



# Izooptikus görbék és felületek a számítógépes geometriai modellezésben

Egyetemi doktori (PhD) értekezés

A szerző neve: Nagy Ferenc

A témavezető neve: Dr. Hoffmann Miklós

DEBRECENI EGYETEM  
Természettudományi és Informatikai Doktori Tanács  
Informatikai Tudományok Doktori Iskola  
Debrecen, 2022.

*Ezen értekezést a Debreceni Egyetem Természettudományi és Informatikai Doktori Tanács Informatikai Tudományok Doktori Iskola Diszkrét matematika, adatfeldolgozás és vizualizáció programja keretében készítettem a Debreceni Egyetem műszaki tudományokban doktori (PhD) fokozatának elnyerése céljából. Nyilatkozom arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.*

Debrecen, 2022. ....

.....  
Nagy Ferenc  
(jelölt)

*Tanúsítom, hogy Nagy Ferenc doktorjelölt 2016–2020 között a fent megnevezett Doktori Iskola Diszkrét matematika, adatfeldolgozás és vizualizáció programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Nyilatkozom továbbá arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét. Az értekezés elfogadását javasolom.*

Debrecen, 2022. ....

.....  
Dr. Hoffmann Miklós  
(témavezető)

# Izooptikus görbék és felületek a számítógépes geometriai modellezésben

Értekezés a doktori (PhD) fokozat megszerzése érdekében  
az informatika tudományágban

Írta: Nagy Ferenc okleveles programtervező informatikus

Készült a Debreceni Egyetem  
Informatikai Tudományok doktori iskolája  
(*Diszkrét matematika, adatfeldolgozás és vizualizáció* programja)  
keretében

Témavezető: Dr. Hoffmann Miklós

Az értekezés bírálói:

Dr. ....  
Dr. ....  
Dr. ....

A bírálóbizottság:

elnök: Dr. ....  
tagok: Dr. ....  
Dr. ....  
Dr. ....  
Dr. ....

Az értekezés védésének időpontja: 20... .. .



# Tartalomjegyzék

Bevezetés	1
<b>1. Izooptikus görbék</b>	<b>6</b>
1.1. Előzmények	7
1.1.1. Síkgörbék izooptikus görbéi	7
1.2. Új eredményeink	8
1.2.1. Izooptikus görbék térbeli általánosítása	8
1.2.2. A felület kontrollpontjainak forgatása	10
1.2.3. Az $\vec{m}$ és $\vec{v}$ vektorok forgatása	11
1.2.4. Számolás lineáris kombinációval	13
<b>2. Izooptikus felületek</b>	<b>15</b>
2.1. Előzmények	16
2.1.1. A látószög térbeli általánosításával felírt definíció	16
2.1.2. Analitikus számolási módszer	17
2.2. Új eredményeink	18
2.2.1. Izooptikus felületek keresése	18
2.2.1.1. Konvex modellek esetén	18
2.2.1.2. Konkáv modellek esetén	21
2.2.1.3. Az algoritmusok sebessége	31
2.2.2. Alternatív izooptikus felület definiálása	31
2.2.3. Izooptikus felületek keresésének párhuzamosítása	34
2.2.3.1. Térszögszámítás párhuzamosítása	35
2.2.3.2. Keresési módszerek párhuzamosítása	41
2.2.3.3. A szekvenciális és párhuzamos algoritmusok sebességének összehasonlítása	43

<b>3. Log-aesthetic görbék</b>	<b>47</b>
3.1. Előzmények . . . . .	49
3.1.1. Log-aesthetic görbe származtatása . . . . .	49
3.1.2. Log-aesthetic görbe egyenletei . . . . .	52
3.1.3. Interaktív modellező algoritmus . . . . .	56
3.2. Új eredményeink . . . . .	59
3.2.1. Log-aesthetic görbe és izooptikusának kapcsolata	59
3.2.2. Továbbfejlesztett modellező algoritmus . . . . .	69
3.2.2.1. Kiterjesztett log-aesthetic görbék . . . . .	70
3.2.2.2. Kibővített interpolációs eljárás . . . . .	77
3.2.2.3. Kezdőpont görbületének szabályozása .	81
<b>Összefoglalás</b>	<b>90</b>
<b>Summary</b>	<b>92</b>
<b>Köszönetnyilvánítás</b>	<b>94</b>
<b>Irodalomjegyzék</b>	<b>95</b>
<b>Publikációs lista</b>	<b>104</b>
<b>Függelék</b>	<b>113</b>
A. Kódrészletek . . . . .	113
B. Különböző részletességű modellek izooptikus felületei .	118

# Bevezetés

A számítógéppel segített geometriai tervezés (Computer-Aided Geometric Design - CAGD) elsődleges feladata különböző objektumok (számítógépes modellek) létrehozása, megtervezése. Alapvető eszköztárát a görbék és felületek adják, melyek pontos matematikai leírásán túlmenően a hozzájuk kapcsolódó geometriai problémák megoldása is a témakör része. Ezen problémák főként ipari szükségleteket kielégítő alkalmazásokhoz köthetők, bár ezen túl önmagában is érdekes matematikai kutatási területről beszélhetünk.

A síkbeli tervezés központi elemei a modellező görbék, például a különböző CAD/CG (Computer Aided Design/Computer Graphics) szoftverekben standard Bézier-, B-spline vagy NURBS görbék. A síkbeli objektumok tervezésén felül, bármely síkgörbéből különböző egyéb görbék is le lehet vezetni: evolvensok családját (lefejtési görbék), evolutát (normálsereg burkolója), párhuzamos (offset) görbék, illetve az izooptikus görbék, amelyekkel kutatásunk során foglalkoztunk. Ezen görbék szinte mindegyike gyakorlati alkalmazást nyújt a geometriai modellezés vagy a fizika valamely területén.

Az izooptikus görbe minden pontjában az adott görbe érintőegyenesei (vagy ha több ilyen van, akkor a két szélső érintőegyenese) egy megadott szögben metszik egymást. Ez a szög a látószög. Az izooptikus görbe elnevezést Taylor javasolta 1884-ben [56], habár ennél korábbi eredményekre utaló hivatkozásokat találhatunk például Yates munkájában [62], mely néhány klasszikus görbe izooptikus görbését és azok tulajdonságait mutatja be. sőt, egyes klasszikus eseteket (például a parabola izooptikusa hiperbola) már az ókori görög matematikusok is ismertek. További izooptikus síkgörbék számolásával kapcsolatos ismert eredményeket az 1.1.1. fejezetben mutatunk be.

Izooptikus görbék gyakorlati alkalmazására a mechanika területén találunk példát [59, 60], de egy térbeli izooptikus definiálása és haszná-

lata kiemelkedő jelentőségű lehet olyan ipari vagy kutatási területeken, melyek a láthatóságot érintik. Például a különböző modellező szoftverekben egy adott objektum alakjától függő feltételeknek eleget tevő nézőpontok, illetve kamerapozíciók meghatározására jelenleg csak nagyon egyszerű esetekben és csak nagyon kezdetleges eszközök állnak rendelkezésre. Kutatásunk fő témakörét is ez motiválta.

Gyakran előforduló probléma ugyanis, hogy egy vizsgált háromdimenziós modell valamelyik tengely körül elforgatva vagy más szemszögből nézve „kilóg a képernyőről”, egyes részei nincsenek megjelenítve. Ezért egy megfelelő háromdimenziós izooptikus görbe vagy felület segítségével olyan pontok definiálhatók, melyekből, mint kamerapozíciókból, a modell 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 legközelebről szemlélhetjük az adott látószög figyelembevételével.

Az ehhez kapcsolódó algoritmust egy speciális helyzetben lévő Bézier-felületre terveztük meg. Az eljárás megkeresi azon izooptikus pontokat, amelyek a kamerának a felülethez a lehető legközelebbi helyzetét adják meg az adott szög figyelembevételével mellett. Ezek a pontok egy zárt izooptikus görbét képeznek a felület körül. Az algoritmust és a hozzá tartozó számolásokat az 1.2. fejezetben részletezzük. Egy ilyen térbeli izooptikus görbével látványos hatást érhetünk el a kameramozgatás során, de elméleti jelentősége is számottevő, hiszen ez az első, síkbeli izooptikus problémák térbeli általánosítása felé megtett lépés. Az erre vonatkozó eredményeket [45]-ben publikáltuk.

