

2. CAD RENDSZEREK GEOMETRIAI ALAPJAI

Dr. Mikó Balázs – Hervay Péter – Tóth Georgina Nóra

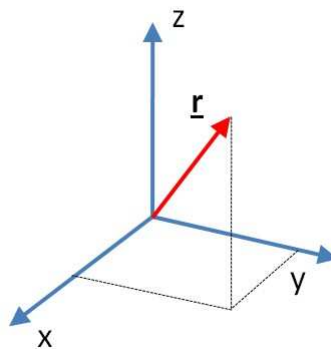
A számítógéppel segített gépészeti tervezés során egy virtuális modellt hozunk létre. Ez a modell több összetevőből áll, egyrészt a geometriai modellből, mely megadja az alkatrészek alakját, másrészt az ezt kiegészítő információkat leíró attribútumokból, mint például az alkatrész anyagminősége.

A modellezés során pontokat, görbéket, felületeket és testeket hozhatunk létre, ezek matematikai leírásával, valamint a hozzájuk kapcsolódó manipulációs eljárásokkal és megjelenítési problémákkal foglalkozik ez a fejezet. Egy CAD rendszer használatához általában nem szükséges ezen matematikai háttér ismerete, azonban segít megérteni és kijavítani bonyolultabb modellek létrehozása során tapasztalható modellezési problémákat, hibaüzeneteket.

2.1 Geometriai elemek ábrázolása

A geometriai elemeket jobbsodrású, derékszögű (un. Descartes-s) koordináta rendszerben ábrázoljuk, a koordináta tengelyek x , y , z (2.1. ábra).

A legegyszerűbb geometriai elem, amit általában segéd elemként használunk a modellezés során, az a **pont**. A pont megadása koordináta értékeinek megadásával történik.



2.1. ábra Pont helyvektora derékszögű jobbsodrású koordináta rendszerben

$$P(x,y,z) \tag{2.1}$$

$$\underline{r} = x \cdot \underline{i} + y \cdot \underline{j} + z \cdot \underline{k} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Görbén folytonos vonalat értünk. Matematikai szempontból a görbe pontok halmaza. Ez a halmaz skalár egyenletekkel vagy vektor egyenletekkel definiálható, amelyeket a görbe pontjai elégítenek ki. Egy két dimenziós (2D-s) görbét felírhatunk *explicit* módon:

$$\begin{aligned} x &= x(t) \\ y &= y(t) \\ t &\in [0,1] \end{aligned} \tag{2.2}$$

vagy *implicit* formában is:

$$f(x,y)=0 \quad (2.3)$$

Például egy (x_o, y_o) középpontú, R sugarú kör explicit egyenlete:

$$\begin{aligned} x &= x_o + R \cdot \cos 2\pi t \\ y &= y_o + R \cdot \sin 2\pi t \\ t &\in [0,1] \end{aligned} \quad (2.4)$$

illetve *implicit* egyenlete:

$$(x - x_o)^2 + (y - y_o)^2 - R^2 = 0 \quad (2.5)$$

Az explicit forma alkalmazása a görbén lévő pontok generálására alkalmas, míg az *implicit* forma segítségével egy adott pont illeszkedését tudjuk ellenőrizni, 0-t kell kapnunk, ha a pont koordinátáit behelyettesítjük az explicit egyenletbe. Mivel számítógépes grafikában az első feladatra van szükségünk, így az explicit alakot használjuk.

3D-s görbét *explicit* formában adhatjuk meg:

$$\begin{aligned} x &= x(t) \\ y &= y(t) \\ z &= z(t) \\ t &\in [0,1] \end{aligned} \quad (2.6)$$

Például egy (x_1, y_1, z_1) ponttól (x_2, y_2, z_2) pontig tartó 3D-s egyenes szakasz *explicit* egyenlete:

$$\begin{aligned} x &= x_1 \cdot t + x_2 \cdot (1-t) \\ y &= y_1 \cdot t + y_2 \cdot (1-t) \\ z &= z_1 \cdot t + z_2 \cdot (1-t) \\ t &\in [0,1] \end{aligned} \quad (2.7)$$

Egy modellezési feladat általában nem oldható meg klasszikus görbeszegmensek alkalmazásával (egyenes, kör, ellipszis stb.). Bármely görbe kellőpontossággal közelíthető egyenes szakaszokkal, azonban így nem mindenhol differenciálható görbét kapunk eredményül, amely bizonyos típusú analízisek esetén nem megfelelő. *Polinomok* alkalmazásával folyamatosan differenciálható görbéket kapunk, melyeket az a_i , b_i polinom együtthatók megadásával specifikálhatunk:

$$\begin{aligned} x(t) &= \sum_{i=0}^n a_i \cdot t^i \\ y(t) &= \sum_{i=0}^n b_i \cdot t^i \end{aligned}$$

$$t \in [0,1] \quad (2.8)$$

A polinom együtthatók nehezen tehetők szemléletessé, így megadásuk egy CAD rendszerben nehézkes, így a legtöbb CAD rendszerben vezérlő pontokat adhatunk meg, a pontos illesztést pedig a program végzi el. Amennyiben a megadott vezérlő pontokon átmegy a görbe, *interpolációról* beszélünk, abban az esetben, ha a megadott pontokat csak közelíti a görbe, *approximációnak* nevezzük. Mindkét módszerre találunk példát mind a matematikai leírások, mind a CAD rendszerek körében.

Lagrange-féle interpoláció esetén tegyük fel, hogy a megadott vezérlőpontok a következők: $\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n$. Keressük azt a minimális fokszámú $L(t)$ polinomot, amely t_1 -nél $\vec{r}_1$, t_2 -nél $\vec{r}_2$ stb. értékeket vesz fel. Az eredmény $(n-1)$ -ed fokú polinom, az $[a_i, b_i]$ együtthatókat pedig a minden $j=1, 2, \dots, n$ vezérlőpontra felírt egyenletrendszer megoldása adja:

$$\vec{r}(t_j) = [x(t_j), y(t_j)] = \sum_{i=0}^{n-1} [a_i, b_i] \cdot t_j^i = \vec{r}_j \quad (2.9)$$

Az egyenletrendszer mindig megoldható, azonban van egyszerűbb megoldás is:

$$\vec{r}(t) = \sum_{i=0}^{n-1} L_i(t) \cdot \vec{r}_i, \text{ ahol } L_i(t) = \frac{\prod_{j \neq i} (t - t_j)}{\prod_{j \neq i} (t_i - t_j)} \quad (2.10)$$

Az $L_i(t)$ függvényt *súlyfüggvénynek* nevezzük. A Lagrange-interpoláció hajlamos az úgynevezett oszcillációra, ami azt jelenti, hogy a súlyfüggvény értéke negatív lesz, a vezérlő pont taszítja a görbét. A másik nehézség, hogy egy vezérlőpont a görbe minden részére hat, vagyis a pontok kis mozgásával a görbe módosított vezérlő pontjától távoli szakasza is módosul.

