD, J., Local energy fairing of B-spline curves, in: Farin, G., Hagen, H., Noltemeier, H. (Eds.), Computing 10. Springer, Berlin, 1995.
- [21] FARIN, G., Curves and Surfaces for Computer Aided Geometric Design A Practical Guide, Academic Press, 1996.

- [22] FLOATER, M., Parametrization and smooth approximation of surface triangulations, *Computer Aided Geometric Design*, 14, pp. 231-250, 1997.
- [23] FOLEY, T., HAGEN, H., *Advances in Scattered Data Interpolation*, *Surv. Math. Ind. Vol.4.*, pp.71-84., 1994.
- [24] FREEMAN, J., SKAPURA., D., *Neural Networks; Algorithms, Applications and Programming Techniques*, Addison-Wesley, 1991.
- [25] FREY, W., H., BINDSCHADLER, D., *Computer Aided Design of a Class of Developable Bézier Surfaces*, General Motors R&D Publication 8057, 1993.
- [26] GORDON, W., RIESENFELD, R., B-spline curves and surfaces, In R. E. Barnhill and R. F. Riesenfeld editors, *Computer Aided Geometric Design*, p. 95-126, Academic Press, 1974.
- [27] GREINER, G., HORMANN, K., Interpolating and approximating scattered 3D data with hierarchical tensor product B-splines, in: Méhauté, A. L., Rabut, C., Schumaker, L. (Eds.), *Surface Fitting and Multiresolution Methods*. Vanderbilt University Press, Nashville, pp. 163-172., 1997.
- [28] GU, P., YAN, X., *Neural Network Approach to the Reconstruction of Freeform Surfaces for Reverse Engineering*, *Computer Aided Design*, Vol. 27, No.1. pp.59-64, 1995.
- [29] HERMAN, I., *The use of projective geometry in computer graphics*, *LectureNotes in Computer Science*, 564, Springer-Verlag, 1991.
- [30] HERMANN, T., KOVÁCS, Z., VÁRADY, T., *Special applications in surface fitting*, in: Starsser, W., Klein, R., Rau, R. (Eds.), *Geometric Modeling: Theory and Practice*. Springer, pp. 14-31, 1997.
- [31] HLAVATY, V., *Differential line geometry*, Nordhoff Ltd, Groningen, 1953.

- [32] HOFFMANN M., KOVÁCS E., Interpolation possibilities using rational B-spline curve, *Acta Acad. Paed. Agriensis*, Vol. XXV., pp.103-107., 1998.
- [33] HOFFMANN M., KOVÁCS E., Constructing ruled B-spline surfaces by Kohonen neutral network, *Acta Acad. Paed. Agriensis*, TOM. XXVII., Sectio Mathematicae, pp. 63-68, 2000.
- [34] HOFFMANN M., KOVÁCS E., Developable surface modelling by neutral network, *Computers & Mathematics with Applications (közlésre elfogadva 2002)*
- [35] HOFFMANN M., Interpolation by Beta-spline, *Proceedings of the International Conference of Applied Informatics*, pp. 36-44, Eger, 1993.
- [36] HOFFMANN M., Modified Kohonen Neural Network for Surface Reconstruction, *Publ. Math. Debrecen*, 54 Suppl. 857-864, 1999.
- [37] HOFFMANN, M., VÁRADY, L., Free-form curve design by neural networks, *Acta. Acad. Paed. Agriensis*, , Tom.XXIV., pp. 99–104, 1997.
- [38] HOFFMANN, M., VÁRADY, L., Free-form Surfaces for Scattered Data by Neural Networks, *Journal for Geometry and Graphics*, Vol.2, No.1, pp. 1–6, 1998.
- [39] HORMANN, K., GREINER, G., An efficient global parametrization method, in: Laurent, P.-J., Sablonnière, P., Schumaker, L. (Eds.), *Curve and Surface Design: Saint Malo 1999*. Vanderbilt University Press, Nashville, pp. 153-162., 2000.
- [40] HORN-YANG CHEN, POTTMANN, H., Approximation by Ruled Surfaces, *Technical Report*, Nr.46, 1997.
- [41] HORVÁTH, I., JUHÁSZ, I., Számítógéppel segített gépészeti tervezés, *Műszaki könyvkiadó*, 1996.
- [42] HOSCHEK, J., LASSER, D., *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, Wellesley, MA, 1993.