Izooptikusok háromdimenziós általánosításának egy másik lehetőségét írja le Csima és Szirmai [9] a látószög térbeli értelmezésének segítségével. Ha a látószöget a láthatóság mértékének tekintjük, és síkban a nézőpont körüli egységkör ívhosszával is mérhetjük, akkor egy kézenfekvő térbeli általánosítás lehet a láthatóság mérésére a nézőpont

körüli egységgömbre vetített terület. Ezek alapján, a szerzők az említett műben egy zárt, háromdimenziós tartomány izooptikus felületét definiálják, továbbá egy analitikus számolási módszert is leírnak poliéderek izooptikus felületének meghatározására. A pontos definíciót, illetve az izooptikus felület számolásának módszerét a 2.1. fejezetben ismertetjük.

Sajnos a módszer csak konvex alakzatok esetén működik és az izooptikus felület kiszámítása akár 10-20 percet is igénybe vehet, már egyszerű, szabályos poliéderek esetén is. Ezért a kutatásunk célja az volt, hogy egy olyan új, bonyolultabb alakzatok esetén is hatékony izooptikus felületet kereső eljárást írjunk le, amely konkáv alakzatokra is működik. Ezen új eredményeinket, illetve a keresőalgoritmust a 2.2.1. fejezetben mutatjuk be.

Ha az izooptikus görbét, mint az adott görbe érintővonalainak metszéspontjait tekintjük, akkor egy pontjából, mint a pozícióból vett legszélesebb nézetet is értelmezhetjük. Ezen elgondolás alapján a látószög térbeli általánosítása lehet a nézőpont körüli egységgömbre való vetítésének a legszélesebb átmérője. Az új megközelítéssel egy alternatív izooptikus felületet definiáltunk, melyet a 2.2.2. fejezetben mutatunk be. A kutatás fontossága és az algoritmusok újszerűsége végett az izooptikus felületekkel kapcsolatos eredményeinket szintén publikáltuk, [46]-ban.

Habár az általunk fejlesztett algoritmus képes konkáv alakzatok izooptikus felületét is meghatározni, bonyolultabb, akár többszáz poligonból álló modell esetén a folyamat még mindig több percet vehet igénybe. Ezért a következő célunk az volt, hogy felgyorsítsuk az általunk fejlesztett eljárást [46], hogy ésszerű időn belül megtaláljuk az izooptikus felületet összetettebb objektumok esetén is. A gyorsítást a szekvenciális számolási folyamatok párhuzamosításával, azok grafikus processzorokon történő implementálásával értük el (GPGPU – general-purpose computing on graphics processing units). Az új algoritmus lépéseit a 2.2.3. fejezetben ismertetjük. Mivel egyes esetekben

a párhuzamos algoritmussal történő izooptikus felület meghatározásának sebessége akár 100-szorosa is lehet az előző módszernek, az eredményeket tudományos folyóiratban is publikáltuk [43].

A főleg térbeli izooptikusokkal kapcsolatos kutatások mellett egy, a számítógéppel segített tervezésben fontos görbe, az úgynevezett log-aesthetic (logaritmikus görbületi sugarú, esztétikus) görbével és annak izooptikus görbéjének kapcsolatával is foglalkoztunk. Amellett, hogy a vizsgált görbét a magas ipari követelményeknek megfelelő (esztétikai) tervezésben a legígéretesebbnek definiálták [31], az izooptikus görbe számolása szempontjából is érdekes tulajdonságokkal rendelkezik.

A log-aesthetic görbe  $\alpha$  alakparaméterének módosításával több, különböző típusú görbe állítható elő. Többek között a logaritmikus spirál, az Euler-féle spirál, a Nielsen spirál, a kör, illetve a kör evolvens is. A görbe paraméterezhető húr hossz, illetve az irányszöge (az adott pontban vett érintőegyenese és az  $X$  tengely által bezárt szöge) szerint. A log-aesthetic görbe származtatását és paraméteres egyenleteit a 3.1.1. fejezetben részletezzük.

A log-aesthetic görbe úgynevezett autoevoluta, azaz egy  $\alpha$  alakparaméterű log-aesthetic görbe evolútája a  $-1/(\alpha - 2)$  alakparaméterű log-aesthetic görbe [65]. Ezen tulajdonság ösztönzött arra, hogy a görbe úgynevezett autoizooptikus tulajdonságát is megvizsgáljuk. Az autoizooptikus tulajdonság azt jelenti, hogy egy görbe izooptikus görbéje ugyanolyan típusú görbe. Mivel a kör izooptikus görbéje is kör, illetve a logaritmikus spirál izooptikus görbéje is logaritmikus spirál [50, 51], ezért azt vizsgáltuk, hogy a log-aesthetic görbe különböző alakparamétere által felírható különböző típusú görbék izooptikus görbéje log-aesthetic görbe-e. A kutatás során meghatároztuk az irányszög szerint paraméterezett görbék izooptikusának explicit egyenletét, illetve azt találtuk, hogy a log-aesthetic görbék csak  $\alpha = 1$  és az  $\alpha = \pm\infty$  alakparaméter esetén autoizooptikus görbék. Az elért eredményeket, melyeket tudományos folyóiratba közlésre benyújtottunk [44] a 3.2.1. fejezetben összegezzük.

A log-aesthetic görbével történő modellezés egyik alapvető módszere a Yoshida és Saito által felírt interaktív algoritmus [64], mely bizonyos, esetenként szigorú feltételek mellett megadott geometriai adatokra valós időben interpolál egy log-aesthetic görbét. Az eljárás lépéseit a 3.1.3. fejezetben ismertetjük. Kutatásunkban továbbfejlesztettük az algoritmust, enyhítve a geometriai adatok megadásának feltételeit, illetve egy új eljárással kiegészítettük, mely a  $G^1$  Hermite-interpoláció mellett log-aesthetic görbék  $G^2$  folytonos csatlakoztatását is lehetővé teszi. Az új módszert a 3.2.2. fejezetben ismertetjük. Ezen új modellező algoritmus tudományos folyóiratba történő közlését elfogaták, megjelenése a dolgozat írásakor folyamatban van [47].

# 1. Izooptikus görbék

Az izooptikus görbe elnevezés a 19. századból származik [56], azonban síkgörbék izooptikusait már az ókori görög matematikusok is tanulmányozták. Különböző klasszikus görbék izooptikusainak meghatározására számottevő hivatkozást találunk a szakirodalomban, ezért az 1.1.1. alfejezetben a pontos definíció megadásán kívül, néhány konkrét példát is bemutatunk.

A fejezet második részében a síkbeli izooptikus problémák térbeli általánosítása felé megtett első lépésünket ismertetjük. Az elméleti jelentőség mellett a gyakorlati felhasználás lehetőségét is szem előtt tartottuk, ugyanis a számítógéppel segített geometriai modellezés egyik fontos témaköre az úgynevezett legjobb nézet (best view) probléma, mely olyan pozíciók automatikus meghatározását jelenti, melyek a lehető leginkább informatív és intuitív képet adnak egy vizsgált háromdimenziós objektumról.

Természetesen a legjobb nézet(ek) megítélése, amellet, hogy nem feltétlenül objektív, erősen függhet az adott alakzat szemantikájától, továbbá a meghatározandó nézetek, illetve optimális kamerapozíciók felhasználásától is. Christie M. és társai azonban, az automatizált kameraelhelyezési technikákat összehasonlító kutatásukban [5] három csoportba osztották a különböző pozíciókból látható kép tulajdonságait geometriai, észlelési, illetve esztétikai jellemzők alapján. Úgy találták, hogy az általuk vizsgált algoritmusok általános megközelítése a felhasználó által megfogalmazott feltételeknek megfelelő geometriai tulajdonságok alapján történő legjobb pozíció kiválasztása. Ezért úgy gondoltuk, kézenfekvő lehet egy térbeli izooptikus alapján történő legjobb kamerahelyzet definiálása is.

A fejezet hátralévő részében bemutatunk egy új, speciális helyzetben lévő Bézier-felület izooptikus görbéjét meghatározó algoritmust, mely egy olyan kamerapályát alkot, ami segítségével a vizsgált felü-

letet a lehető legközelebből szemlélhetjük egy adott látószög figyelembevételével. Az eljárás leírását követően a izooptikus térgörbe meghatározására felírt számolásokat három különböző alfejezetben ismertetjük.