A **Coons-Hermite interpoláció** az illesztést helyileg, szakaszonként oldja meg, ehhez azonban ismerni kell a kontrollpontokban az érintővektorokat. Két szomszédos kontrollpontot összekötő görbeív analitikus előállítás egy harmadfokú, egyváltozós vektorfüggvény:

$$\vec{r}(t) = a_3 \cdot t^3 + a_2 \cdot t^2 + a_1 \cdot t + a_0 \quad (2.11)$$

Az ív két vége a két kontrollpontra ($\mathbf{p}_0$ és $\mathbf{p}_1$) kell támaszkodjon, a végpontokban az érintő egyenlő kell legyen az adott értékkel ($\mathbf{p}_0^u$ és $\mathbf{p}_1^u$):

$$\begin{aligned} \vec{r}(0) &= \mathbf{p}_0 & \vec{r}(1) &= \mathbf{p}_1 \\ \frac{d\vec{r}}{dt}(0) &= \mathbf{p}_0^u & \frac{d\vec{r}}{dt}(1) &= \mathbf{p}_1^u \end{aligned} \quad (2.12)$$

A harmadfokú súlyfüggvények, melyek kielégítik ezen feltételeket:

$$\begin{aligned}
f_1(t) &= 2t^3 - 3t^2 + 1 \\
f_2(t) &= -2t^3 + 3t^2 \\
f_3(t) &= t^3 - 2t^2 + t \\
f_4(t) &= t_3 - t_2 \\
t &= [0,1]
\end{aligned} \tag{2.13}$$

A görbeív egyenlete tehát:

$$\mathbf{r}(t) = f_1(t) \mathbf{p}_0 + f_2(t) \mathbf{p}_1 + f_3(t) \mathbf{p}_0^u + f_4(t) \mathbf{p}_1^u \tag{2.14}$$

A folytonosság feltétele, hogy a szomszédos íveknek a közös végpontjukban azonos legyen az érintő, a görbület viszont általában nem folytonos.

Az érintők értékének meghatározása kényelmetlen a felhasználó számára, ezért a belső kontrollpontokhoz tartozó érintőt automatikusan határozza meg a program a két szomszédos kontrollpontot összekötő vektorként:

$$\mathbf{r}(t) = f_1(t) \mathbf{p}_i + f_2(t) \mathbf{p}_{i+1} + f_3(t) \mathbf{p}_i^u + f_4(t) \mathbf{p}_{i+1}^u \tag{2.15}$$

ahol

$$\begin{aligned}
\mathbf{p}_0^u &= \mathbf{p}_1 - \mathbf{p}_0 \\
\mathbf{p}_i^u &= \mathbf{p}_{i+1} - \mathbf{p}_{i-1} \\
\mathbf{p}_n^u &= \mathbf{p}_n - \mathbf{p}_{n-1} \\
i &= 1, 2, \dots, n-1
\end{aligned} \tag{2.16}$$

Nyílt görbe esetén a végpontokban az érintő az illető pontot a szomszédos kontrollponttal összekötő vektor.

A Coons-Hermite görbék alkalmazását tovább színesíti, ha a $\mathbf{p}_i^u$ érintővektorokat megszorozzuk egy k_0 állandóval. Ha $k_0 < 1$, a görbék csúcsosabbak lesznek. A $k_0 = 0,5$ érték alkalmazása esetén *Overhauser interpolációról* beszélünk.

A **Bézier interpoláció** célja sima görbék és felületek több kontrollpont segítségével való szerkesztése volt, hogy megkönnyítse a számítógépes formatervezést. Az 1972-ben megalkotott görbe a következő tulajdonságokkal rendelkezik:

- a görbe átmegy a szélső pontokon ($\mathbf{p}_0$ és $\mathbf{p}_n$),
- az érintő a szélső pontokban $\mathbf{p}_1 - \mathbf{p}_0$ és $\mathbf{p}_{n-1} - \mathbf{p}_n$.
- a súlyfüggvények szimmetrikusak legyenek t és $1-t$ -re nézve, azaz a sorrend megfordítása ne befolyásolja a görbe alakját.

A súlyfüggvények meghatározása tehát a görbetervezés kulcsa. Egy fontos súlyfüggvény készlethez jutunk, ha a $(t+(1-t))^n$ binomiális tétel szerinti kifejtésével. E kifejtés egyes tagjait **Bernstein-polinom**nak nevezzük:

$$(t + (1-t))^n = \sum_{i=0}^n \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i} \tag{2.17}$$

$$B_i^{(n)}(t) = \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i} \tag{2.18}$$

A polinom átmegy az első és utolsó vezérlő ponton, de általában nem megy át a többi vezérlőponton. A Bernstein-polinomot alkalmazó approximációs görbét **Bézier-görbének** nevezzük.

Az $n=3$ eset a Bézier-görbék fontos speciális esete, amikor négy kontrollpont harmadrendű görbét eredményez. A harmadrendű görbék a legalacsonyabb rendű görbék melyekkel tér-görbe modellezhető, valamint olyan görbék, amelyeknek inflexiós pontja, csúcspontja vagy önmetszéspontja lehet.

A harmadrendű Bézier-görbe Bernstein-polinom alakja a következő:

$$\begin{aligned}
 b(t) &= \sum_{i=0}^3 b_i \cdot \binom{3}{i} \cdot t^i \cdot (1-t)^{3-i} \\
 b(t) &= b_0 \cdot (1-t)^3 + 3 \cdot b_1 \cdot t \cdot (1-t)^2 + 3 \cdot b_2 \cdot t^2 \cdot (1-t) + b_3 t^3 \\
 b(t) &= b_0 \cdot (-t^3 + 3 \cdot t^2 - 3 \cdot t + 1) + b_1 \cdot (3 \cdot t^3 - 6 \cdot t^2 + 3t) + b_2 \cdot (-3 \cdot t^3 + 3 \cdot t^2) + b_3 t^3
 \end{aligned} \tag{2.19}$$

A kezdő és végpontbeli érintő a következő:

$$\underline{b}'(0) = 3 (\underline{b}_1 - \underline{b}_0) \quad \underline{b}'(1) = 3 (\underline{b}_3 - \underline{b}_2)$$

Mátrix alakban felírva:

$$b(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \underline{b}_0 \\ \underline{b}_1 \\ \underline{b}_2 \\ \underline{b}_3 \end{bmatrix} \tag{2.20}$$

Bonyolult görbéket sok vezérlő ponttal tudunk definiálni. A görbék illesztése során kétféle stratégiát alkalmazhatunk, vagy magas fokszámú polinomot illesztünk a görbére, vagy több kisebb fokszámút. A magas fokszámú polinomok hajlamosak a hullámosodásra, ezért a több, alacsonyabb fokszámú polinomból álló összetett görbéket preferáljuk.

Az illeszkedő szegmenseknek folytonosnak kell lenni, a folytonosságnak több szintje létezik. Két görbe geometriai értelemben 0-ad rendben folytonos (G^0 folytonos), ha az első görbe végpontja megegyezik a második görbe kezdőpontjával. Ugyan ezen tulajdonsággal rendelkeznek a 0-ad rendben paraméteresen folytonos görbék (C^0 folytonos), melyeknél $\vec{r}_1(t_{end}) = \vec{r}_2(t_{start})$. Beszélhetünk mind geometriai, mind parametrikus értelemben magasabb rendű folytonosságról, például G^1 folytonosságról beszélünk, ha a csatlakozó görbék érintői párhuzamosak. Parametrikus értelemben tetszőleges fokszámra általánosítva a definíciót két görbe parametrikusan C^n folytonos illeszkedésű, ha a két görbe deriváltja a vég- illetve kezdőpontban n . deriváltig megegyezik.

A műszaki tervezésben általában C^2 folytonos görbékre van szükségünk, melyek összefoglaló neve **szplájn (spline)**.

