

# **TDK-dolgozat**

Nagy Ferenc

Debreceni Egyetem, Informatikai Kar,  
Komputergrafika és Képfeldolgozás Tanszék

# **Izooptikus görbéken alapuló módszer kamera mozgatási pályájának optimalizálására**

**Témavezető:**

Kunkli Roland, egyetemi tanársegéd  
Dr. Hoffmann Miklós, főiskolai tanár

**Szerző:**

Nagy Ferenc  
III. évfolyam

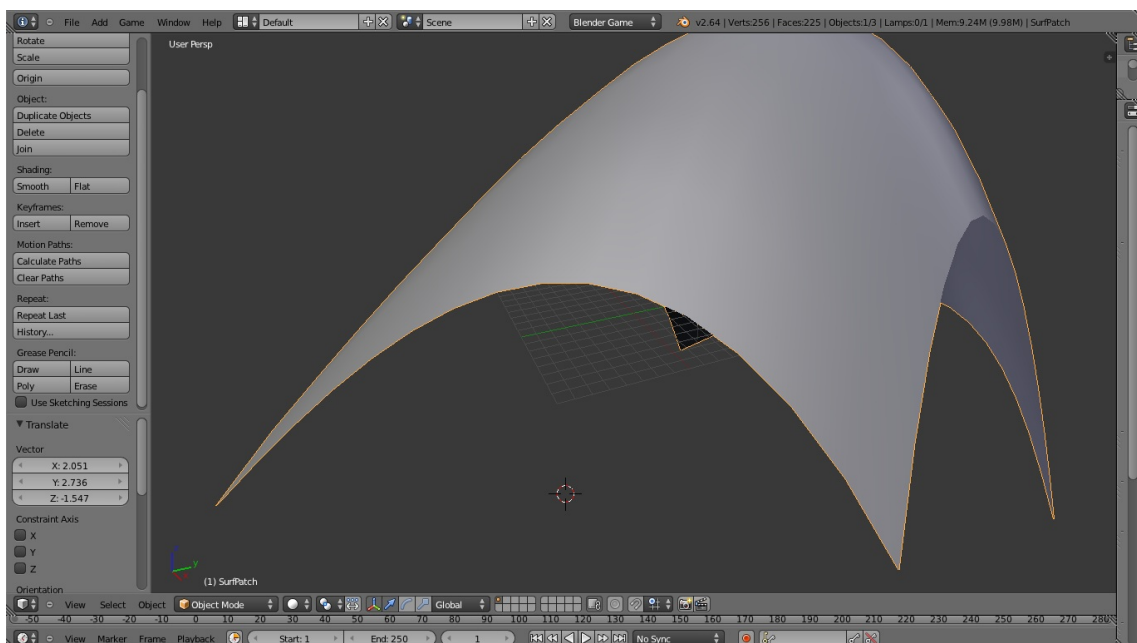
Debrecen, 2012

# Tartalomjegyzék

|                                                                                          |           |
|------------------------------------------------------------------------------------------|-----------|
| <b>Előszó</b>                                                                            | <b>4</b>  |
| <b>1. Görbék izooptikusa</b>                                                             | <b>6</b>  |
| 1.1. Bézier-görbére felírt izooptikus kereső módszerrel . . . . .                        | 6         |
| 1.2. NURBS görbére felírt izooptikus kereső módszerrel . . . . .                         | 8         |
| 1.3. Izooptikus görbe egzakt számolása . . . . .                                         | 10        |
| <b>2. Felületek izooptikusa</b>                                                          | <b>12</b> |
| 2.1. Felület elmetszése síkokkal . . . . .                                               | 12        |
| 2.2. Normálvektoros keresés kontrollpont forgatással . . . . .                           | 13        |
| 2.3. Normálvektoros keresés $\mathbf{m}$ és $\mathbf{v}$ vektorok forgatásával . . . . . | 15        |
| 2.4. Izooptikus keresés lineáris kombinációval . . . . .                                 | 16        |
| <b>Összefoglalás</b>                                                                     | <b>18</b> |
| <b>Köszönetnyilvánítás</b>                                                               | <b>19</b> |
| <b>Irodalomjegyzék</b>                                                                   | <b>20</b> |
| <b>Saját munka leírása</b>                                                               | <b>21</b> |

## Előszó

Kisebb tárgyakat akár a kezünkben tartva vizsgálhatunk. Nagyobb objektumok esetén azonban előfordulhat, hogy hátrébb kell lépünk, hogy az egész tárgy benne legyen a látómezőnkben. Ha fényképezünk valamit, akkor is fontos lehet, hogy olyan távol álljunk, hogy az egész tárgy benne legyen a képben, illetve úgy közelítsünk hozzá, hogy minél inkább betöltse azt. Számítógépes megjelenítésben ez azt jelenti, hogy arra kell törekednünk, hogy a vizsgált objektum elférjen a képernyőn. Sokunk által tapasztalt probléma, például különböző modellező programok használata esetén, hogy a vizsgált tárgy elforgatva vagy más szemszögből nézve „kilóg a képernyőről”, egyes részei nincsenek megjelenítve, ahogy például az 1. ábrán is látható.



1. ábra. A felület nem fér el a képernyőn

Az ilyen szoftverekben a Bézier és B-spline görbék, illetve felületek alapvető szerepet töltenek be, a leggyakrabban használt modellező eszközök közé tartoznak. Ezért a tárgyalni kívánt módszerek is ezekkel a görbékkel és felületekkel foglalkoznak.

Kétdimenziós esetben tetszőleges görbére definiált az izooptikus görbe, melynek minden pontján az adott görbe érintőegyenesei adott szögben metszik egymást, azaz az izooptikus görbe minden pontjából az eredeti görbe az adott szög alatt látszik. Ennek speciális esete a szakasz fölé emelt látókörív. Háromdimenziós esetben az izooptikus görbe vagy felület olyan pontok

halmaza, melyekből mint kamerapozíciókból, a felületünk az adott szög alatt látszik. Ezt a látószöget meghatározhatjuk oly módon, hogy a felületünk bármely szemszögből nézve elférjen a képernyőn. Továbbá egy ilyen módon választott izooptikus görbével kamerapályát hozhatunk létre a felület körül, melyen végighaladva a felületet teljes terjedelmével és a lehető legközelebből szemlélhetjük az adott látószög figyelembevételével.