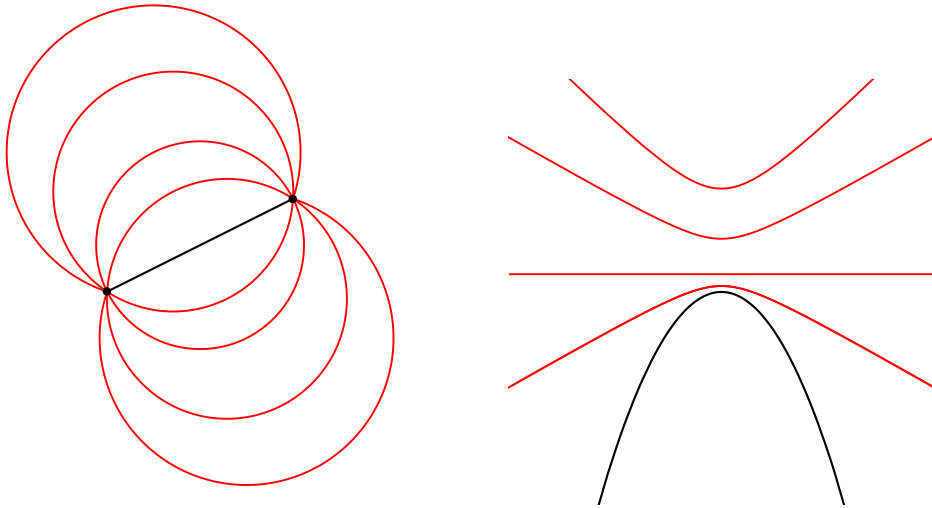
## 1.1. Előzmények

### 1.1.1. Síkgörbék izooptikus görbéi

**1. definíció.** *Adott síkgörbe izooptikus görbéjén azon pontok halmazát értjük, melyekből az eredeti görbe adott  $\alpha$  szög alatt látszik ( $0 < \alpha < \pi$ ). A derékszöghöz tartozó izooptikust ortooptikus görbének nevezzük [62].*

A definíció alapján másodfokú görbék esetén különböző klasszikus geometriai eredményeken alapuló számolási módszerekkel az izooptikus görbe egyszerűen meghatározható (1. ábra). További példák találhatóak [35] és [61]-ben, valamint kúpszeletek izooptikusainak meghatározásával kapcsolatos eredmények [55]-ben, illetve olyan görbékkel kapcsolatos kutatás, melyek izooptikusa kör [60, 30], vagy ellipszis [61]. Újabb eredmények közül megemlítendő Cieślak és társai munkája [6, 7], melyben zárt, konvex görbék izooptikusait határozzák meg támaszfüggvények segítségével. Emellett izooptikus görbék nemeuklideszi geometriákban történő meghatározására is találunk eredményeket, például [10] és [8]-ban.

Érdeemes megjegyezni azonban, hogy a definíció alapján történő direkt számoláson alapuló izooptikus meghatározása magasabb rendű görbék esetén rendkívül bonyolult is lehet. Ezért Kunkli és társai [29] egy új számolási módszert írtak le szabadformájú görbék izooptikusainak meghatározására. Az említett munkában konvex Bézier-görbék izooptikus görbéjét látókörivek burkolóinak a burkolójának kiszámolásával állítják elő. Habár a levezetés könnyen általánosítható más, például B-spline vagy NURBS görbére is, magasabb rendű, illetve bo-



1. ábra. Szakasz és parabola izooptikus görbéje  $\alpha = \frac{3\pi}{4}, \frac{2\pi}{3}, \frac{\pi}{2}$  és  $\frac{\pi}{3}$  esetben. A szakasz ortooptikus görbéje kör, a parabola ortooptikus görbéje a vezéregyenes.

nyolcultabb görbék esetén az izooptikus általában csak közelítő eljárással határozható meg.

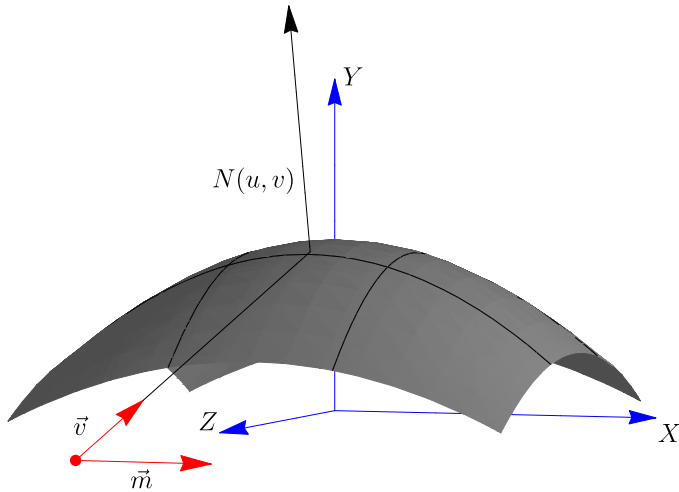
## 1.2. Új eredményeink

### 1.2.1. Izooptikus görbék térbeli általánosítása

A kétdimenziós izooptikus definíció általánosítása a háromdimenziós térben nem egyszerű és egyértelmű feladat. Nem triviális a látószög definiálása a 3D-ben. A síkgörbékhez hasonlóan meghatározhatók azonban olyan pontok, ahonnan az érintőegyenesek megadott szög alatt metszik egymást. Mivel ez a számolási módszer sem egyértelmű, speciális helyzetben adott Bézier-felületre keresünk térbeli izooptikus

pontokat.

A 2. ábra mutatja be ezt a speciális helyzetet. Feltételezzük, hogy a felület konvex és az  $Y = 0$  egyenlet által megadott alapsík felett, de ahhoz a lehető legközelebb helyezkedik el. Továbbá, hogy a koordináta-rendszer origója a felület alapsíkra vetített merőleges vetületére esik. Ezen alapsík azon pontjait fogjuk keresni a felület körül, amelyekből a legmeredekebb érintőegyenes és a sík által bezárt szög az előre megadott  $\alpha$  látószög. Ezeket a pontokat a felület izooptikus pontjainak nevezzük.



2. ábra. A Bézier-felület izooptikus pontjainak megtalálására szolgáló speciális helyzet ( $\vec{v} = [0 \ \sin \alpha \ -\cos \alpha]^T$ ,  $\vec{m} = [1 \ 0 \ 0]^T$ ).

Egy izooptikus pont meghatározásához két vektort használunk:  $\vec{m}$  és  $\vec{v}$ . Az  $\vec{m}$  vektor párhuzamos az  $X$  tengellyel, míg a  $\vec{v}$  vektor az  $X$  tengely körül  $\alpha$  szöggel el van forgatva. Ezt a szöveget úgy kell válasz-

tani, hogy a pontból az origó felé nézve a felület függőleges irányban elférjen a képernyőn. Egy izooptikus pontra igaz, hogy a hozzá tartozó  $\vec{m}$  és  $\vec{v}$  vektor merőleges a felület egyik normálvektorára. Ezzel a feltétellel több pozíciót találunk az  $Y = 0$  egyenletű síkon és ezen pontok halmaza egy izooptikus görbét alkot a vizsgált felület körül.

A görbét alkotó pontokat többféleképpen kereshetjük meg. A legegyszerűbben úgy, hogy egy adott lépésközzel forgatjuk az  $\vec{m}$  és  $\vec{v}$  vektort vagy az adott felületet az  $Y$  tengely körül és minden lépésben, egy brute-force keresés segítségével meghatározzuk azt az  $u$  és  $v$  paramétert, amelyhez tartozó Bézier-felületpontból húzott  $v$  vektor irányú érintőegyenest  $\alpha$  szöveget zár be az alapsíkkal. Ezen keresés optimalizálása, illetve függvényyszerű felírása nem képezte a munkánk részét, hanem inkább a különböző keresési lehetőségekből adódó egzakt izooptikust meghatározó képletek felírása volt a célunk, ezért a következő alfejezetekben ezen számolások részleteit ismertetjük bővebben.