A legegyszerűbb polinom, amelynél a második derivált nem állandó, harmadfokú, így a C^2 folytonosságot legegyszerűbben **harmadfokú szplájnnal** teljesíthetjük, melynek általános alakja:

$$p(t) = \bar{a}_3 \cdot t^3 + \bar{a}_2 \cdot t^2 + \bar{a}_1 \cdot t + \bar{a}_0 \tag{2.21}$$

Folytonosságot feltételezve:

$$\begin{aligned}
 \bar{p}(0) &= \bar{a}_0 \\
 \bar{p}(1) &= \bar{a}_3 + \bar{a}_2 + \bar{a}_1 + \bar{a}_0 \\
 \bar{p}'(0) &= \bar{a}_1 \\
 \bar{p}'(1) &= 3 \cdot \bar{a}_3 + 2 \cdot \bar{a}_2 + \bar{a}_1
 \end{aligned} \tag{2.22}$$

Amennyiben az $\bar{r}_1, \bar{r}_2, \dots, \bar{r}_n$ vezérlőpontok úgy illeszkednek a görbeszegmensekhez, hogy az első szegmens $\bar{r}_1$ -től $\bar{r}_2$ -ig tart, a második $\bar{r}_2$ -től $\bar{r}_3$ -ig stb., akkor az i . szegmens paramétereit úgy kell megválasztani, hogy

$$\bar{p}_i(0) = \bar{r}_i \quad \bar{p}_i(1) = \bar{r}_{i+1} \tag{2.23}$$

Ha a görbe C^1 illetve C^2 folytonos, akkor az első és második deriváltak azonosak, vagyis:

$$\bar{p}_i'(1) = \bar{p}_{i+1}'(0) \quad \bar{p}_i''(1) = \bar{p}_{i+1}''(0) \tag{2.24}$$

A kapott lineáris egyenletrendszert megoldva megkaphatjuk a kívánt görbeszegmenseket, azonban az ismeretlenek száma több, mint az egyenletek száma, így végtelen sok megoldás létezik.

Amennyiben a görbe kezdő és végpontjában is megadjuk a deriváltak értékét, a feladat megoldhatóvá válik és eljutunk a **B-szplájn**okhoz, melyek általános alakja az r_i vezérlőpontok mellett B_i súlyfüggvényeket tartalmaz:

$$\bar{r}(t) = B_0(t) \cdot \bar{r}_0 + B_1(t) \cdot \bar{r}_1 + B_2(t) \cdot \bar{r}_2 + B_3(t) \cdot \bar{r}_3 \tag{2.25}$$

A folytonossági és peremfeltételeket figyelembe véve a súlyfüggvények értéke:

$$\begin{aligned}
 B_0(t) &= \frac{(1-t)^3}{6} \\
 B_1(t) &= \frac{1+3 \cdot (1-t)+3 \cdot t \cdot (1-t)^2}{6} \\
 B_2(t) &= \frac{1+3 \cdot t+3 \cdot (1-t) \cdot t^2}{6} \\
 B_3(t) &= \frac{t^3}{6}
 \end{aligned} \tag{2.26}$$

A B-szplájn approximációs görbe, vagyis nem megy át a vezérlő pontokon, lokálisan vezérelhető, vagyis egy vezérlő pont mozgatása csak az adott görbeszakaszra van hatással.

A szplájnok általánosításával rugalmasabban alkalmazható görbékhez jutunk, például a B-splinet kiterjeszthetjük oly módon, hogy az egymást követő szegmensek eltérő méretű paramétertartományt fednek el (eddig feltételeztük, hogy minden szegmens paramétertartományt fednek el).

mánya egységnyi). A B-szplájn ezen változatát nem uniform B-szplájnnak (*non-uniform B-spline* vagy *NUBS*) nevezzük. Amennyiben a súlyfüggvény két polinom hányadosa is lehet, racionális B-szplájnt kapunk (*rational B-spline* vagy *RBS*). A két kiterjesztést egyszerre alkalmazva kapjuk a nem-uniform racionális B-szplájnt (*non-uniform rational B-spline* vagy *NURBS*).

A **3D felületek**, a görbékhez hasonlóan definiálhatók explicit egyenletekkel:

$$\begin{aligned} x &= x(u, v) \\ y &= y(u, v) \\ z &= z(u, v) \\ u, v &\in [0, 1] \end{aligned} \quad (2.27)$$

vagy implicit egyenlettel:

$$f(x, y, z) = 0 \quad (2.28)$$

Például egy (x_o, y_o, z_o) középpontú, R sugarú gömb explicit egyenlete:

$$\begin{aligned} x &= x_o + R \cdot \cos 2\pi u \cdot \sin \pi v \\ y &= y_o + R \cdot \sin 2\pi u \cdot \sin \pi v \\ z &= z_o + R \cdot \cos \pi v \\ u, v &\in [0, 1] \end{aligned} \quad (2.29)$$

illetve implicit egyenlete:

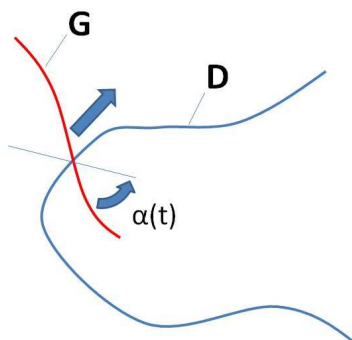
$$(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2 - R^2 = 0 \quad (2.30)$$

Kvadratikus felületeket hozhatunk létre akkor, ha bármely változó legfeljebb másodfokú alakban szerepelhet. Ezen felületek homogén koordináták alakban egyszerűen kezelhetők, ahol Q egy 4×4 -es konstans együttható mátrix. Ilyen formában kezelhető a gömb, hengerpálást, kúp, hiperboloid, paraboloid stb. felületek.

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \underline{Q} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0 \quad (2.31)$$

Ezen analitikus felületekkel azonban nem írható le a gépipari alkatrészek határoló felületei, gyakran kell alkalmaznunk úgynevezett bonyolult vagy szabad formájú felületeket. Három csoport különböztethető meg: (1) transzlációs felületek, (2) vonal felületek, (3) szoborfelületek.

A **transzlációs felületek** úgy hozhatók létre, hogy egy rendszerint síkbeli nyitott vagy zárt görbe (direktrix D) mentén egy másik görbe (generátor G) úgy mozdul el, hogy általános kúp- vagy hengeralkotóként működik, vagy hajlásszögét meghatározott módon változtatja. A felület meghatározásához tehát két görbe megadására és az esetleges „billegés” definiálása szükséges. Transzlációs felületként analitikus felületek is megadhatók, mint például sík, gömb, henger, kúp stb.

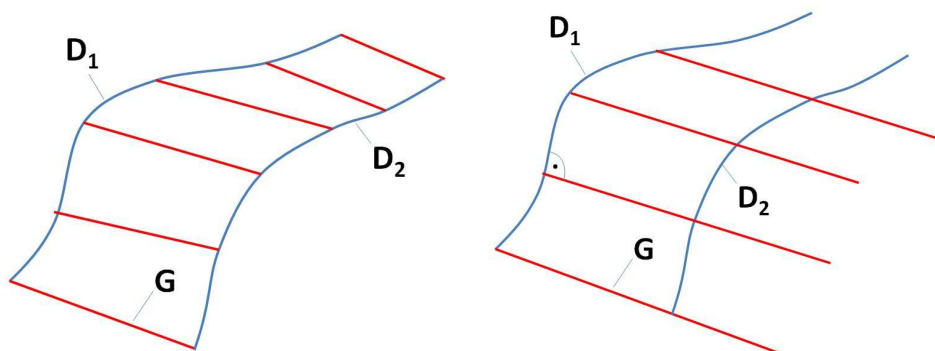


2.2. ábra Transzlációs felület létrehozása

Vonalfelületek létrehozásához egy generátor (G) térgörbére és két direktrix térgörbére (D_1 , D_2) van szükségünk. A generálás sokféle módon valósítható meg (2.3. ábra).

Első esetben a két direktrixen úgy halad végig az egyenes generátor görbe, hogy D_1 és D_2 görbe kezdőpontjából egyszerre indul a generátor egyenes, és a görbék végpontjába egyszerre érkezik meg. A második esetben az egyik direktrixnek kitüntetett szerepe van (pl. D_1), melyre a generátor egyenes mindig merőleges.

További altípusokat hozhatunk létre, ha a generátor görbe nem egyenes, illetve ha a generátor görbe a direktrixek mentén folyamatosan változtatja az alakját.