Egy ilyen térbeli izooptikus görbével látványos hatást érhetünk el a kameramozgatás során, de elméleti jelentősége is számottevő, hiszen ez lehet az első, síkbeli izooptikus problémák térbeli általánosítása felé megtett lépés. A célunk egy olyan algoritmus tervezése volt, mely megtalálja a felület egy izooptikus görbáját, mégpedig a lehető leghatékonyabb módon. A tervezések során igyekeztünk egzakt számolást, illetve minél pontosabb és gyorsabb eljárást fejleszteni. Önálló eredményeink a következők: kétdimenziós görbékre az [1]-ben leírt izooptikus kereső módszer volt a kiindulópont, mely Bézier-görbékre oldja meg a problémát. Ezen algoritmus hatékonyságának növelése érdekében tettünk módosításokat, majd ezt az algoritmust általánosítottuk más görbetípusra, elsősorban a geometriai modellezésben standard NURBS görbékre. Végül a síkbeli eredményeket térbeli esetekre is alkalmaztuk. A számítások bonyolultsága miatt három dimenzióban az általános probléma nem kezelhető, ezért egy speciális helyzetben egyetlen izooptikus görbét, illetve kamerapályát hoztunk létre.

Egy izooptikus pontot az adott görbe érintőegyeneseinek metszéspontjaként definiáltunk. Ebből következik, hogy a különböző típusú görbék deriváltjainak gyors és pontos meghatározása fontos szerepet tölt be a számolások során. Mivel a Bézier-görbékre ez gyorsabb eljárás, ezért először ezekre terveztünk algoritmusokat, majd általánosítottunk a további görbetípusokra.

Háromdimenziós térben viszont már az egyszerűbbnek gondolt Bézier-felületekre megalkotott módszer is meglehetősen bonyolult, ezért az általánosítás helyett inkább ezen egyszerűbb felülettípus izooptikus pontjainak a keresését igyekeztünk optimalizálni. Ezen túlmenően beszámolunk további elméleti lehetőségekről is, melyekre egyelőre nem sikerült a gyakorlatban alkalmazható algoritmusokat megalkotni, de fontosnak tartjuk az ezek alapján szerzett tapasztalatokat. És végül a működő elméleti módszer lényegét szeretnénk feltárni, és az ezekre megalkotott gyakorlati, kereső algoritmos megvalósításokat bemutatni.

# 1. Görbék izooptikusa

**1. Definíció.** *Adott síkgörbe izooptikus görbéjén azon pontok halmazát értjük, melyekből az eredeti görbe adott szög alatt látszik.*

Másodfokú görbékre különböző klasszikus geometriai eredményeken alapuló számolási módszerekkel az izooptikus görbe egyszerűen meghatározható, melyek magától értetődően a definícióból következnek. Szabadformájú görbék esetén az adott görbetípus deriváltjából származó érintőegyeneselek metszéspontjai adják az izooptikus pontokat. Ezzel az algoritmussal konvex görbékre találunk olyan összefüggő ponthalmazokat, melyek görbét alkotnak. Az eljárás egyértelmű, de a sebesség növelése érdekében az algoritmus optimalizálható, a keresési intervallum szűkíthető. E módszer lényegét szeretnénk a következőkben bemutatni.

## 1.1. Bézier-görbére felírt izooptikus kereső módszerrel

Az  $n$ -edrendű Bézier-görbe általános képlete a következő:

$$B(t) = \sum_{i=0}^n b_i^n(t) P_i.$$

Ahol  $P_i$  ( $i = 0, \dots, n$ ) kontrollpontok, a

$$b_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

pedig az  $n$ -edrendű Bernstein-polinom,  $t = 0, \dots, 1$ . A Bézier-görbék deriváltját a Bernstein-polinomok tulajdonságai alapján a következőképpen lehet meghatározni:

$$B(t) = n \sum_{i=0}^n (b_{i-1}^{n-1}(t) - b_i^{n-1}(t)) P_i, \quad (1.1)$$

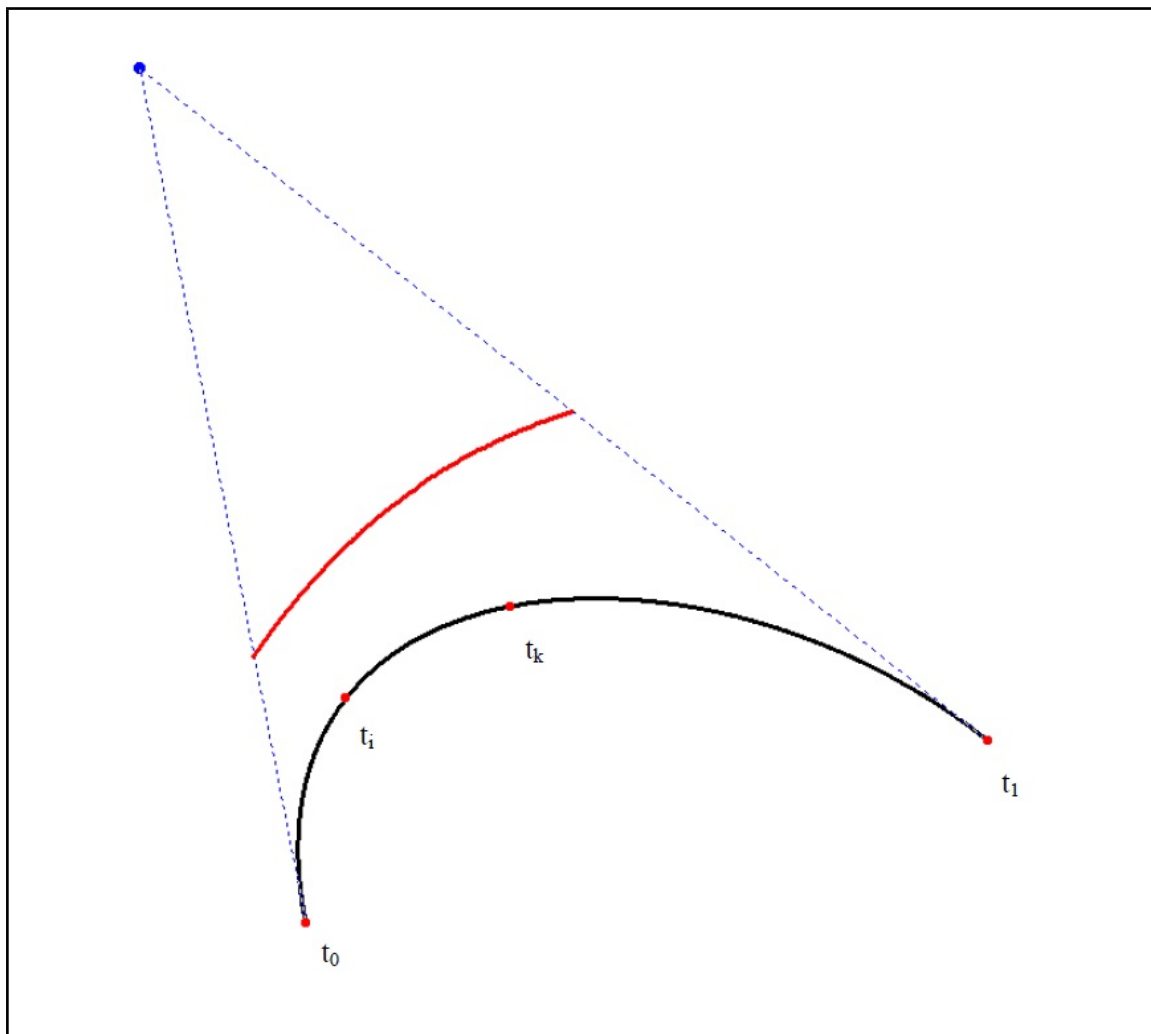
és ebből levezethető a

$$B(t) = n \sum_{i=0}^{n-1} b_i^{n-1}(t) (P_{i+1} - P_i). \quad (1.2)$$

A deriváltak mint irányvektorok töltenek be fontos szerepet az algoritmus során. Két vektor skaláris szorzatából meghatározható az általuk bezárt szög. Ha ez a keresett látószöggel egyenlő, akkor ezekkel és a görbe hozzájuk tartozó pontjával felírható az érintőegyeneselek egyenlete. Az egyenletrendszer megoldásaként kapott  $X$  és  $Y$  koordináta lesz egy pont az izooptikus görbén. A metszéspont természetesen homogén koordinátákkal is meghatározható. Mivel

a deriváltakból származó vektoroknak csak az iránya fontos, ezért a skalárt a képletből elhagyhatjuk. Ez az (1.1) és az (1.2) esetében az  $n$ -es szorzó, harmadrendű görbéknél természetesen 3.

Az eljárás során logaritmikus keresés használata ajánlott, mert az érintők bezárt szögeit tekinthetjük egy rendezett tömbnek. Ugyanis konvex görbék érintőegyenesei minél kisebb távolságú paraméterértékekhez tartoznak, annál nagyobb szöget zárnak be, minél nagyobb a távolság az indexek között, annál kisebbet. Ugyanezen tulajdonság miatt a keresés a következőképpen szűkíthető: A Bézier-görbe első pontjához ( $t_0$  paraméterértékhez) húzott érintőegyenes párját ( $t_k$  indexnél) keressük meg először, majd az utolsó görbeponthoz ( $t_1$  paraméterértékhez) húzott érintőegyenes párját ( $t_i$  indexnél). A  $t_0, t_i$  mutatószámok közötti görbepontok érintőinek párai biztosan a  $t_k, t_1$  paraméterértékű görbepontok között lesznek. Ezt szemlélteti a 2. ábra. Majd minden izooptikus pont megtalálása után ezek az intervallumok szűkülnek, függetlenül attól, hogy melyik irányból indítjuk a keresést.



2. ábra. Bézier-görbe  $90^\circ$ -os látószögű izooptikus görbéje keresési intervallumokkal

Szélsőséges helyzetben előfordulhat, hogy a keresés a görbe mindkét oldalán talál izooptikus pontokat. Ilyenkor érdemes koordinátavizsgálattal szűrni, hogy csak az azonos oldalon lévő pontok jelenjenek meg, amik így eredményül egy görbét alkotnak.

## 1.2. NURBS görbére felírt izooptikus kereső módszerrel

Az imént bemutatott algoritmus felírható bármely szabadformájú görbére, ha ismerjük annak differenciálhányados függvényét. A következőkben az előző módszer általánosításaként a NURBS görbe izooptikus görbét keressük. A  $p$ -adrendű NURBS görbe általános képlete és

deriváltja a [2] alapján:

$$C(u) = \frac{\sum_{i=0}^n N_i^p(u) w_i P_i}{\sum_{i=0}^n N_i^p(u) w_i}.$$

Ahol  $P_i$  ( $i = 0, \dots, n$ ) kontrollpontok, a  $w_i$  a hozzájuk tartozó súlyértékek ( $w_i > 0$ , bármely  $i$ -re). Az  $U = \{a, \dots, a, u_{p+1}, \dots, u_{m-p-1}, b, \dots, b\}$  a csomóérték vektor ( $m = n + p + 1$ ,  $a \leq u \leq b$ ). Meghatározza, hogy az egyes kontrollpontok hogyan hatnak a görbére. Az  $N_i^p(u)$  bázisfüggvény a következőképpen definiált:

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

$$N_i^p(u) = \frac{u - u_i}{u_{i+p} - u_i} N_i^{p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(u).$$

A differenciálhányados függvény pedig:

$$C'(u) = \frac{A'(u) - w'(u)C(u)}{w(u)}. \quad (2.1)$$

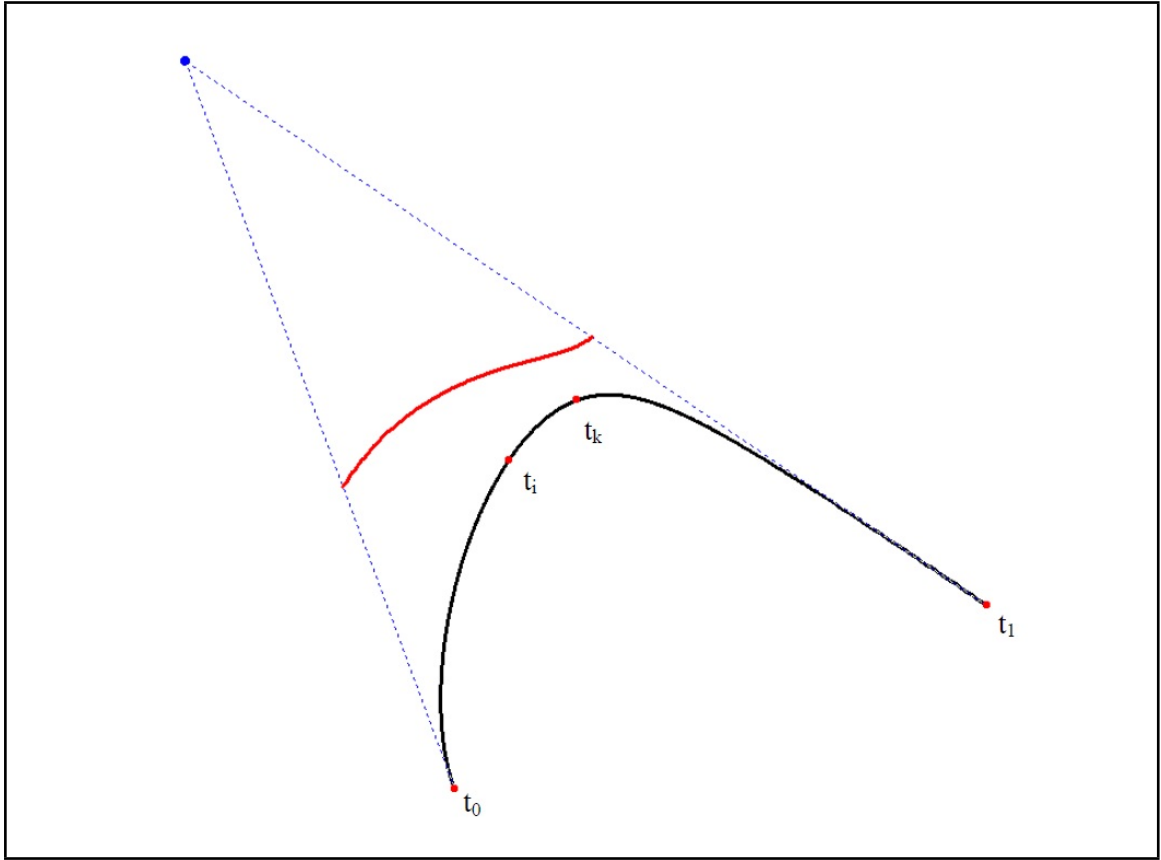
Ahol  $C(u)$  a NURBS görbe, és

$$w(u) = \sum_{i=0}^n N_i^p(u) w_i \quad (2.2)$$

$$w'(u) = \sum_{i=0}^{n-1} N_{i+1}^{p-1}(u) p \frac{w_{i+1} - w_i}{u_{i+p+1} - u_{i+1}} \quad (2.3)$$

$$A'(u) = \sum_{i=0}^{n-1} N_{i+1}^{p-1}(u) p \frac{P_{i+1} w_{i+1} - P_i w_i}{u_{i+p+1} - u_{i+1}}. \quad (2.4)$$

Az algoritmus a Bézier-görbénél megismert, eredménye a 3. ábrán látható. A deriváltaknak itt is csak az iránya fontos számunkra. Tehát a  $p$  konstans a (2.3) és a (2.4) képletből, a számolás gyorsítása érdekében elhagyható.



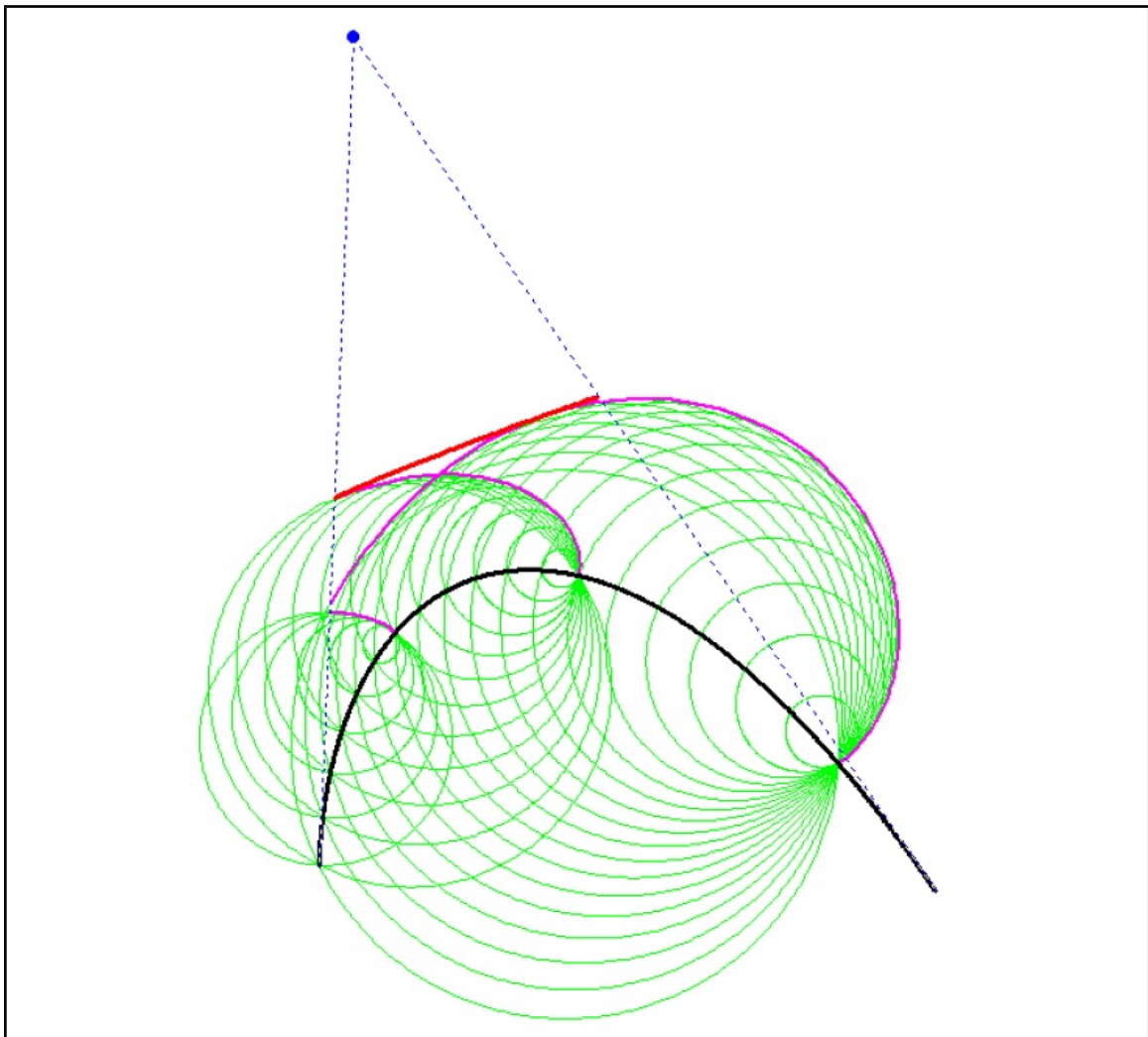
3. ábra. NURBS görbe  $90^\circ$ -os látószögű izooptikus görbéje keresési intervallumokkal

### 1.3. Izooptikus görbe egzakt számolása

Szabadformájú görbék izooptikusának meghatározására a bemutatott keresési algoritmus alkalmazható. A következőt mint az izooptikus görbe egzakt számolását mutatjuk be az [1] alapján.

Két pontot összekötő szakaszra felírt, adott szöghöz tartozó látóörívek gyakorlatilag a szakasz izooptikus görbéjét adják. Ha két pontot veszünk egy tetszőleges görbén, akkor az őket összekötő szakaszra felírt látóörív egy pontja izooptikusa lesz a görbének. A konvexitás miatt ezeket a szakaszokat irányított szakaszoknak tekintjük és azonos oldali látóörívekkel dolgozunk.