### 1.2.2. A felület kontrollpontjainak forgatása

Első esetnek tekintsük azt, amikor a Bézier-felület kontrollpontjait forgatjuk az  $Y$  tengely körül. A felület a következő képlettel adott:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^n B_i^n(u) B_j^n(v) P_{i,j}, \quad (1.1)$$

ahol  $P_{i,j}$ ,  $i = 0, \dots, n$ ,  $j = 0, \dots, n$  a kontrollpontok, a  $B_i^n(u)$  és  $B_j^n(v)$  a Bernstein-polinomok. A felület normálvektorát az  $u$  és  $v$  szerinti parciális deriváltak segítségével határozhatjuk meg:  $U(u, v) = \frac{\partial S(u, v)}{\partial u}$  és  $V(u, v) = \frac{\partial S(u, v)}{\partial v}$ . Tehát a Bézier-felület  $u$  és  $v$  paraméterértékhez tartozó normálvektora  $N(u, v) = U(u, v) \times V(u, v)$  alakban számolandó. Ezek alapján felírható a Bézier-felület azon érintősíkjának egyenlete, amelynek normálvektora merőleges az  $\vec{m}$  és  $\vec{v}$  vektorra:

$$N_x(u, v)(X - X_0) + N_z(u, v)(Z - Z_0) = N_y(u, v)Y_0, \quad (1.2)$$

ahol az  $X_0, Y_0, Z_0$  a Bézier-felület  $u$  és  $v$  paraméterértékekhez tartozó pontjának koordinátái. Az izooptikus pont ezen sík és az  $Y = 0$  egyenletű alapsík metszéspontján van. Ez az egyenes az  $X$  tengellyel párhuzamos, ezért az (1.2) egyenletet  $X = 0$ -val megoldva kapjuk az origótól való távolságot, az izooptikus pont  $Z$  koordinátáját (az  $X$  és  $Y$  koordináta értéke nulla):

$$I_z = \frac{N_y(u, v)Y_0 - N_x(u, v)X_0}{N_z(u, v)} + Z_0. \quad (1.3)$$

Ezt a pontot ugyanazzal a szöggel kell elforgatnunk az  $Y$  tengely körül, amivel a felület kontrollpontjait is elforgattuk. A forgatás során a következő feltételeknek kell teljesülnie:  $\langle N(u, v), \vec{m} \rangle = 0$  és  $\langle N(u, v), \vec{v} \rangle = 0$ . Ez felbontva a következő:

$$\begin{aligned} N_x(u, v) &= 0 \\ \sin(\alpha) \cdot N_y(u, v) &= \cos(\alpha) \cdot N_z(u, v). \end{aligned} \quad (1.4)$$

Ha az egyenletrendszer meg tudjuk oldani  $u$ -ra és  $v$ -re, akkor a megfelelő normálvektorok képletét kapjuk, amiből az izooptikus pont pontosan kiszámolható. Sajnos a megoldást  $n = 2$  esetben sem tudjuk meghatározni, még komputeralgebrai programok segítségével sem. Kereső módszerrel viszont a megfelelő normálvektorok megtalálhatóak.

### 1.2.3. Az $\vec{m}$ és $\vec{v}$ vektorok forgatása

Az izooptikus pontok keresésének egy másik módja az  $\vec{m}$  és  $\vec{v}$  vektor elforgatása az  $Y$  tengely körül. Ebben az esetben a vektorok koordinátái a következők lesznek:

$$\begin{aligned} \vec{m} &= (\cos(\beta), 0, -\sin(\beta)) \\ \vec{v} &= (-\sin(\beta) \cdot \cos(\alpha), \sin(\alpha), -\cos(\beta) \cdot \cos(\alpha)), \end{aligned} \quad (1.5)$$

ahol  $\beta$  az  $Y$  tengely körüli forgatás szöge a  $[0, 2\pi]$  intervallumon. A keresési feltételeknek ( $\langle N(u, v), \vec{m} \rangle = 0$  és  $\langle N(u, v), \vec{v} \rangle = 0$ ) továbbra is teljesülniük kell. Ebből a következő egyenletrendszert kapjuk:

$$\begin{aligned} N_x(u, v) &= \tan(\beta) \cdot N_z(u, v) \\ N_y(u, v) &= \cot(\alpha) \cdot \frac{1}{\cos(\beta)} \cdot N_z(u, v). \end{aligned} \quad (1.6)$$

Mivel a képlet nem egyszerűbb, mint az (1.4) egyenlet, ezért az  $N(u, v)$ -t sajnos ezzel a módszerrel sem tudjuk pontosan megadni. Viszont a kereső módszerrel meghatározott normálvektorok esetén az izooptikus pontok közvetlenül kiszámolhatók (mivel nem kell forgatni a kontrollpontokat). Legyen  $e$  a felületet érintő sík és az  $Y = 0$  egyenlet által megadott alapsík metszévonalára, és  $f$  az az alapsíkbeli egyenes, amelynek egy pontja az origó, és normálvektora  $\vec{m}$ . Az izooptikus pontok ezen egyenesek metszéspontjában vannak. A koordinátákat a következő képletekkel lehet megadni:

$$\begin{aligned} I_x &= \frac{\sin(\beta) \cdot c}{\cos(\beta) \cdot N_z(u, v) - \sin(\beta) \cdot N_x(u, v)} \\ I_z &= \frac{\cos(\beta) \cdot c}{\cos(\beta) \cdot N_z(u, v) - \sin(\beta) \cdot N_x(u, v)}, \end{aligned} \quad (1.7)$$

ahol  $c = N_x(u, v)X_0 + N_y(u, v)Y_0 + N_z(u, v)Z_0$ . Az izooptikus pontok  $Y$  koordinátája minden esetben nulla. Mivel  $\langle N(u, v), \vec{m} \rangle = 0$ , a koordináták a következő egyenletekkel is számolhatók:

$$\begin{aligned} I_x &= \frac{cN_x(u, v)}{\langle N_x(u, v), N_x(u, v) \rangle + \langle N_z(u, v), N_z(u, v) \rangle} \\ I_y &= \frac{cN_z(u, v)}{\langle N_x(u, v), N_x(u, v) \rangle + \langle N_z(u, v), N_z(u, v) \rangle} \end{aligned} \quad (1.8)$$

#### 1.2.4. Számolás lineáris kombinációval

A megadott speciális helyzetben lévő Bézier-felületre meghatározható izooptikus pontok kiszámítására egy harmadik lehetőség is adott. Ezen módszerhez továbbra is az  $\vec{m}$  és  $\vec{v}$  vektorokat használjuk. Ha a felület normálvektora az  $U(u, v)$  és  $V(u, v)$  vektoriális szorzata, és  $N(u, v)$  merőleges az  $\vec{m}$  és  $\vec{v}$  vektorra, akkor  $\vec{v}$ ,  $\vec{m}$ ,  $U(u, v)$  és  $\vec{v}$ ,  $\vec{m}$ ,  $V(u, v)$  lineárisan függők, azaz  $U(u, v)$  és  $V(u, v)$  lineárisan kombinálható az  $\vec{m}$  és  $\vec{v}$  vektorból. Ez a következőket jelenti:

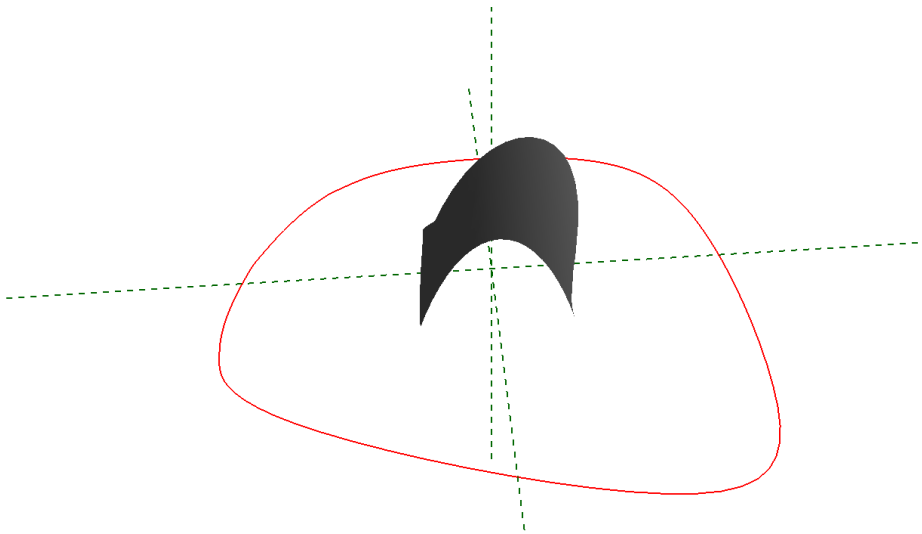
$$\begin{vmatrix} 1 & 0 & 0 \\ 1 & \sin(\alpha) & -\cos(\alpha) \\ U_x(u, v) & U_y(u, v) & U_z(u, v) \end{vmatrix} = 0, \quad (1.9)$$

$$\begin{vmatrix} 1 & 0 & 0 \\ 1 & \sin(\alpha) & -\cos(\alpha) \\ V_x(u, v) & V_y(u, v) & V_z(u, v) \end{vmatrix} = 0. \quad (1.10)$$

Ezek alapján a következő egyenletrendszert írhatjuk fel:

$$\begin{aligned} \sin(\alpha) \cdot U_z(u, v) + \cos(\alpha) \cdot U_y(u, v) &= 0 \\ \sin(\alpha) \cdot V_z(u, v) + \cos(\alpha) \cdot V_y(u, v) &= 0, \end{aligned} \quad (1.11)$$

melynek megoldásával  $u$ -ra és  $v$ -re másodfokú felület esetén egy komplex, ötödfokú egyenletet kapunk. Sajnos az eredmény gyökeinek közelítése, bonyolultságukból adódóan nem hozott eredményt. Az izooptikus pontok meghatározása ezért csak a felírt egyenletek alapján implementált keresőalgoritmussal lehetséges, melyek interpolációjából egy zárt B-spline görbét állíthatunk elő a felület körül (3. ábra).