2.3. ábra Vonalfelület generálása

Szoborszerű felületekről akkor beszélünk, amikor a felület sem analitikusan, sem görbék mozgásával nem írható le. Ezen felületek leírására többféle technika létezik, melyek közös jellemzője, hogy térbeli ponthálóra simítják a felületet.

A **parametrikus felületek** kétváltozós polinomok:

$$\vec{r}(u, v) \quad u, v \in [0, 1] \quad (2.32)$$

A polinomokat vezérlő pontok segítségével, súlyfüggvényekkel állíthatjuk elő, hasonlóan a görbéknél látottakhoz:

$$\bar{r}(u, v) = \sum_{i=0}^m \sum_{j=0}^n \bar{r}_{ij} \cdot B_{ij}(u, v) \quad u, v \in [0, 1] \quad (2.33)$$

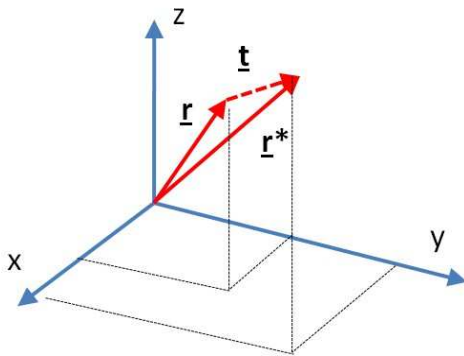
A Bézier-görbék súlyfüggvényéből képzett szorzatot használva a $B_{ij}(u, v)$ súlyfüggvényként megkapjuk a **Bézier-felületet**, melynek így a súlyfüggvénye:

$$B_{ij}(u, v) = \binom{n}{i} \cdot u^i \cdot (1-u)^{n-i} \cdot \binom{m}{j} \cdot v^j \cdot (1-v)^{m-j} \quad (2.34)$$

2.2 Geometriai transzformációk

A definiált geometriai elemek megváltoztatását hívjuk transzformációnak, mely elengedhetetlen a modellezés során. A különböző transzformációkat egy pont manipulálásán keresztül mutatjuk be, mivel bonyolultabb geometriai elemek transzformációja ezen műveletek pontonkénti ismétlésével oldható meg.

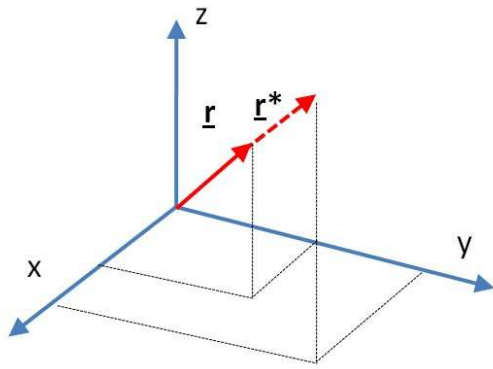
Eltolás során az $\underline{r}$ helyvektorhoz egy $\underline{t}$ eltolás vektort adunk, így kapva meg az $\underline{r}^*$ eltolt helyvektort (2.4. ábra).



$$\underline{r}^* = \underline{r} + \underline{t} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} x + t_1 \\ y + t_2 \\ z + t_3 \end{bmatrix}$$

2.4. ábra Eltolás, eltolási vektor

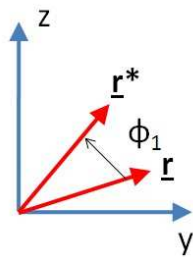
Skálázás vagy **nyújtás** során a koordináta értékeket egy konstanssal szorozzuk (2.5. ábra). A konstansok értéke 0-nál nagyobb, C_1 , C_2 , C_3 értéke lehet egyenlő, de különböző is. A skálázás mátrix szorzásként hajtható végre.



$$\underline{r}^* = \underline{\underline{C}} \cdot \underline{r} = \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} C_1 \cdot x \\ C_2 \cdot y \\ C_3 \cdot z \end{bmatrix}$$

2.5. ábra Skálázás, skálázási mátrix

Az elforgatás legegyszerűbb módja az x_i koordináta tengely körüli elforgatás φ_i szöggel. Az elforgatott pont koordinátáit az $\underline{E}_i$ forgatómátrixszal való szorzás adja (2.6. ábra).



$$\underline{E}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi_1 & -\sin \varphi_1 \\ 0 & \sin \varphi_1 & \cos \varphi_1 \end{bmatrix}$$

$$\underline{E}_2 = \begin{bmatrix} \cos \varphi_2 & 0 & \sin \varphi_2 \\ 0 & 1 & 0 \\ -\sin \varphi_2 & 0 & \cos \varphi_2 \end{bmatrix}$$

$$\underline{E}_3 = \begin{bmatrix} \cos \varphi_3 & -\sin \varphi_3 & 0 \\ \sin \varphi_3 & \cos \varphi_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Forgatás x tengely körül: $\underline{r}^* = \underline{E}_1 \underline{r}$

Forgatás y tengely körül: $\underline{r}^* = \underline{E}_2 \underline{r}$

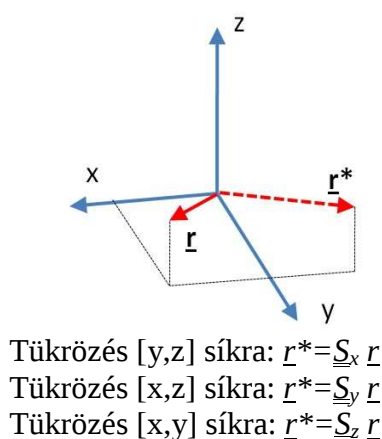
Forgatás z tengely körül: $\underline{r}^* = \underline{E}_3 \underline{r}$

2.6. ábra Forgatás koordináta tengely körül, forgatási mátrixok

Amennyiben több tengely mentén kell elforgatni az objektumot, az elforgatást tengelyenként sorban tudjuk elvégezni. Például x_2 majd x_1 tengely körüli elforgatás esetén:

$$\underline{r}^* = \underline{E}_1 \cdot \underline{E}_2 \cdot \underline{r} \quad (2.35)$$

Koordináta síkra való tükrözés esetén a kijelölt síkra merőleges koordináta értéket kell -1-gyel szorozni. A transzformáció mátrixszorzásként az $\underline{S}_i$ tükrözési mátrixszal oldható meg (2.7. ábra).



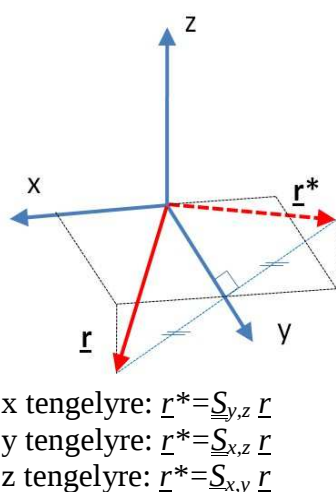
$$\underline{S}_x = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underline{S}_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underline{S}_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

2.7. ábra Tükrözés koordináta síkra, tükrözési mátrixok

Koordináta tengelyre való tükrözés esetén a tükrözési tengelyen kívüli koordináta értékeket kell -1-gyel megszorozni, amely az $\underline{S}_{i,j}$ tükrözési mátrixszal való szorzást jelent (2.8. ábra).