Legyen adott a  $B(t)$   $n$ -edrendű Bézier-görbe. Legyenek  $t$  és  $t_0$  a  $[0, 1]$  intervallumon, úgy hogy  $t < t_0$ . Írjunk fel látóöríveket a görbe  $B(t_0)$  és  $B(t)$  pontjait összekötő szakaszokra. Ezeknek a körsoroknak a burkolóinak a burkolója lesz az izooptikus görbe (4. ábra). A burkológörbe



4. ábra. Bézier-görbe 90°-os látószögű körsorainak burkolója

egy olyan görbe, amely egy görbesereg minden elemét érinti egy pontban.

A Bézier-görbe körsorainak az  $X$  és  $Y$  koordináta-függvényeit  $f_x(t, u)$  és  $f_y(t, u)$  kétparaméteres függvények adják, ahol  $t$  a  $[0, 1]$ ,  $u$  a  $[0, 2\pi]$  intervallum elemei. Ezekre a körsorokra felírt burkoló koordináta-függvényei:  $f_x(u(t), t)$  és  $f_y(u(t), t)$ . Az  $u(t)$ -t a következőképpen felírt parciális deriváltakból álló mátrix determinánsából számoljuk, úgy hogy kifejezzük belőle az  $u$ -t:

$$\begin{vmatrix} \frac{\partial f_x(u, t)}{\partial u} & \frac{\partial f_x(u, t)}{\partial t} \\ \frac{\partial f_y(u, t)}{\partial u} & \frac{\partial f_y(u, t)}{\partial t} \end{vmatrix} = 0.$$

Az első burkoló a formula alapján kiszámolható. Viszont a burkolók burkolóját a számítások bonyolultsága miatt még komputeralgebrai program segítségével sem sikerült eddig meghatározni. Ezt azért szeretnénk kihangsúlyozni, mert még két dimenzióban is nehézkes az egzakt számolás. Három dimenzióban a képletek bonyolultsága pedig tovább nő.

## 2. Felületek izooptikusa

Az első dolgunk az volt, hogy meghatározzuk, hogy a felület milyen típusú izooptikusát keressük. Egy pontot biztosan találunk, melyről megállapíthatjuk, hogy milyen szög figyelembevétele mellett vizsgáljuk a felületünket. Egy görbét is, mely kamerapálya lehet a modellünk körül. De akár egy burkoló felület is létrejöhet izooptikus pontokból, mely térben is ténylegesen körülhatárolja a felületünket. Síkban az izooptikus pont a megfelelő érintőegyenese metszéspontja. Térben ilyen metszéspontok keresése nagyon nehéz feladat. Ezért megpróbáltuk valamiképp egyszerűsíteni a helyzetünket, és speciális esetekben kerestünk, majd igyekeztünk általánosítani.

### 2.1. Felület elmetszése síkokkal

Az első elképzelésünk az volt, hogy a felületet elmetszük egy síkkal. A feladat így egy felületre illeszkedő síkgörbére specializálódik. Erre a megismert módon meghatározhatunk egy izooptikus görbét, mely természetesen a felületünk izooptikusa is lesz. A síkmetszetet a [3] segítségével igyekeztünk meghatározni.

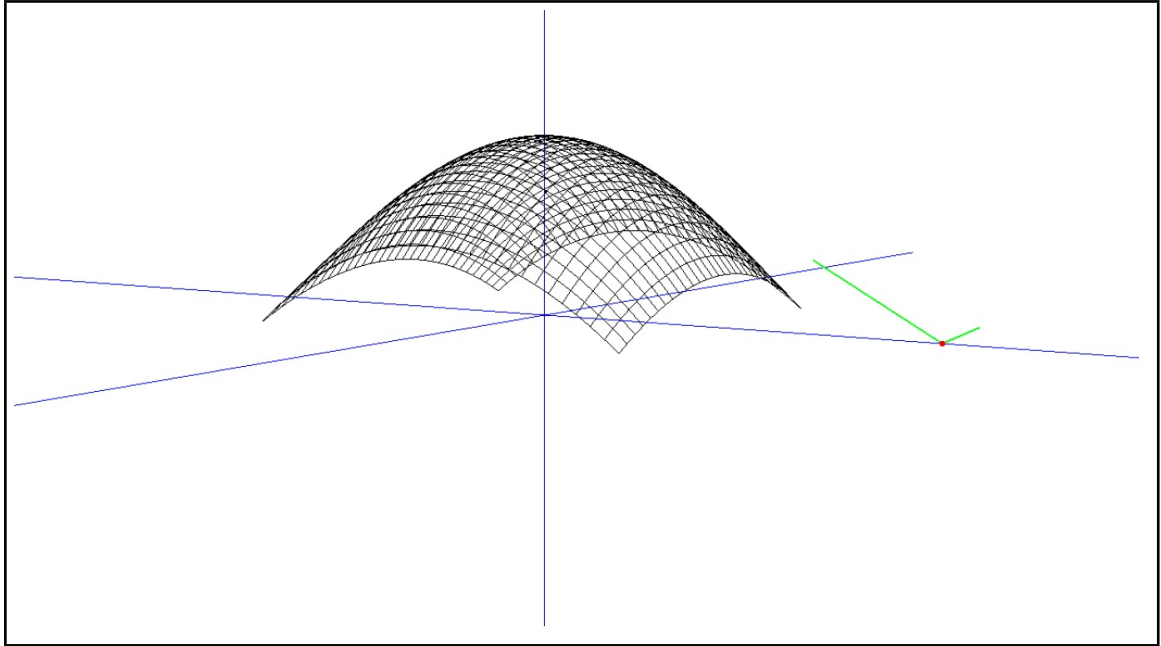
A cikk harmadrendű Bézier-felület ( $S$ ) és egy adott  $P$  sík metszésének algoritmusát mutatja be. A eljárás eredményeként olyan  $u$  paramétereket kapunk, melyeket az  $S$  képletébe beírva a felület azon pontjait határozhatjuk meg, melyek a  $P$  síkon találhatók. A módszer legfőbb hiányossága, hogy a pontok darabszámát nem tudja előre meghatározni. Csak azt tudjuk, hogy találunk-e a metszetben felületpontot vagy sem. Ezáltal a kapott ponthalmazra valamilyen interpolációs eljárást kellett volna alkalmazni, hogy ténylegesen görbét határozzunk meg a térben. Majd erre kellett volna a keresést alkalmazni.

Az algoritmusok sebessége, az interpoláció és a keresés pontatlansága nem adott elfogadható eredményt. A módszer nem volt megfelelő. Így más oldalról igyekeztünk megközelíteni a problémát.