3. ábra. Bézier-felület izooptikus görbéje.

## 2. Izooptikus felületek

Háromdimenziós izooptikus pontok a térben nem csak görbét határozhatnak meg, ahogyan azt az előző fejezetben láttuk, hanem a látószög megfelelő térbeli általánosításával egy adott alakzat körül egy izooptikus felület is definiálható. Egy ilyen általánosítási lehetőséget írt le Csima és Szirmai [9] a térszög fogalmának segítségével. Jelen fejezet első részében az általuk megadott definíciót ismertetjük és konvex poliéderek izooptikus felületét kiszámoló eljárásukat foglaljuk össze. Sajnos a bonyolult számolások csak komputeralgebrai program segítségével oldhatók meg, ezért az izooptikus felület meghatározása és megjelenítése már egyszerű, szabályos poliéderek esetén is rendkívül időigényes. Továbbá a módszer csak konvex poligonhálók esetén működik.

A fenti problémák megoldására kifejlesztettünk egy új algoritmust, mely képes az izooptikus felület gyors meghatározására konkáv alakzatok esetén is, komputeralgebrai szoftver használata nélkül. Az eljárás külön konvex és konkáv objektumokra felírt lépéseit a 2.2.1. alfejezetben részletezzük, melynek végén a bemutatott keresőmódszerek sebességét is összehasonlítjuk.

Az így meghatározott izooptikus felületek különböző tudományterületek esetén is hasznosak lehetnek, hiszen a definíció a fotometria területén (a fényerősség és fénysűrűség meghatározásánál [33]), vagy a nukleáris fizikában (detektorelhelyezés vagy sugárerősség kiszámításánál [18]) is használt térszög fogalmán alapul.

Ezen kívül természetesen az izooptikus felületek a láthatóságot érintő elméleti és gyakorlati problémák esetén is kiemelt szerepet kaphatnak, például stadionok, színházak, múzeumok, vagy akár tantermek tervezésekor, melyekre szemléletes ábrákat (2.3-2.5) találunk [20]-ban. Ezért definiáltunk egy alternatív izooptikus felületet is a síkbeli izooptikus fogalom új megközelítéssel történő térbeli általánosításával,

melyet a 2.2.2. alfejezetben ismertetünk.

Habár az általunk fejlesztett eljárás hatékonyan megkeresi az izooptikus felület pontjait, összetettebb alakzatok esetén további gyorsítások szükségesek, ezért kifejlesztettünk egy GPU-ra optimalizált, párhuzamos algoritmust is a Csima és Szirmai [9] definíciója alapján, melyet a 2.2.3. alfejezetben ismertetünk. Az eljárás az egyes párhuzamos lépéseket (térszögszámítás, pontkeresés) több szinten végzi el, ezért az egyes lépéseket külön szakaszokban tárgyaljuk. Az alfejezet utolsó részében a párhuzamos és szekvenciális eljárások teljesítményét hasonlítjuk össze.

## 2.1. Előzmények

### 2.1.1. A látószög térbeli általánosításával felírt definíció

Az izooptikus felületet Csima és Szirmai az alábbi elv alapján definiálta [9]: egy síkgörbe esetén az  $A$  pontban értelmezett szög mérhető az  $A$  pont köré húzott egységsugarú kör szögtartományba eső ívének hosszával. Ez a szög általánosítható a térre a térszög definíciója alapján [16]:

**2. definíció.** *Egy  $\mathcal{S}$  felület  $P$  pontban vett  $\Omega_{\mathcal{S}}(P)$  térszögén azt a mennyiséget értjük, mely az  $\mathcal{S}$  felület  $P$  középpontú egységömbre vetített képének felszínével egyenlő.*

Ezek alapján az izooptikus felületet a következőképpen határozták meg [9]:

**3. definíció.** *Egy háromdimenziós zárt  $\mathcal{D}$  tartomány  $\mathcal{H}_{\mathcal{D}}^{\alpha}$  izooptikus felülete azon  $P$  pontok halmaza, melyek esetén a  $\mathcal{D}$  tartomány  $P$  középpontú egységömbre vetített térszöge  $\alpha$  ( $0 < \alpha < 2\pi$ ).*

Tekintettel arra, hogy az egységgömb felszíne  $4\pi$ , az  $\alpha$  térszög természetes felső határa konvex alakzatok esetén  $2\pi$ , mivel a konvex alakzat vetítése az egységgömbre nem haladhatja meg a félgömböt. A felső határ konkáv esetben azonban  $4\pi$ , mivel a vetítés területe akár nagyobb is lehet, mint a félgömb.

### 2.1.2. Analitikus számolási módszer

Az említett szerzők leírtak egy konvex poliéderek izooptikus felületének meghatározására szolgáló számolást is. Ha adott egy zárt alakzat poligonok (lapok) és pontok (vertexek) halmazával, akkor az egységgömbre vetített területe kiszámolható azon gömbi sokszögek területével, amelyeket ezen poliédert alkotó poligonok lefednek. Egy adott pontban tehát a térszög az összes levetített gömbi sokszög területösszegének a fele lesz, mivel csak a pontból látható éleket vesszük. Ezek alapján egy  $\alpha$ -val adott izooptikus felület implicit egyenlete a következőképpen írható fel [9]:

$$\alpha = \frac{1}{2} \sum_{i=1}^m \Omega_i(P), \quad (2.1)$$

ahol  $m$  a poliédert alkotó lapok száma és az  $\Omega_i(P)$  a poliédert alkotó lapok  $P \in \mathbb{R}^3$  pontokban számolt térszöge.

Ezek a számítások azonban rendkívül bonyolultak még egyszerű, szabályos poliéderek esetén is. Ezért a (2.1) egyenlet megoldása és ábrázolása csak komputeralgebrai szoftverekkel lehetséges és tulajdonképpen „black box” módra viselkedik, illetve az izooptikus felület megjelenítése körülbelül 20–40 percet vesz igénybe. Továbbá a Csima és Szirmai által bemutatott módszer csak konvex poligonhálókra működik.

Az ismertetett eljárás a konkáv modellek egymást fedő lapjait nem kezeli, illetve nem definiálja az izooptikus felületet alkotó pon-

tok meghatározásának módját, ezért kutatásunkban elsődlegesen ezen hiányosságok pótlására fókuszáltunk.

## 2.2. Új eredményeink

### 2.2.1. Izooptikus felületek keresése

Az algoritmusban egy adott háromdimenziós poligonmodell körüli tér azon pontjait keressük meg, amelyekből a kiszámított térszög megegyezik az előre megadott mennyiséggel. A meghatározott pontokból standard felület-rekonstrukciós algoritmusok segítségével az izooptikus felület poligonhálóként előállítható.

Legyen adott egy  $\mathcal{M}$  zárt poliédermodell (háromszögháló) az  $\mathbb{E}^3$  Euklideszi térben egy  $\mathcal{F}$  sokszögeket tartalmazó listával (lapok), amelyben az egyes sokszögek pontjait a  $V$  csúcsok halmaza tartalmazza, az óramutató járásával ellentétes irányban megadva. Továbbá legyen adott az  $\mathcal{E}$  élek halmaza, amely a  $V$  elemeinek  $\{V_i, V_{i+1}\}$  párjaiból áll. A következőkben ilyen módon megadott 3D modellel felírt izooptikus felületet kereső algoritmusainkat mutatjuk be konvex, illetve konkáv esetben.

#### 2.2.1.1. Konvex modellek esetén

Először tekintsük az adott  $\mathcal{M}$  konvex modell egy tetszőleges  $P$  pontban vett  $\Omega(P)$  térszögének meghatározását. A számoláshoz először meghatározzuk az alakzat  $\mathcal{S}_{\mathcal{M}}(P)$  sziluettjét. Az  $\mathcal{S}_{\mathcal{M}}(P)$  azon élek halmaza, amelyekhez tartozó lapok normálvektorai közül pontosan az egyik zár be hegyesszöget a vizsgált élből a  $P$  pontba mutató vektorral. Az  $\Omega(P)$  térszög a  $P$  középpontú egységgömbre vetített azon sokszög területe lesz, melyet az  $\mathcal{S}_{\mathcal{M}}(P)$  élhalmaz lefed. Ez a következő képlettel határozható meg:

$$\Omega(P) = \theta - (n - 2)\pi, \quad (2.2)$$

ahol  $n$  a sziluettet alkotó csúcsok száma, illetve  $\theta$  a gömbi sokszög csúcsaiban számolandó szögek összege.