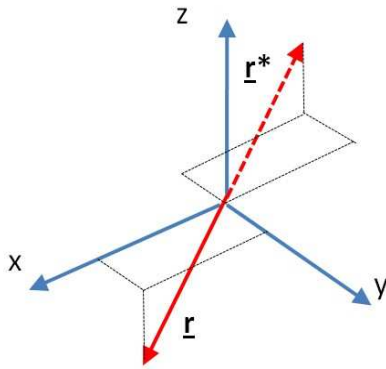
$$\underline{S}_{y,z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\underline{S}_{x,z} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\underline{S}_{x,y} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.8. ábra Tükrözés koordináta tengelyre, tükrözési mátrixok

A koordináta rendszer középpontjára, vagyis az **origóra tükrözés** esetén mindhárom koordináta értéke előjelet vált. Az $\underline{S}$ tükrözési mátrix adja az új pont koordinátáit (2.9. ábra).



$$\underline{r}^* = \underline{S} \underline{r}$$

$$\underline{S} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

2.9. ábra Tükrözés az origóra, tükrözési mátrix

A transzformációk egy kivételével mátrix szorzásként végezhető. Több transzformáció végrehajtása esetén azonban az eltolás nehezíti ezt a műveletet. A probléma kiküszöbölésére vezethetjük be a **homogén transzformációs mátrix**okkal végzett műveleteket. Ezen mátrixok 4x4-es mátrixok, a helyvektorok pedig kiegészülnek egy negyedik, 1 értékű elemmel. Ennek megfelelően az eltolás a következő képen alakul:

$$\underline{r}^* = \underline{r} + \underline{t} \rightarrow \underline{R}^* = \underline{T} \underline{R} \quad (2.36)$$

ahol $\underline{R}$ a kiterjesztett helyvektor, $\underline{T}$ pedig az eltolás homogén transzformációs mátrixa:

$$\underline{R} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \underline{T} = \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.37)$$

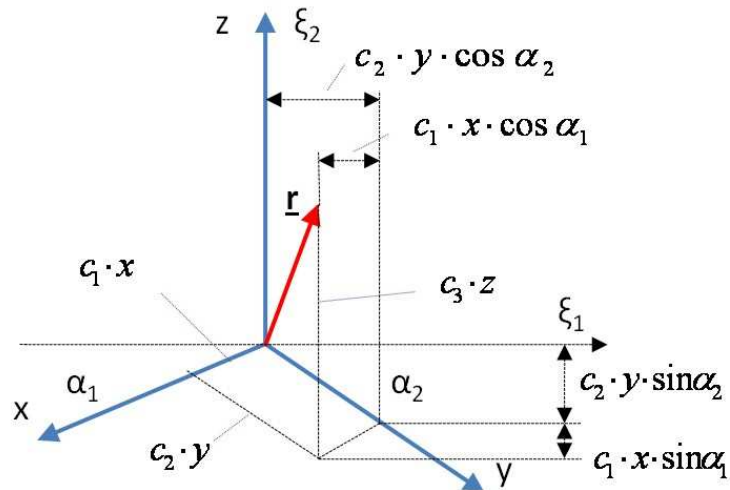
Ennek megfelelően az x_1 tengely körüli elforgatás homogén transzformációs mátrixa:

$$\underline{F}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi_1 & -\sin \varphi_1 & 0 \\ 0 & \sin \varphi_1 & \cos \varphi_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.38)$$

2.3 Térbeli alakzat síkban ábrázolása

A virtuális térben létrehozott 3D-s modellt a számítógép képernyőjének 2D-s síkjában kell megjeleníteni, tehát egy síkba vetítést kell először alkalmazni. A síkbeli ábrázolás általános esetét mutatja a 2.10. ábra.

A c_1, c_2, c_3 tengelyléptékek a koordináta tengelyek rövidülését mutatják, az α_1, α_2 szögek pedig az x, y tengelyek vetített képe és a ξ_1 tengely által bezárt szöget.



2.10. ábra Térbeli pont ábrázolása síkban.

Az ábra alapján könnyen belátható, hogy a 2D-s koordináta-rendszerben a 3D-s pont koordinátái a következők:

$$\begin{aligned}\xi_1 &= c_2 \cdot y \cdot \cos \alpha_2 - c_1 \cdot x \cdot \cos \alpha_1 \\ \xi_2 &= c_3 \cdot z - c_2 \cdot y \cdot \sin \alpha_2 - c_1 \cdot x \cdot \sin \alpha_1\end{aligned}\quad (2.39)$$

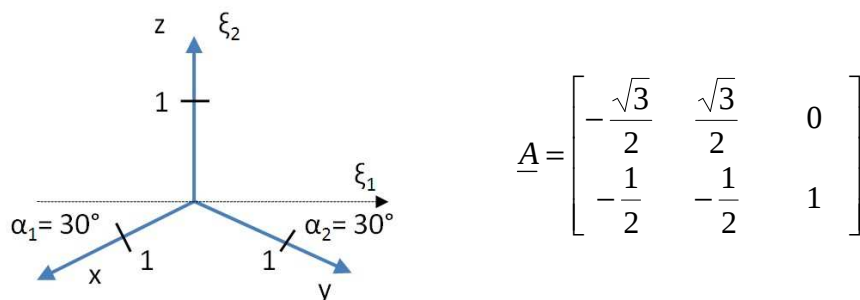
Az eddigiekhez hasonlóan ez a transzformáció is felírható mátrix műveletként, ahol a $\underline{\rho}$ 2D-s vektort az $\underline{A}$ axonometria mátrix és az $\underline{r}$ vektor szorzata adja:

$$\underline{\rho} = \underline{A} \underline{r}$$

$$\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} -c_1 \cdot \cos \alpha_1 & c_2 \cdot \cos \alpha_2 & 0 \\ -c_1 \cdot \sin \alpha_1 & -c_2 \cdot \sin \alpha_2 & c_3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}\quad (2.40)$$

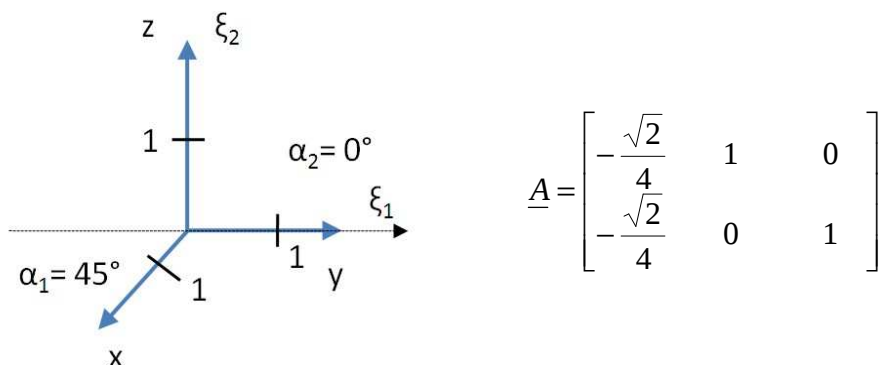
A c_1, c_2, c_3 , valamint az α_1, α_2 szögek speciális megválasztásával különféle axonometrikus ábrázolásokat tudunk létrehozni, valamint a számítógép képernyőjén a nézőpont dinamikus változtatásával ezen paraméterek változnak.

Izometrikus axonometriáról beszélünk, ha $\alpha_1 = \alpha_2 = 30^\circ$ és $c_1 = c_2 = c_3 = 1$, tehát x, y tengelyek szimmetrikusan helyezkednek el és a tengelyeken nincs rövidülés (2.11. ábra).



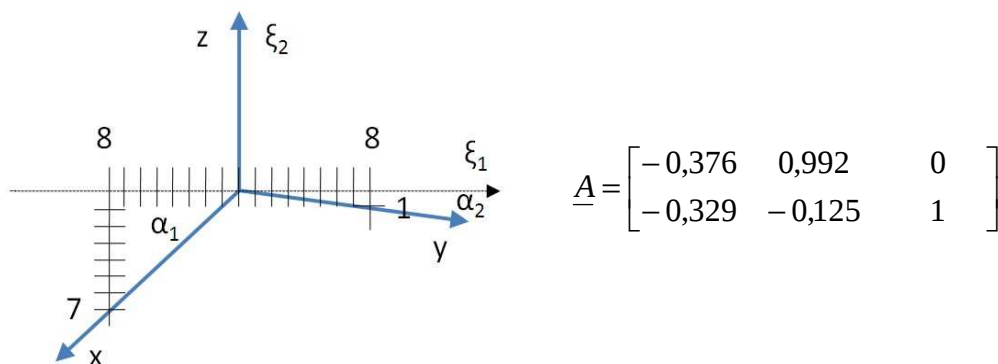
2.11. ábra Izometrikus axonometria és transzformációs mátrixa.