- [43] HOSCHEK, J., POTTMANN, H., Interpolation and approximation with developable B-spline surfaces. In: *Mathematical Methods for Curves and Surfaces*, (Edited by M. Dæhlen *et al.*), pp.255-264, Vanderbilt University Press, Nashville, 1995.
- [44] HOSCHEK, J., SCHNEIDER, M., Interpolation and approximation with developable surfaces. In: *Curves and Surfaces with Applications in CAGD*, (Edited by A. Le Méhauté *et al.*), pp.185-202, Vanderbilt University Press, Nashville, 1997.
- [45] HOSCHEK, J., SCHWANECKE, U., Interpolation and approximation with ruled surfaces, in: *The Mathematics of Surfaces VII.* (ed.:Robert Cripps), Elsevier, pp. 213–231, 1988.
- [46] HOSCHEK, J., *Dual Bézier Curves and Surfaces*, *Surfaces in Computer Aided Geometric Design*, R. E. Barnhill & W Boehm, eds., North Holland, pp.147-156, 1983.
- [47] IGLESIAS, A., GÁLVEZ, A., Applying functional Networks to fit data points from B-spline surfaces, in: *Proceedings of the Computer Graphics Internatinal, CGI'*, Hong-kong pp., 329-332, 2001.
- [48] JUHÁSZ, I., *Számítógépi grafika és geometria*, Miskolci Egyetemi Kiadó, 1993.
- [49] KLEIN, F., *Vorlesungen Über Höhere Geometrie*, Springer, Berlin, 1926.
- [50] KOHONEN, T., *Self-organization and associative memory*, Springer Verlag, 1984.
- [51] KOVÁCS E., Studying and improving linear mappings by artificial neural networks, *Acta Acad. Paed. Agriensis TOM. XXVI., Sectio Mathematicae*, pp.104-107, 1999.
- [52] KOVÁCS E., Curve reconstruction from scattered data by Kohonen network, *Acta Acad. Paed. Agriensis, Vol. XXVII., Sectio Mathematicae*, pp.69-74., 2000.



- [53] LANG, J., RÖSCHEL, O., Developable  $(1, n)$ -Bézier surfaces. *Computer Aided Geom. Design.* 9 291-298, 1992.
- [54] LIPPMANN, R.P., An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, pp.18-21, April 1987.
- [55] LOOP, C., DEROSE, T., Generalized B-spline Surface of Arbitrary Topology, *Computer Graphics*, Vol.24., N.4., pp.347-356, 1990.
- [56] MA, W., KRUTH, J., Parametrization of randomly measured points for least squares fitting of B-spline curves and surfaces , *Computer Aided Design*, 27 (9), pp. 663-675, 1995.
- [57] Márkus, A., Renner, G., Vánca, J., Genetic algorithms in free form curve design, Daehlen, M., Lyche, T., Schumaker, L. (Eds.), *Mathematical Methods for Curves and Surfaces*, Vanderbilt University Press, pp. 343-354., 1995
- [58] MCCULLOGH, W. ÉS PITTS, W., How We Know Universals: the Perception of Auditory and Visual Forms, *Bulletin of Math. Biophysics*, Vol.9, pp.127-147, 1947.
- [59] MCCULLOGH, W. ÉS PITTS, W., A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Math. Biophysics*, Vol.5, pp.115-133, 1943.
- [60] NEUMANN, J., Probabilistic Logic and the Synthesis of Reliable Organisms from Unreliable Components, in: *Automata Studies*, Princeton University Press, Princeton, pp.43-98, 1956.
- [61] NIELSON, G.M., FRANKE, R., Surface Construction Based Upon Triangulations, in: *Surfaces in CAGD*, North-Holland, Amsterdam, 1983.
- [62] PFEIFLE, R., SEIDEL, H.P., Spherical Triangular B-splines with Application to Data Fitting, *EUROGRAPHICS '95*, Vol.14., N.3., pp.89-96, 1995.
- [63] PIEGL, L., TILLER, W., *The NURBS Book*, Springer Verlag, 1997.