A modell tényleges vetítése az egységgömbre azonban nem szükséges a  $P$  középpontú gömbi poligon területének kiszámításához, mivel a poligon  $\vartheta_i$  szögeit közvetlenül számolhatjuk a vetítendő lapok segítségével. A  $\vartheta_i$  meghatározásához a szomszédos  $\mathcal{S}_{\mathcal{M}}(P)$ -beli élek ( $\{V_{i-1}, V_i\}$  és  $\{V_i, V_{i+1}\}$ ) által bezárt szöget kell kiszámolni. Ez [9] alapján:

$$\vartheta_i = \arccos \left( \frac{\langle \overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i, \overrightarrow{PV}_{i+1} \times \overrightarrow{PV}_i \rangle}{|\overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i| |\overrightarrow{PV}_{i+1} \times \overrightarrow{PV}_i|} \right). \quad (2.3)$$

Vegyük figyelembe, hogy a  $\vartheta_i$  kiszámításához rendezett  $\mathcal{S}_{\mathcal{M}}(P)$  élhalmazra van szükség, mely rendezést az élek végpontjainak összekapcsolásával egyszerűen elvégezhetjük. Az adott  $\mathcal{M}$  konvex modell  $P \in \mathbb{R}^3$  pontból látható  $\mathcal{S}_{\mathcal{M}}(P)$  sziluettjének  $\Omega(P)$  térszöge tehát:

$$\Omega(P) = \sum_{i=1}^n \arccos \left( \frac{\langle \overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i, \overrightarrow{PV}_{i+1} \times \overrightarrow{PV}_i \rangle}{|\overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i| |\overrightarrow{PV}_{i+1} \times \overrightarrow{PV}_i|} \right) - (n-2)\pi. \quad (2.4)$$

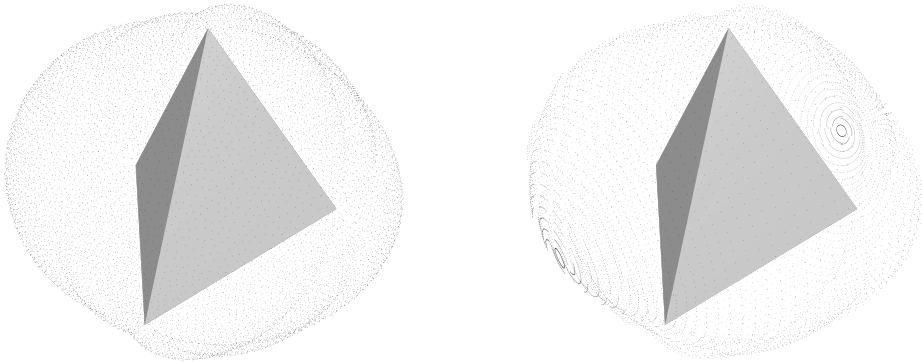
Az  $\mathcal{M}$  modell izoptikus felületének meghatározásához meg kell keresni azokat a háromdimenziós pontokat, ahol az  $\Omega(P)$  térszög megegyezik az előre megadott  $\alpha$  értékkel. A kereséshez a következő módszereket használhatjuk:

1. nyers erő (brute-force): a triviális megoldás, hogy bejárjuk a modell körüli térrészt egy adott lépésközzel és összehasonlítjuk a  $P \in \mathbb{E}^3$  pontokban kiszámolt  $\Omega(P)$  térszöget az előre megadott  $\alpha$  értékkel.
2. felület-kitöltés: a következő módszer alapja a jól ismert, síkbeli poligonok kitöltésénél használt elárasztásos kitöltés (flood-fill).

A keresésben egy adott lépésközzel vizsgáljuk egy korábban megtalált izooptikus felületpont szomszédos pozícióit. Az első pontot az alakzatunk egy belső pontjából egy egyenes mentén az alakzattól távolodva találhatjuk meg. Mivel a modell zárt, az egyenes biztosan metszi az izooptikus felületet.

- gömbi keresés: ebben a módszerben az előző eljárás első lépését hajtjuk végre a lehető legtöbb irányban. Ezen irányok meghatározásához egy, a modellt tartalmazó gömb használható, amelynek középpontja az alakzat súlypontja. A keresés során a különböző irányokban az optimális sugarat keressük.

A keresőalgoritmusok eredménye az adott konvex modell izooptikus felületének pontfelhője (4. ábra). Ezekből a pontokból standard felület-rekonstrukciós algoritmusok segítségével (például [27]) az izooptikus felület poligonhálóként előállítható.



4. ábra. Tetraéder izooptikus felületének pontfelhője a nyers erő és a felület-kitöltés módszerével (bal oldal), illetve gömbi kereséssel (jobb oldal)  $\alpha = \frac{\pi}{3}$  esetben.

Az első két módszer ugyanazzal a lépésközzel ugyanazt a pontfelhőt adja eredményül. Természetesen a felület-kitöltés algoritmus gyorsabb, mert a keresés során kevesebb térbeli pont tesztelése szükséges. Ezen algoritmusok elsődleges célja az adott modell izooptikus felületének megjelenítése. Ezzel szemben a gömbi keresés módszere jól használható a kamera pozíciójának kezelésére az izooptikus felület bejárása esetén.

Az eljárás a jól ismert Hughes és társai által leírt algoritmushoz [23] hasonló. A módszert úgy képzelhetjük el, mintha az adott modell egy nagy üveggömbbe lenne zárva. Egy pontot megjelölve rajta, a kamera az adott pozícióba megy és a modell abból az irányból látható. A gömb egy pontjának megjelölésén a gömb egyenletének  $\theta$  és  $\varphi$  paraméterének a meghatározását értjük:

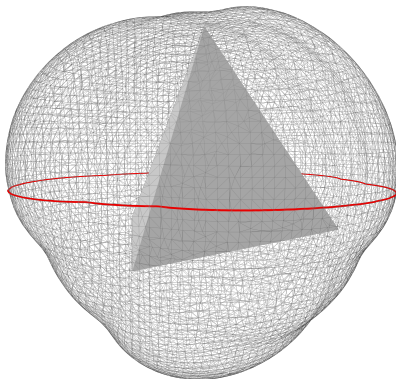
$$\begin{aligned}x &= C_x + r \cos \theta \sin \varphi \\y &= C_y + r \sin \theta \sin \varphi \\z &= C_z + r \cos \varphi,\end{aligned}\tag{2.5}$$

ahol  $r$  a gömb sugara,  $C$  pedig a középpontja. A bejáró algoritmus során a  $C$  pont a kamera fókuszpontja, illetve a kamerapozíciók az említett algoritmus gömbpontjai helyett a bemutatott gömbi keresés módszerével megtalált izooptikus pontok (5. ábra). Ez a megoldás egy új eszköz lehet interaktív CAD modellező eszközök számára.

Fontos megjegyezni, hogy amennyiben célunk az adott modell izooptikus felületének megjelenítése, a pontkeresés helyett az implicit izooptikus tesszellációja is lehetséges például marching cubes [34] vagy dual contouring [26] alkalmazásával.

### 2.2.1.2. Konkáv modellek esetén

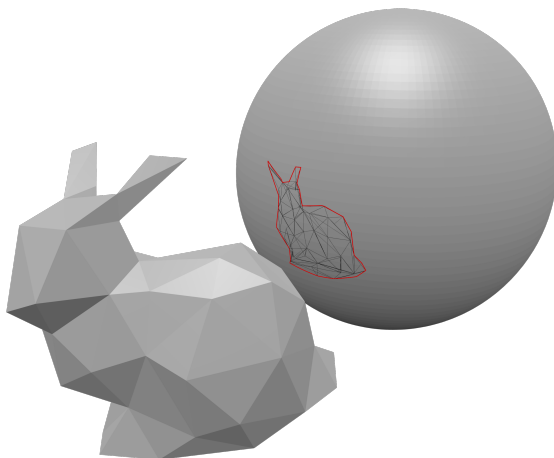
Az izooptikus felület pontjainak keresésére vonatkozó módszerek konkáv modellekre is alkalmazhatók. A térszög számításakor azonban figyelembe kell venni, hogy konkáv alakzatok esetén a gömbi vetület



5. ábra. Tetraéder  $\alpha = \frac{\pi}{3}$  izooptikus felületének pontjai rögzített  $\varphi = \frac{\pi}{2}$  esetén.

konkáv sokszöget képezhet az egységgömbön. Továbbá egy új algoritmust kell használni az adott modell  $P$  pontból látható körvonalának meghatározásához, nem elég a sziluett detektálása. A 6. ábrán látható egy adott  $\mathcal{M}$  modell  $\mathcal{C}_{\mathcal{M}}(P)$  kontúrja, mely nem egy részhalmaza a levetített  $\mathcal{E}$  élhalmaznak. A belső sziluettélek eltávolításán kívül, bizonyos esetekben előfordulhat, hogy a sziluettélek metszik egymást, hiszen konkáv alakzat esetén a modell gömbi képe részben fedheti egymást. Tehát az  $\mathcal{C}_{\mathcal{M}}(P)$  kontúr olyan térbeli szakaszok gömbi vetülete, melyek nem feltétlenül teljes élei  $\mathcal{M}$ -nek. Ezért az  $\mathcal{E}$  élhalmazt ténylegesen le kell vetíteni a  $P$  középpontú egységgömbre, és meg kell határozni az egymást fedő élek metszéspontjait. Továbbá a térszög kiszámolásához a  $\mathcal{C}_{\mathcal{M}}(P)$  óramutató járásával ellentétes irányú rendezése konkáv esetben is szükséges.