## 2.2. Normálvektoros keresés kontrollpont forgatással

Egy speciális helyzetet képzelünk el. A felületünket pontosan az  $y = 0$  egyenletű sík „fölött” helyezzük el. Abból kiindulva, hogy egy monitor szélessége legalább akkora, mint a magassága, a látószöget függőleges irányban mérjük fel a felületünkre, és úgy vizsgáljuk a modellt az  $y = 0$  egyenletű síkról, hogy a kamera az origóba néz.

A kamerát egy  $X$  tengellyel párhuzamos  $\mathbf{m} = (1, 0, 0)$  vektorral és a  $Z$ -tengellyel párhuzamos  $(0, 0, 1)$  vektor  $\alpha$  szögű elforgatottjával, egy  $\mathbf{v}$  vektorral írjuk le. Az  $\alpha$  a függőleges irányban megadott látószög. Így  $\mathbf{v} = (0, \sin(\alpha), -\cos(\alpha))$  lesz. Ezt a kamerapontot úgy kell meghatározni, hogy a  $\mathbf{v}$  irányvektorú, a ponton átmenő egyenes egy érintője legyen a felületnek (5. ábra).



5. ábra. Kamera megadása

Tehát az eljárásban a felületnek azokat a normálvektorait keressük, amelyek derékszöget zárnak be  $\mathbf{m}$  és  $\mathbf{v}$  vektorokkal.

Az algoritmust Bézier-felületre terveztük meg, melynek képlete:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^n B_i^n(u) B_j^n(v) P_{i,j}.$$

A  $B_i^n(u)$  és  $B_j^n(v)$  Bernstein-polinomok, a kontrollpontok  $P_{i,j}$ ,  $i = 0, \dots, n$ ,  $j = 0, \dots, n$ . A Bézier-felület normálvektorát az  $U(u, v) = \frac{\partial S(u, v)}{\partial u}$  és  $V(u, v) = \frac{\partial S(u, v)}{\partial v}$  szerinti parciális deriváltak vektoriális szorzataként kapjuk. A differenciálhányados függvényeket a Bernstein-polinom tulajdonságai alapján itt is kétféleképpen határozhatjuk meg:

$$U(u, v) = n \sum_{i=0}^n \sum_{j=0}^n (B_{i-1}^{n-1}(u) - B_i^{n-1}(u)) B_j^n(v) P_{i,j}$$

és ebből levezethető a

$$U(u, v) = n \sum_{i=0}^{n-1} \sum_{j=0}^n B_i^{n-1}(u) B_j^n(v) (P_{i+1,j} - P_{i,j}).$$

Ez az  $u$  szerinti parciális derivált függvény. A  $V(u, v)$   $v$  szerinti differenciálhányados függvény hasonlóképpen. Tehát a felület  $u$  és  $v$  paraméterértékhez tartozó pontjához tartozó érintősík normálvektora  $N(u, v) = U(u, v) \times V(u, v)$  alakban számolandó.

Első esetnek tekintsük azt, amikor a felület kontrollpontjait forgatjuk az  $Y$  tengely körül. A normálvektorral és a Bézier-felület azon pontjával, amelyre a normálvektort megtaláltuk, felírhatjuk az érintősík egyenletét. Ahol ez az érintősík és az  $y = 0$  egyenletű sík el metszik egymást, azon az egyenesen lesz az izooptikus pont.

$$N_x(u, v)(X - X_0) + N_z(u, v)(Z - Z_0) = N_y(u, v)Y_0$$

Az  $X_0, Y_0, Z_0$  a Bézier-felület adott pontjának a koordinátái. A kapott egyenes az  $X$  tengellyel párhuzamos, hiszen a kameránk az  $Y$  tengely körül nem forog, tehát az egyenletet  $X = 0$ -val megoldva kapjuk az izooptikus pont origótól való távolságát.

$$Z = \frac{N_y(u, v)Y_0 - N_x(u, v)X_0}{N_z(u, v)} + Z_0$$

A pontot annyival kell elforgatni az  $Y$  tengely körül, amennyivel a kontrollpontokat is elforgattuk. A forgatás során a következőknek kell teljesülnie:  $\langle N(u, v), \mathbf{m} \rangle = 0$  és  $\langle N(u, v), \mathbf{v} \rangle = 0$ . Ez felbontva a következő:

$$\begin{aligned} N_x(u, v) &= 0 \\ \sin(\alpha) \cdot N_y(u, v) &= \cos(\alpha) \cdot N_z(u, v). \end{aligned}$$

Ha ezt az egyenletrendszert megoldjuk  $u$ -ra és  $v$ -re, akkor a megfelelő normálvektorok egzakt számolását kapjuk, melyből az izooptikus pont pontosan meghatározható. Sajnos az egyenletrendszer megoldásait  $n = 2$  esetben sem tudtuk meghatározni, még komputeralgebrai programok segítségével sem. Az imént ismertetett módszer hátránya a pontatlanság és a lassúság. Az izooptikus pontokat csak a sebesség jelentős romlásával tudjuk pontosabban megkeresni.

### 2.3. Normálvektoros keresés $\mathbf{m}$ és $\mathbf{v}$ vektorok forgatásával

Másik lehetőség az izooptikus pontok megkeresésére, ha az  $\mathbf{m}$  és a  $\mathbf{v}$  vektorokat forgatjuk el az  $Y$  tengely körül. A vektorok koordinátái a következőképpen alakulnak:  $\mathbf{m} = (\cos(\beta), 0, -\sin(\beta))$  és  $\mathbf{v} = (-\sin(\beta) \cdot \cos(\alpha), \sin(\alpha), -\cos(\beta) \cdot \cos(\alpha))$ , ahol  $\beta$  az  $Y$  tengely körüli forgatás szöge,  $[0, 2\pi]$  intervallumban. A normálvektor keresésének a feltétele továbbra is, hogy  $\langle N(u, v), \mathbf{m} \rangle = 0$  és  $\langle N(u, v), \mathbf{v} \rangle = 0$ . Egyszerűsítések után ez a következő egyenletrendszert adja:

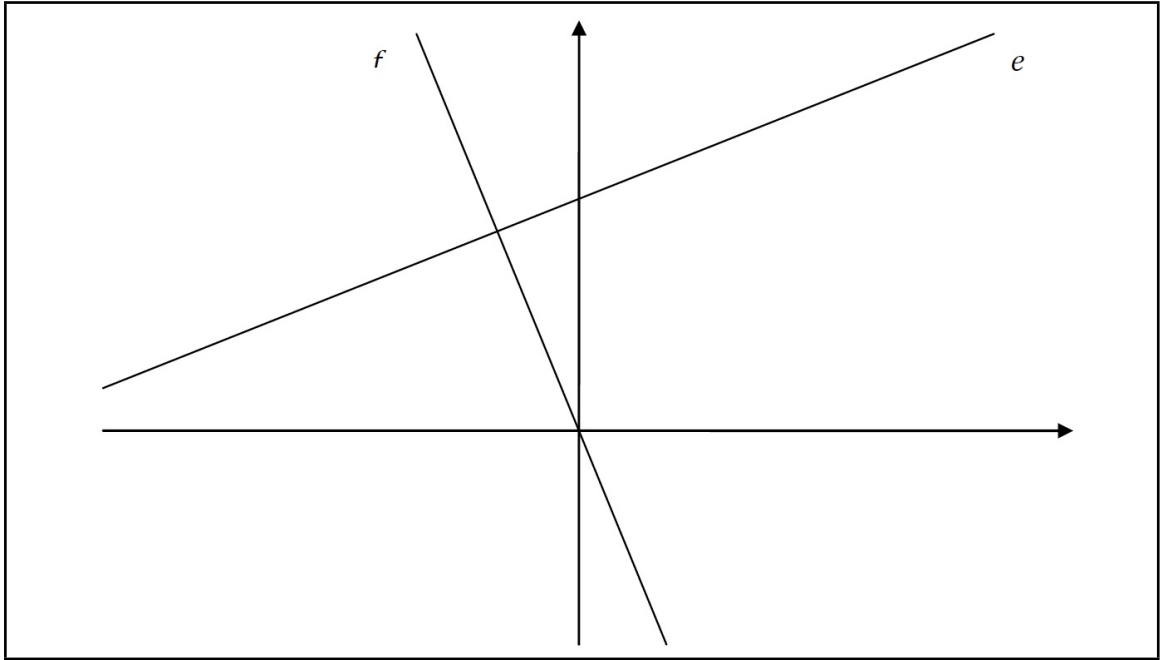
$$\begin{aligned} N_x(u, v) &= \tan \beta \cdot N_z(u, v) \\ N_y(u, v) &= \cot \alpha \cdot \frac{1}{\cos \beta} \cdot N_z(u, v). \end{aligned}$$

A megoldását, mivel nem lett egyszerűbb,  $u$ -ra és  $v$ -re így sem tudtuk meghatározni. Viszont a további számolások egyszerűbbek lettek azáltal, hogy nem kell forgatnunk a kontrollpontokat.

Az izooptikus pont  $X$  és  $Y$  koordinátáit szeretnénk meghatározni. Ehhez tekintsük az előző algoritmusból megismert érintősíket és az  $y = 0$  egyenletű sík metszésvonalát ( $e$ ), illetve azt az egyenest, amelynek egy pontja az origó, és normálvektora az  $m$  vektor ( $f$ ). A két egyenes metszéspontja lesz az izooptikus pont (6. ábra).

$$\begin{aligned} X &= \frac{\sin(\beta) \cdot c}{\cos(\beta) \cdot N_z(u, v) - \sin(\beta) \cdot N_x(u, v)} \\ Z &= \frac{\cos(\beta) \cdot c}{\cos(\beta) \cdot N_z(u, v) - \sin(\beta) \cdot N_x(u, v)} \end{aligned}$$

Ahol  $c = N_x(u, v)X_0 + N_y(u, v)Y_0 + N_z(u, v)Z_0$ .



6. ábra. Az  $e$  és  $f$  egyenes metszéspontja

Mivel  $\langle N(u, v), \mathbf{m} \rangle = 0$ , a koordináták felírhatóak a következőképpen is:

$$X = \frac{cN_x(u, v)}{\langle N_x(u, v), N_x(u, v) \rangle + \langle N_z(u, v), N_z(u, v) \rangle}$$

$$Z = \frac{cN_z(u, v)}{\langle N_x(u, v), N_x(u, v) \rangle + \langle N_z(u, v), N_z(u, v) \rangle}$$

## 2.4. Izooptikus keresés lineáris kombinációval

Az imént tárgyalt speciális helyzet egy másik számolási módszerét szeretnénk bemutatni. A kamerapontjainkat továbbra is  $\mathbf{m}$  és  $\mathbf{v}$  vektorok segítségével keressük meg. Mivel azt mondtuk, hogy a felületünk normálvektorát az  $U(u, v)$  és  $V(u, v)$  vektorok vektoriális szorzataként adtuk meg, és az  $N(u, v)$  vektorunk  $90^\circ$ -ot zár be  $\mathbf{m}$ -mel és  $\mathbf{v}$ -vel, akkor a  $\mathbf{v}$ ,  $\mathbf{m}$ ,  $U(u, v)$  illetve a  $\mathbf{v}$ ,  $\mathbf{m}$ ,  $V(u, v)$  vektorok lineárisan függők.  $U(u, v)$  és  $V(u, v)$  lineárisan kombinálható  $\mathbf{m}$  és  $\mathbf{v}$  vektorokból. Ez a következőt jelenti:

$$\begin{vmatrix} 1 & 0 & 0 \\ 1 & \sin(\alpha) & -\cos(\alpha) \\ U_x(u, v) & U_y(u, v) & U_z(u, v) \end{vmatrix} = 0,$$

$$\begin{vmatrix} 1 & 0 & 0 \\ 1 & \sin(\alpha) & -\cos(\alpha) \\ V_x(u, v) & V_y(u, v) & V_z(u, v) \end{vmatrix} = 0.$$

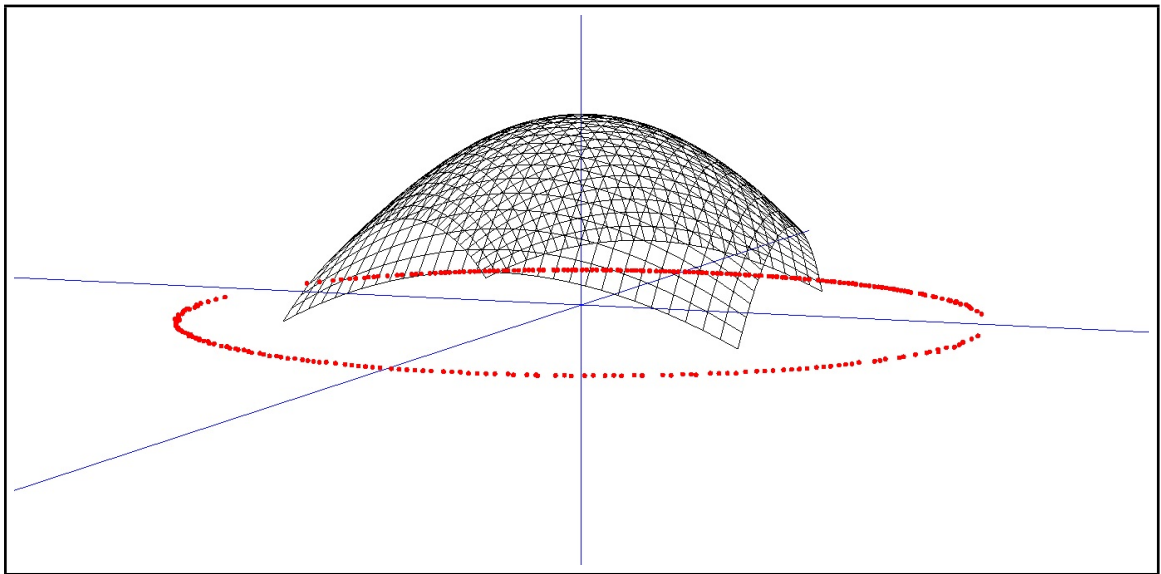
Ezek alapján a következő egyenletrendszert kapjuk:

$$\sin(\alpha) \cdot U_z(u, v) + \cos(\alpha) \cdot U_y(u, v) = 0$$

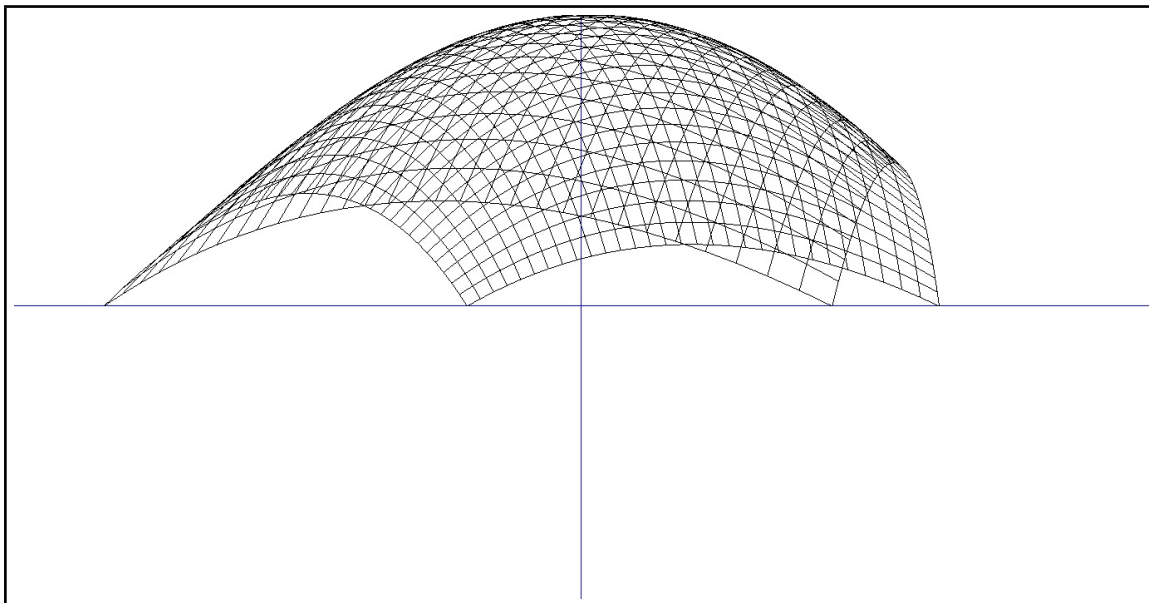
$$\sin(\alpha) \cdot V_z(u, v) + \cos(\alpha) \cdot V_y(u, v) = 0.$$

Kézi számításokkal nem sikerült kifejezni, viszont komputeralgebrai programmal megoldva az egyenletrendszert,  $u$ -ra és  $v$ -re egy ötödfokú egyenletet kaptunk, mely gyökeinek közelítése, bonyolultságából adódóan nem hozott eredményt.

A kereső módszerek eredményei láthatók a 7. és 8. ábrán.



7. ábra. A megtalált 30°-os látószögű izooptikus pontok



8. ábra. A felület szemléltetése egy  $30^\circ$ -os látószögű izooptikus pontból

## Összefoglalás

Síkbeli görbék izooptikusát a keresési algoritmusok hatékonyan megtalálják. Az eljárás általános, bármely görbetípusra felírható.

A térbeli izooptikus kereső módszerek még gyerekcipőben járnak. Nagyon fontos előrelépés viszont, hogy a bemutatott speciális helyzetre egy térbeli izooptikus görbe megtalálható. Megemlítendő, hogy egyéb speciális helyzeteket is meg lehet határozni, melyekre más keresési algoritmus is felírható, erre tett lépést is bemutattunk, de ez sajnos nem vezetett eredményre. Ugyanis az eddig tárgyalt problémamegoldás alapján, miszerint az egyszerűbb helyzetek keresése után általánosítunk, más kiinduló helyzetek is elképzelhetők.

A megtervezett és bemutatott térbeli kereső módszerek nem tökéletesek. Mint említettük, a sebesség és pontosság terén vannak hiányosságok. További módosítások lehetnek szükségesek a nagyobb gyorsaság elérésére, illetve az egyenletes kamerapálya kialakítása érdekében. De nagyon fontos az említett elméleti jelentősége, miszerint ez az első kezdeményezés térbeli izooptikus problémák megoldására, tehát a bemutatott eljárások kiinduló pontok lehetnek további módszerek számára. Az egzakt számolások terén is szükséges a folytatás. A bonyolult számolások nagyobb teljesítményt igényelnek, amihez szuperszámítógép használatát tervezzük.

## **Köszönetnyilvánítás**

Ezúton szeretném megköszönni Dr. Hoffmann Miklósnak és Kunkli Rolandnak a sok segítséget és hasznos tanácsot, melyeket a TDK pályamunkám készítése során nyújtottak.

## Irodalomjegyzék

- [1] Kunkli, R., Papp, I., Hoffmann, M. (2013), Isoptics of Bézier curves, *Computer Aided Geometric Design*, 30, 78–84.
- [2] Les A. Piegl, Wayne Tiller (2012), The Nurbs book 2nd Edition, 117–128.
- [3] Qingxiang Fu (1990), The intersection of a bicubic Bézier patch and a plane, *Computer Aided Geometric Design*, 7, 475–488.

## Saját munka leírása

- A síkbeli izooptikus görbe kereső algoritmus implementációjának általánosítása.
- A térbeli síkmetsző módszer ötlete.
- A normálvektor fogatásából számított mindkét keresési eljárás ötlete és algoritmus.
- A lineáris kombináció felismeréséből adódó számítás és implementáció.