Frontális axonometria esetén az x_2 tengely egybe esik ξ_1 tengellyel, x_1 tengely 45° -os szöget zár be, az y, z tengelyeken nincs rövidülés, míg x tengelyen a rövidülés $\frac{1}{2}$ (2.12. ábra). Tehát: $\alpha_1 = 45^\circ$; $\alpha_2 = 0^\circ$; $c_1 = 1/2$; $c_2 = c_3 = 1$.



2.12. ábra Frontális axonometria és transzformációs mátrixa.

Dimetrikus axonometria esetén az x tengely hajlásszöge $\alpha_1 = \arctg 7/8 = 41^\circ 10'$, az y tengelyé $\alpha_2 = \arctg 1/8 = 7^\circ 10'$, a tengelyek mentén nincs rövidülés, tehát $c_1 = c_2 = c_3 = 1$ (2.13. ábra).



2.13. ábra Dimetrikus axonometria és transzformációs mátrixa.

2.4 Megjelenítés

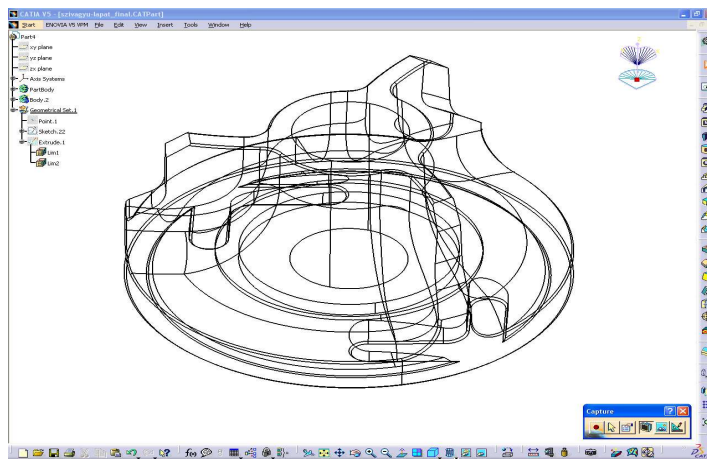
A számítógépes modellezés segítségével a tervezés során képet kaphatunk az elkészítendő objektumról. A korábbi 2D-s tervezést felváltotta a 3D-s, az új megjelenítési technikák és algoritmusok elterjedésével. Egyre inkább arra törekednek, hogy a megjelenítendő objektum képe minél valóság hűbb és szemléletesebb legyen. Ennek érdekében lehetőséget kell biztosítani, hogy különböző fényviszonyok mellett is megtekinthető legyen az objektum. Különböző árnyalási és árnyékolási technikák alkalmazhatóak valamint a fényviszonyok mellett beállítható az megjelenítendő objektum anyaga (textúra) is.

A továbbiakban a 3D-s megjelenítés néhány típusáról, technikájáról lesz szó valamint az objektum megjelenítési beállításairól.

Drótvázás megjelenítés

A 3D-s modell megjelenítése során először a szabadformájú elemek közelítését végezzük el. A drótvázás megjelenítés segítségével a test éleit megjelenítve kaphatunk képet az objektum felépí-

tésérő (2.14. ábra). Problémát jelenthet azonban, hogy gyakran ebben a formában nem lehet egyértelműen megjeleníteni az ábrázolt testet, mivel nem kivehető, hogy melyek a látszó és a takart élek. A megjelenítés előnye viszont a gyors megjelenítés, rövid számítási idő.



2.14. ábra Drótvázis modell megjelenítése CATIA v5 tervezőrendszer segítségével

Különösen szimmetrikus testek esetében a *renderelt* kép teszi világossá a test éleinek elhelyezkedését, egyértelmű képét [4.].

Takarási feladatok megoldása

3D-s modell esetén el kell döntenünk, hogy melyek azok az élek, amelyek megjelennek egy adott képen és melyek azok, amelyek a nézőpontból nem láthatók.

Több olyan algoritmus is ismert, amelyek segítségével ez a probléma hatékonyan megoldható. A láthatósági algoritmusok segítségével lehetőség van egy modellezett objektum látható éleinek és felületeinek kiválasztására. Egy meghatározott nézőpont mellett megadható, hogy az elkészített objektum melyik része látható, melyek a takart és a megjelenítendő élek [6.][7.].

A továbbiakban néhány ismert, a takarási feladatok megoldására alkalmazott módszer rövid leírása látható.

Sugárkövető algoritmus (ray-tracing)

A sugárkövető algoritmus során a nézőpontból a fotonok útját követve határozhatjuk meg, hogy az objektumnak mely részei láthatóak. Az algoritmus segítségével a képernyő pixeleire egyenként végezhetőek el az árnyalási és takarási feladatok [6.].

Az algoritmus működéséhez definiálni kell egy nézőpontot valamint egy ablakot egy tetszőleges vetítési síkon. Az ablak minden pixelébe a nézőpontból képzeletbeli vetítősugarakat indítva a nézőponthoz legközelebb elhelyezkedő objektum határozza meg a pixel színét. A legfontosabb követelmény ezekkel az algoritmusokkal szemben, hogy az említett metszéspontok kiszámítására alkalmas legyen. A sugárkövető algoritmus nagy számítási feladatai miatt időigényes, ezért a metszéspontok kiszámításának idejét próbálják csökkenteni, a módszer hatékonyságának növelésével.

Rekurzív sugárkövető algoritmus

A számítógépes grafikában az egyik legelterjedtebben alkalmazott algoritmus. Alkalmazása során lényegében abban különbözik az előbb ismertetett sugárkövető algoritmustól, hogy a nézőpontba indított sugarak nyomon követése mellett másodlagos sugarakat is használ a pixel színének megha-

tározása során. Ezek lehetnek fényforrással összekötő sugarak, megtörő és visszatükröződött sugarak [6.][7.].

Az algoritmus számítási igénye csökkenthető több módszerrel. Az egyik lehetőség, hogy megfordítva a sugarakat, amelyekkel számolunk, csak azokkal foglalkozunk, amelyek eljutnak az ablak celláin keresztül a nézőpontba.

A másik lehetőség, hogy ha úgynevezett befoglaló testeket (legtöbbször gömböket) definiálva csökkentjük a kiszámítandó metszéspontok mennyiségét azzal, hogy csak azokban az esetekben számítjuk ki azokat, ha a befoglaló gömböt is metszi a sugár [7.].

Z-buffer algoritmus

A z-buffer algoritmus egyszerűen, gyorsan használható, nagy memória igényű, azonban bármilyen alakú 3D-s objektum ábrázolására alkalmazható. Minden egyes pixelnek a színét meghatározza, majd az eredményt kirajzolja.

Az algoritmus két változót használ. Az egyikben tárolja a pixelek színét (frame-buffer), a másikban a nézőponttól való távolság értékét (z-buffer). Alapbeállításként a pixelek színe a háttér színével megegyezik.