A kontúr kiszámításának megkönnyítése érdekében koordinátatranszformációt alkalmazunk úgy, hogy az origót az egységgömb középpontjába, azaz a  $P$  pontba helyezzük, és az  $\mathcal{M}$  modell vetületét



6. ábra. A Stanford Bunny egységgömbre történő vetítése.

az origó középpontú egységgömbön számoljuk. A kontúr meghatározásához elegendő a  $P$ -ből látható lapok csúcsainak vetítése. A  $\mathcal{C}_{\mathcal{M}}(P)$  élei az egységsugarú gömb főkörének szakaszai lesznek (6. ábra).

A  $\mathcal{C}_{\mathcal{M}}(P)$  kontúr számításának egyszerű módja hasonló a konvex alakzatok sziluettjének megtalálásához használt technikához. Csak azokat az éleket vetítjük le, amelyek pontosan egy látható laphoz tartoznak. Ezeknek az éleknek kiszámoljuk a metszéspontjait az egységgömbön. Ha ez a metszéspont egy belső pontja valamelyik szegmensnek, akkor a metsző éleket két-két új gömbi szakaszra kell bontani, megtartva az eredeti  $\mathcal{E}$  él óramutató járásával ellentétes sorrendjét. Ha  $e$  és  $f$  gömbi szakaszok ( $A, B$  és  $C, D$  végpontokkal) metszik egymást  $I$  pontban, az új szegmensek a következők lesznek  $\widehat{AI}$ ,  $\widehat{IB}$ ,  $\widehat{CI}$  és  $\widehat{ID}$ . Ezen élék közül ki kell választani azokat, amelyek ténylegesen

a kontúrt alkotják és nem pedig belső sziluettélek. Ehhez meg kell találni az összes irányított, zárt utat a levetített, illetve feldarabolt gömbi szakaszokon keresztül. Ezek közül az lesz az  $\mathcal{M}$  modell  $\mathcal{C}_{\mathcal{M}}(P)$  kontúrja, amelynek a legnagyobb a térszöge.

Két gömbi szakasz metszéspontjának egységgömbön történő kiszámításához a következő, a főkörívek metszéspontját meghatározó közismert képletet alkalmazhatjuk:

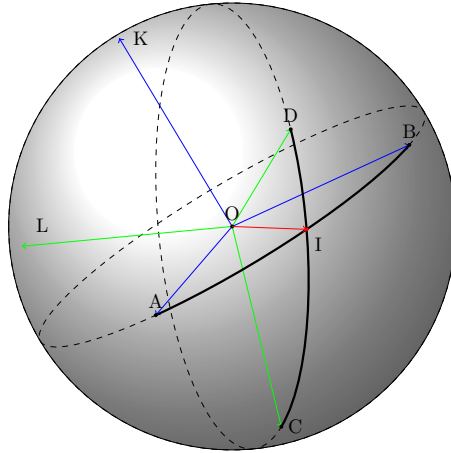
$$\vec{I} = \frac{(\vec{A} \times \vec{B}) \times (\vec{C} \times \vec{D})}{|(\vec{A} \times \vec{B}) \times (\vec{C} \times \vec{D})|}, \quad (2.6)$$

ahol  $\vec{I}$  vektor a főkörívek egyik metszéspontjára mutat és  $\vec{A}$ ,  $\vec{B}$  és  $\vec{C}$ ,  $\vec{D}$  vektorok pedig a gömbi szakaszok végpontjaira mutatnak (7. ábra). Mivel az  $\widehat{AB}$  és  $\widehat{CD}$  ívek hossza mindig kisebb, mint  $\pi$ , mert térbeli élek vetületei, ezért mindig a végpontok közötti rövidebb utat tekintjük.

A megfelelő metszéspont meghatározásához a következő teszt elvégzése is szükséges. Legyen  $\vec{K} = \vec{A} \times \vec{B}$  és  $\vec{L} = \vec{C} \times \vec{D}$ . Az  $\vec{I}$  vektor az  $A$  és  $B$  által meghatározott főkörre mutat  $A$  és  $B$  közé, ha a  $t_1 = \langle (\vec{K} \times \vec{A}), \vec{I} \rangle$  és  $t_2 = \langle (\vec{B} \times \vec{K}), \vec{I} \rangle$  előjele megegyezik. Ezért  $\widehat{AB}$  és  $\widehat{CD}$  akkor és csak akkor metszik egymást, ha  $t_1, t_2, t_3 = \langle (\vec{L} \times \vec{C}), \vec{I} \rangle$  és  $t_4 = \langle (\vec{D} \times \vec{L}), \vec{I} \rangle$  előjele megegyezik. Ha mindegyik előjel pozitív, akkor  $\vec{I}$  mutat a metszéspontra, ha mind negatív, akkor  $-\vec{I}$ .

Ha a  $\vec{K}$ , illetve  $\vec{L}$  vektor hossza nulla, az azt jelenti, hogy a szakaszok végpontjai egybeesnek vagy átellenesek. Esetünkben a végpontok nem lehetnek átellenes pontok, mert a gömbi távolságuk mindig kisebb, mint  $\pi$ . Ha pedig egybeesnek, akkor a lap, amihez tartoznak csak egy pontnak látható  $P$ -ből. Ezért ezeket a kivételes helyzeteket a további számításokból kihagyhatjuk.

Bonyolultabb konkáv modellek esetén érdemes egy fejlettebb algoritmust használni. A kontúr meghatározása az egységgömbön gya-



7. ábra.  $\widehat{AB}$  és  $\widehat{CD}$  gömbi szakaszok metszéspontjának kiszámítása.

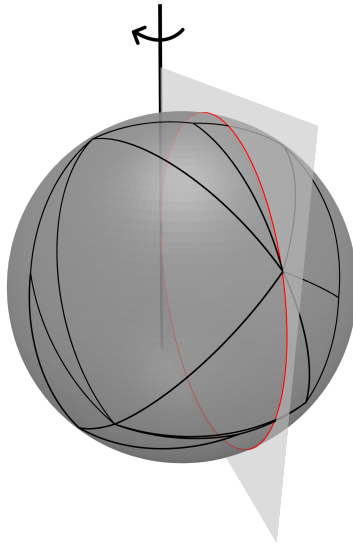
korlatilag gömbi sokszögek összegének kiszámítását jelenti. Ezen feladat elvégzéséhez több, a szakirodalomban jól ismert poligonok unióját előállító algoritmust hasonlítottunk össze ([19], [57], és [36]), illetve ezek gömbi implementációit terveztük meg. Habár a vizsgált eljárások mindegyike jól kezeli az önmetsző poligonokat (melyek esetünkben is előfordulhatnak), de a térszög kiszámításához rendezett sziluettélek szükségesek, ezért választásunk a Martínez és társai által fejlesztett algoritmusra [36] esett, mely az eredmény sokszög csúcsait az óramutató járásával ellentétes sorrendben állítja elő. Az eljárás a metszéspontok kiszámításához az úgynevezett söprővonal (sweep line) [2] elvét használja. Az algoritmus (és így a gömbi implementációja is) három fő lépést hajt végre két gömbi sokszög közötti unió kiszámításakor:

1. Feldarabolja a (gömbi) sokszögek éleit a metszéspontok mentén.

2. Kiválasztja azokat az éleket, amelyek a másik (gömbi) poligonon kívül esnek.
3. A megfelelő élek összekötésével létrehozza a (gömbi) kontúrt.

A levetített lapok uniójának additív kiszámolásával az eljárás végső eredménye az a gömbi sokszög lesz, amely az  $\mathcal{M}$  modell  $\mathcal{C}_{\mathcal{M}}(P)$  kontúrja.