- [64] POTTMANN, H., FARIN, G., Developable rational Bezier and B-spline surfaces, *Computer Aided Geometric Design*, 12, pp. 513–531, 1995.
- [65] POTTMANN, H., PETERNELL, M., RAVANI, B., An introduction to line geometry with applications, *Computer Aided Geom. Design*. 31, pp.3-16, 1999
- [66] POTTMANN, H., WALLNER, J., *Computational Line Geometry*, Springer -Verlag, Berlin Heidelberg, 2001.
- [67] POTTMANN, H., WALLNER, J., *Approximation Algorithms for Developable Surfaces*, Technical Report, Nr.51, 1998.
- [68] PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A., VETTERLING, W. T., *Numerical Recipes in Pascal, The Art of Scientific Computing*, Cambridge University Press, 1989.
- [69] RÉNYI, A., *Wahrscheinlichkeitsrechnung*, VEB Deutscher Verlag der Wissenschaften, 1962.
- [70] RIESENFELD, R.F., *Applications of B-spline Approximation to Geometric Problems of Computer Aided Design*, PhD disszertáció, Syracuse University, 1973.
- [71] ROGERS, D. F., ADAMS, J. A., *Mathematical elements for computer graphics*, Second Edition, McGarw-Hill publishing Company, 1990.
- [72] ROJAS, R., *Neural Networks. A Systematic Introduction*, Springer-Verlag, 1996.
- [73] ROUSSEEUW, P. J., LEROY, A. M., *Robust Regression and Outlier Detection*, Wiley, New York, 1987.
- [74] SCHOENBERG, I. Contributions to the problem of approximation of equidistant data by analytic functions, *Quart. Appl. Math*, 4:45-99, 1946.

- [75] SHEPARD, D., A two Dimensional Interpolation Function for Irregularly Spaced Data, Proceedings of the ACM Nat. Conf., pp.517-524, 1965.
- [76] STROMMER GYULA, Geometria, Tankönyvkiadó, Budapest, 1988.
- [77] STRUIK, D. J., Lectures on Classical Differential geometry, Dover Publications, Inc., Reprint, 1988.
- [78] TILLER, W., Rational B-splines for Curve and Surface Representation, IEEE Comp.Graph. and Appl., Vol.3., N.9., pp.36-46., 1983.
- [79] TORKKOLA, K. ET AL., Status Report of the Finnish Phonetic Typewriter Project, in: Kohonen et al (eds.): Artificial Neural Networks, North-Holland, Amsterdam, pp.771-776, 1991.
- [80] VÁRADY, T., MARTIN, R., COX, J., Reverse engineering of geometric models –an introduction, Computer Aided Geometric Design, 29 (4) pp. 255-268, 1997.
- [81] VÁRADY, L., HOFFMANN, M., KOVÁCS, E., Improved Free-form Modelling of Scattered Data by Dynamic Neural Networks, Journal for Geometry and Graphics, Vol.3, No.2, pp.177–181, 1999.
- [82] VÁRADY, L., Analysis of the Dynamic Kohonen Network Used for Approximating Scattered Data, Proceedings of the 7th ICECGDG, Cracow, pp.433–436, 1996.
- [83] WEISS, V., ANDOR, L., RENNER, G., VÁRADY, T., Advanced surface fitting techniques, Computer Aided Geometric Design, 19 pp. 19-42, 2002.
- [84] WEISSTEIN, ERIC W., Point-Line Distance 2-Dimensional, From MathWorld-A Wolfram Web Resource.  
<http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html>
- [85] WEISSTEIN, ERIC W., Point-Line Distance 3-Dimensional. From MathWorld-A Wolfram Web Resource.