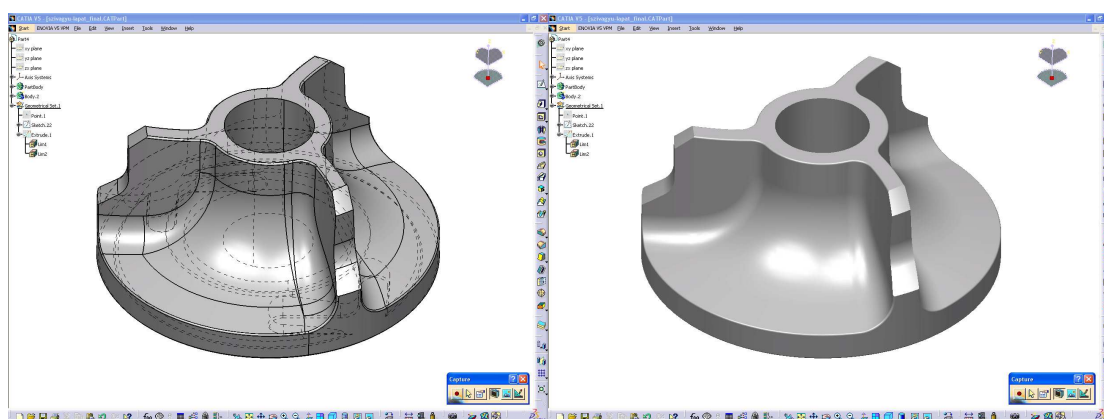
Minden megjelenítendő felület pontjaihoz meghatározza az algoritmus a z értéket és amennyiben ez nagyobb az eddig az adott pixelre kiszámított értéknél (z-buffer), akkor ezzel az értékkel felülírjuk a z-buffer értékét valamint a megfelelő színértéket beállítjuk a frame-bufferben.

Az algoritmus nagy előnye, hogy független a megjelenítendő objektum alakjától, mindössze a nézőponttól való távolságok meghatározhatóságára van szükség valamint az objektum színével, textúrájával valamint a megvilágításával kapcsolatos információkra.

Hátránya a nagy számítás és erőforrásigénye és nehéz kombinálhatósága más eljárásokkal [6.][7.].

Árnyalás (Shading)

A modell megrajzolását követően a kiszínezését kell elvégezni a valósághű megjelenítés érdekében. Ehhez 3D-s modell készítése esetén tisztában kell lennünk a fényviszonyokkal és annak függvényében határozható meg az egyes képpontok színe. (2.15. ábra)

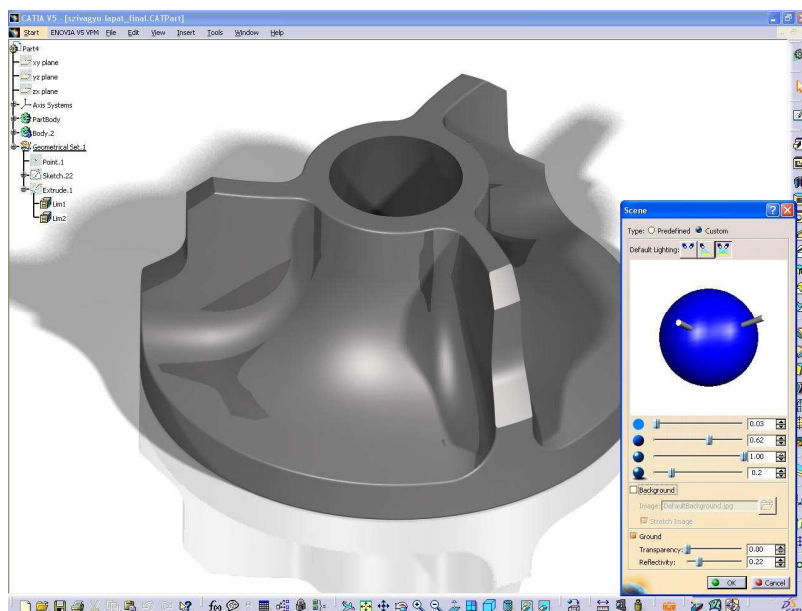


2.15. ábra Ford szivattyúlapát 3D modell árnyalása CATIA v5 tervezőrendszer segítségével

Az árnyalás során a pixelek színének meghatározása és ábrázolása nagy számításigényű, bonyolult feladatot jelenthet. Az idő és erőforrás igény csökkentése érdekében több algoritmus is használható 3D-s megjelenítés során [5.][6.][7.].

Fények

A modell megjelenítése során több eltérő fényforrást is elhelyezhetünk. A fényforrások elhelyezkedése valamint a kibocsátott fény intenzitása közvetlenül hatással van a megjelenítendő képre illetve a modellben ábrázolt árnyékokra (2.16. ábra) [5.][6.][7.].



2.16. ábra Ford szivattyúlapát 3D modell CATIA v5 tervezőrendszer segítségével (fények beállítása)

Színek

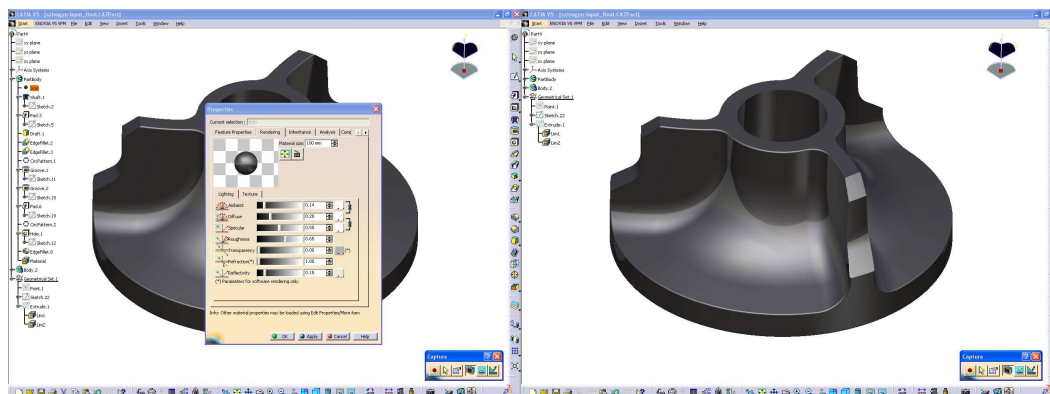
A színek megjelenítéséhez többféle színrendszer terjedt el. Alapvetően a színek 3 skalár mennyiséggel jellemezhetőek. Az egyes képpontok színkódjának meghatározása egy bázis koordináta-rendszer meghatározása szükséges, amelynek megváltoztatásával a színek is más-más módon adhatók meg. Alapvetően három különböző színrendszer ismert a számítógépes grafika területén. Az RGB színrendszer használata során a vörös (Red), a zöld (Green) és a kék (Blue) színek mennyiségét határozzuk meg, ezzel definiálva a megjelenítendő színt. A CMY színrendszer a nyomtatáshoz kapcsolható, ebben az esetben a cián (Cyan), a magenta (Magenta) és a sárga (Yellow) színek mértékét kell meghatározni.

A HLS színrendszer használata során nem alapszínek mennyiségét, hanem természetesebb mennyiségeket használ. Megadható egy szín ebben a rendszerben a színárnyalat, fényesség és telítettség beállításával [7.].

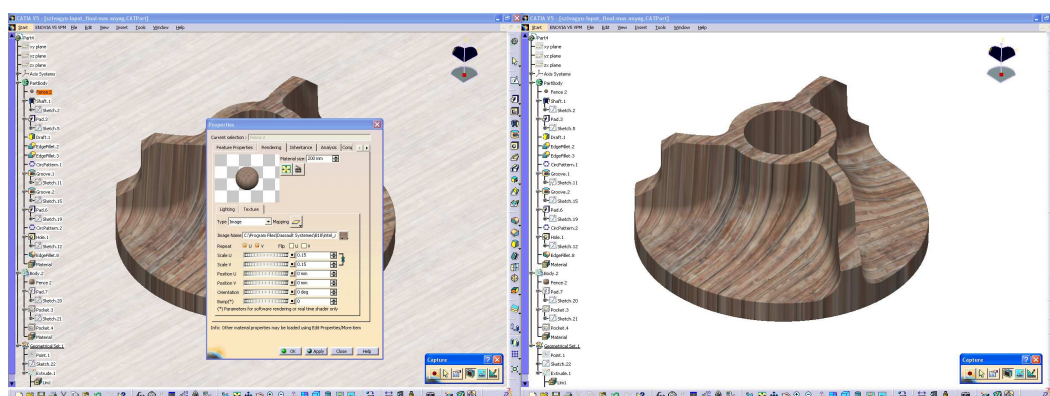
Anyagok, textúrák

Egy modell megjelenítése során fontos szempont az anyagának meghatározása. A felület az anyagválasztás alapján bizonyos tulajdonságokat vesznek fel, amelyek három különböző szinten jelennek meg. Figyelembe kell venni az anyag tükröződéssel kapcsolatos tulajdonságait, a textúrát, mintát valamint ügyelni kell az eltérő geometriák megjelenítésére.

Alapvetően az anyagok definiálásának alapja leírható egy tükröződési függvény segítségével [5.].



2.17. ábra Ford szivattyúlapát 3D modell textúrájának (vas) beállítása CATIA v5 tervezőrendszer segítségével



2.18. ábra Ford szivattyúlapát 3D modell textúrájának (fa) beállítása CATIA v5 tervezőrendszer segítségével

A fenti ábrákon a Ford szivattyúlapát textúrájának beállítási lehetőségei és megjelenítése látható. A 2.17. ábra a termék vasból készült változatát jeleni meg, a 2.18. ábra pedig a fa anyagból modellezve.

Renderelés

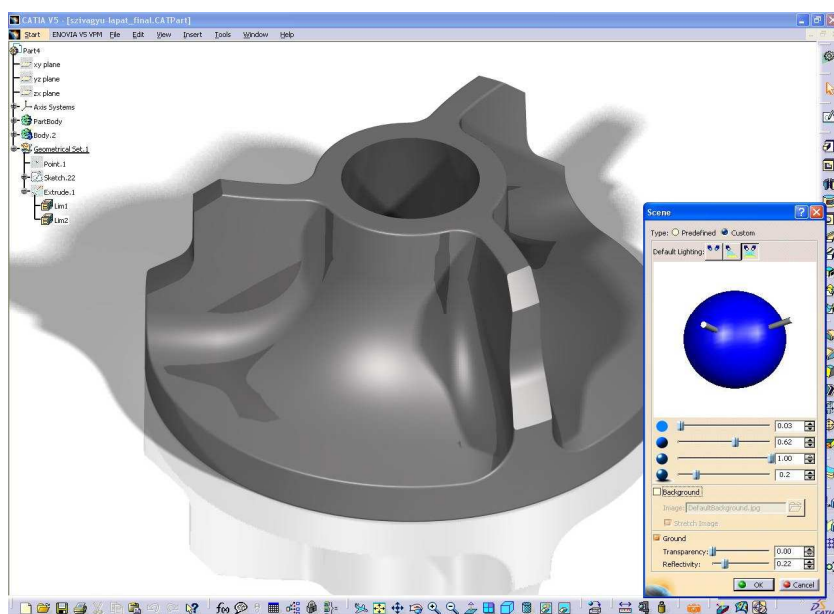
A számítógépes megjelenítés során több tevékenységet kell elvégeznünk, hasonlóan egy fénykép elkészítéséhez. Először is létre kell hozni a geometriai modellt. Ennek elkészítésekor figyelembe kell venni, hogy bizonyos kényszerektől mentesíteni kell a modellt, illetve az átmenetek megtervezésére különösen érdemes odafigyelni. A tervezéshez használt szoftverek szinte mindegyike manapság alkalmas általános fájlformátumok használatára, amelyek segítségével egyszerűvé válik a hordozhatóság (IGES, STL).

A második lépés az objektum felszínének kialakítása, színének, anyagának beállítása. A megjelenítés során az egyes anyagok esetén eltérő a fény visszaverődése. A legtöbb manapság használt szoftver rendelkezik előre definiált anyagokkal és szükség esetén bizonyos paraméterek megadásával lehetőség van további anyagtipusok megjelenítésére is.

A harmadik lépés a modell környezeti viszonyainak beállítása, amelynek során definiálhatóak a fényviszonyok, beállíthatóak az árnyékok valamint a környezet elemei, ahol a modellt meg kívánjuk jeleníteni. Ezzel a kép hangulata is jelentős mértékben befolyásolható.

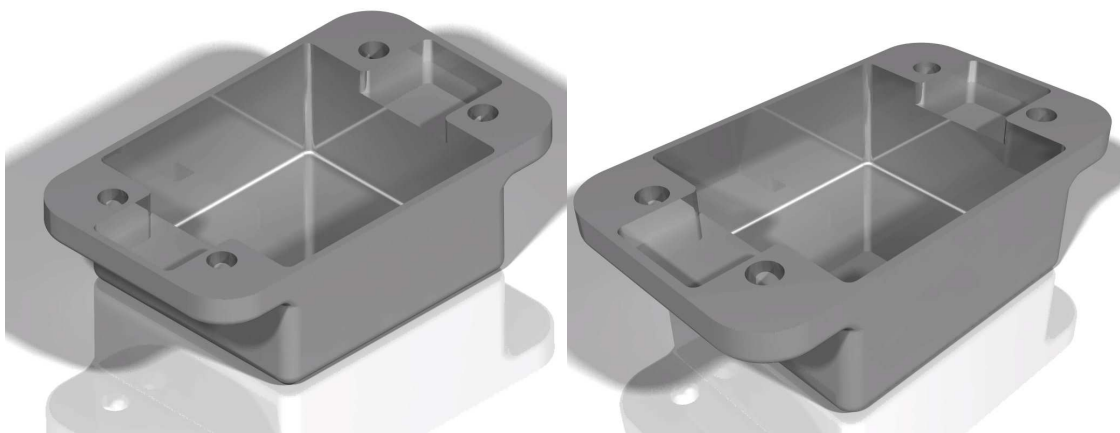
Az utolsó lépés a kép elkészítése (2.19. ábra), amelyhez egy virtuális kamerát hívunk segítségül. A kép elkészítésekor hasonló beállításokat lehet alkalmazni, mint egy fényképezőgép használatakor.

Bár a beállítások időigényesek lehetnek, azonban meghatározóak a modelltől készített kép minősége szempontjából [5.].



2.19. ábra Renderelt kép, beállítási lehetőségek a CATIA v5-ben

A korábban felsorolt lépések egymással összefüggésben vannak, az egyes lépések során történő módosítások meghatározhatják a további beállítási lehetőséget és a megjelenítendő kép minőségét egyaránt [5.].



2.20. ábra Renderelt kép párhuzamos és perspektivikus ábrázolása

További lehetőség van a renderelt kép más-más megjelenítési ábrázolására (2.20. ábra), amelynek segítségével ugyanaz a modell látható párhuzamos és perspektivikus nézetben.

2.5 Irodalom

- [1.] Horváth Imre – Juhász Imre: Számítógéppel segített gépészeti tervezés 1.; Műszaki Könyvkiadó Budapest 1996.
- [2.] Szirmay-Kalos László: Számítógépes grafika; ComputeBooks Budapest 2000.
- [3.] Füzi János: Interaktív grafika; ComputerBooks Budapest 1997.

- [4.] Bognár Géza, Kaczur Sándor: Térbeli gondolkodás segítése elektronikus eszközökkel; Informatika a felsőoktatásban 2008. Debrecen, 2008. augusztus 27-29. (www.agr.unideb.hu/if2008/kiadvany/papers/C42.pdf)
- [5.] Dr. Váradi Károly: CAD Technológiák Magas szintű tervezés támogatás; BME Budapest, 2006.
- [6.] Dr. Szirmay – Kalos László: Számítógépes grafika; ComputerBooks, Budapest, 2002.
- [7.] Schwarcz Tibor: Bevezetés a számítógépi grafikába; (<http://mobidiak.inf.unideb.hu/>) mobiDIÁK könyvtár, 2005.