<http://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html>

- [86] YAMAGUCHI, F., Curves and Surfaces in Computer Aided Geometric Design, Springer-Verlag, Berlin, 1988.

*További hasznos irodalmak:*

- [87] Arfken, G., Circular Cylindrical Coordinates, §2.4 in Mathematical Methods for Physicists, 3rd ed. Orlando, FL: Academic Press, pp. 95-101, 1985.

- [88] Bíró Sándorné, Szabados Tamás: Vektoranalízis, Műszaki Könyvkiadó, Budapest, 1983

- [89] Bódis Katalin: Geometriai transzformációk alkalmazása a geoinformatikában, 1999, online jegyzet

- [90] H. S. M. Coxeter: A geometriák alapjai, Műszaki Könyvkiadó, Budapest, 1987

- [91] Csepregi Szabolcs: Forgatás  
[http://www.geo.info.hu/gisopen/cd\\_2002/dokumentum/doc\\_html/csepregi\\_sz.htm](http://www.geo.info.hu/gisopen/cd_2002/dokumentum/doc_html/csepregi_sz.htm)

- [92] Fried Ervin: Algebra I. Elemi és lineáris algebra, Nemzeti Tankönyvkiadó, Budapest, 2000

- [93] Hajnal Imre: Matematika a speciális matematika I. osztálya számára, 1994, Nemzeti Tankönyvkiadó, Budapest

- [94] Hajós György: Bevezetés a geometriába, Tankönyvkiadó, Budapest, 1966

- [95] Juhász Imre, Lajos Sándor: Számítógépi grafika, Miskolc, 2007, online jegyzet

- [96] Juhász Imre: OpenGL, Mobidiák könyvtár, 2003, online jegyzet

- [97] Kecskemétiné Pál Éva: Homogén koordináták és transzformációk  
<http://gportal.hu/gindex.php?pg=1898479&nid=480731>
- [98] KOVÁCS E., Rotation and Reflection in simple way, Annales Mathematicae et Informaticae, Volume 3X. (to appear)2011.
- [99] Krammer Gergely:Koordináta-rendszereink az euklideszi térben  
<http://.../grafika/GrafikaKrammer41-54.pdf>
- [100] Reiman István: A geometria és határterületei, Gondolat, Budapest, 1986
- [101] Schwarcz Tibor: Bevezetés a számítógépi grafikába, Mobidiák könyvtár, 2005, online jegyzet
- [102] Székely Vladimir, Poppe András: A számítógépes grafika alapjai IBM PC-n, Computerbooks, Budapest, 1992
- [103] Tevian Dray & Corinne A. Manogue, Conventions for Spherical Coordinates,  
<http://www.physics.oregonstate.edu/bridge/papers/spherical.pdf>, 2002.
- [104] Tornai Róbert: Fejezetek a számítógépi grafikából, Mobidiák könyvtár, 2004, online jegyzet
- [105] Alan Watt, Fabio Policarpo: 3D Games: Real-Time Rendering and Software Technology, New York : ACM Press, 2001
- [106] William H., Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Numerical Recipes: The Art of Scientific Computing, Third Edition , Cambridge University Press, 1256 pp., 2007.
- Ábrák:*
- [107] ábra: <http://www.programmersheaven.com/articles/angelcode/worldview/images/rotate.gif>

[108] ábra: <http://www.programmersheaven.com/articles/angelcode/worldview/images/translate.gif>

[109] ábra: <http://www.programmersheaven.com/articles/angelcode/worldview/images/scale.gif>