A söprővonal technika gömbi megvalósítását a következőképpen képzelhetjük el: tegyük fel, hogy az egységgömböt egy forgó sík söpri végig, melynek forgástengelye a gömböt érintő egyenes, melynek egy pontja a  $P$  gömbi vetületével átellenes pont (8. ábra). A forgás során a sík a gömböt „söprőkörökben” metszi el, amely megoldás a síkbeli söprővonalhoz hasonló eljárást eredményez.



8. ábra. Egységgömb síkkal történő söprése.

Egy  $S$  eseményeket tartalmazó rendezett adatstruktúrában tároljuk azokat az éleket (gömbi szakaszokat), amelyek a söprő kört aktuálisan metszik a sík forgásirányával, azon belül pedig a forgástengely irányának megfelelően, fentről lefelé sorrendben. Az  $S$  struktúra tartalmát akkor kell módosítani, amikor a söprőkör eléri valamelyik él végpontját, vagy az élek egy metszéspontját. Ha két él metszi egymást azok  $S$ -ben szomszédosak lesznek. Ezenkívül egy  $Q$  elsőbbségi sor tartalmazza az összes, rendezett  $S$ -beli eseményeket. Két gömbi szakasz metszése esetén a metsző éleket felbontjuk és négy új elem kerül a  $Q$  struktúrába. A metszéspontok kiszámolásához a (2.6) egyenlet használandó.

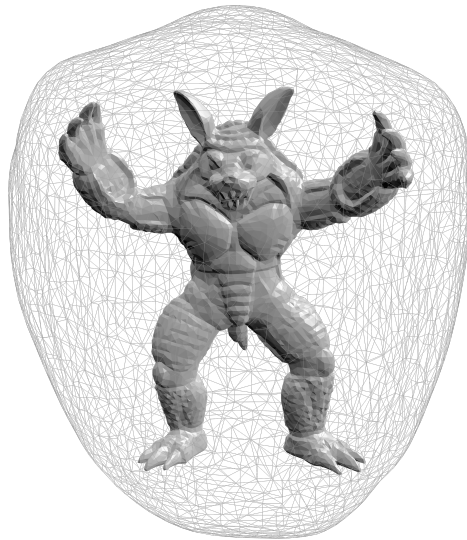
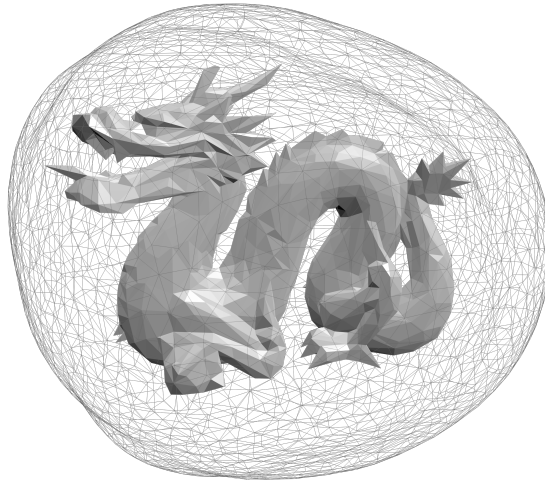
Két sokszög unióját azok az élek alkotják, amelyek kívül esnek a másik gömbi poligonon. Az összes sokszög metszetének kiszámolása után ezen gömbi szakaszok lesznek a  $\mathcal{C}_{\mathcal{M}}(P)$  kontúr elemei. A kiválasztáshoz egy úgynevezett sugarat használunk, amely a fix kiindulási pontjából egy egyenes mentén a végtelenbe tart. Esetünkben a söprő sík forgástengelyének megfelelő, függőleges irányban alulról felfelé tartó sugarat használunk. A megfelelő élek kiválasztása a következő feltétellel történik ([36] alapján): amikor egy  $e$  él bekerül az  $S$  struktúrába kiválasztjuk a sorrendben következő, másik poligonhoz tartozó  $f$  élt, ami az  $S$ -ben van, majd megvizsgáljuk, hogy ezen él a poligonba, amelyhez tartozik, egy kívülről-befelé vagy belülről-kifelé átmenetet képez (azaz a sugár a szakaszt elmetszve a poligon belsejéből kifelé megy vagy fordítva). Amennyiben az  $f$ -hez tartozó átmenet belülről-kifelé esik, akkor az  $e$  él az  $f$ -et tartalmazó poligonhoz képest kívül helyezkedik el, tehát az eredmény sokszöghöz tartozik. Mivel az  $\mathcal{M}$  modell lapjainak élei az óramutató járásával ellentétes sorrendben adódtak, az élek irányának megfelelő vektorok és a sugár irányából az átmenetek könnyedén meghatározhatók.

Az algoritmus további részei, azaz a kiválasztott élek megfelelő sorrendű összekapcsolása megegyezik a [36]-ban leírt eljárással, az implementáció változtatás nélkül alkalmazható gömbi poligonok esetén is.

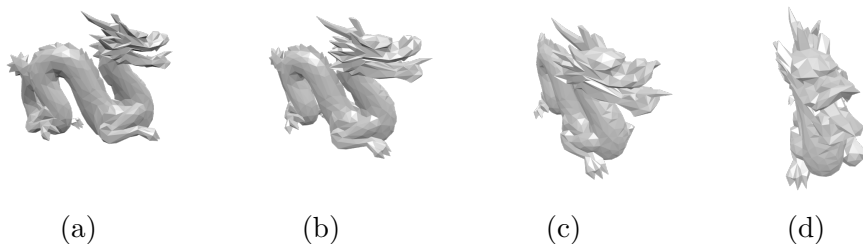
A rendezett  $\mathcal{C}_{\mathcal{M}}(P)$  meghatározása után az  $\Omega(P)$  térszög a konvex esetről tárgyalthoz hasonlóan számolandó. Azonban a gömbi poligon  $\vartheta_i$  szögeinek számolását a következőképpen kell kibővíteni, ugyanis konkáv kontúr esetén a belső szögek nagyobbak is lehetnek mint  $\pi$ :

$$\vartheta_i = \begin{cases} \arccos \left( \frac{\langle \overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i, \overrightarrow{PV}_{i+1} \times \overrightarrow{PV}_i \rangle}{|\overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i| |\overrightarrow{PV}_{i+1} \times \overrightarrow{PV}_i|} \right), & \text{ha } \langle \overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i, \overrightarrow{PV}_{i+1} \rangle > 0 \\ 2\pi \\ -\arccos \left( \frac{\langle \overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i, \overrightarrow{PV}_{i+1} \times \overrightarrow{PV}_i \rangle}{|\overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i| |\overrightarrow{PV}_{i+1} \times \overrightarrow{PV}_i|} \right), & \text{ha } \langle \overrightarrow{PV}_{i-1} \times \overrightarrow{PV}_i, \overrightarrow{PV}_{i+1} \rangle < 0 \end{cases} \quad (2.7)$$

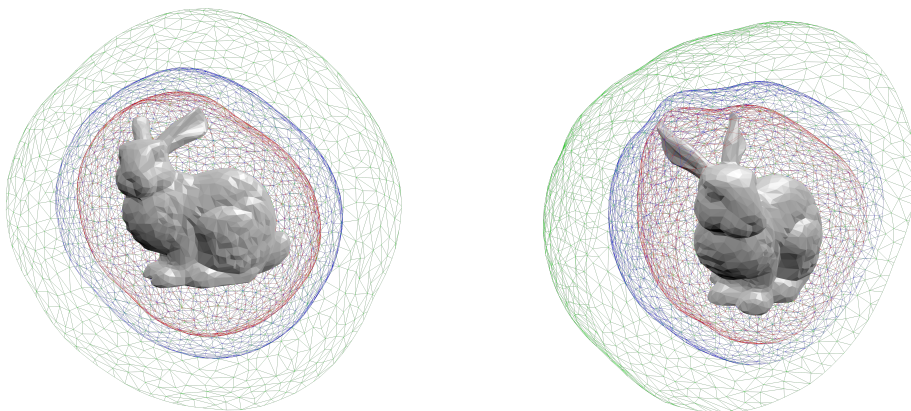
A leírt eljárással tetszőleges konkáv modell izooptikus felületének pontjai a konvex esetben tárgyalt keresési módszerek segítségével meghatározhatók, melyekből ugyancsak felület-rekonstrukciós algoritmusok segítségével az izooptikus felület poligonhálóként előállítható. A bemutatott módszer eredményei a 9. és 11. ábrán láthatóak. A 10. ábrán pedig a Dragon modell figyelhető meg az izooptikus felület különböző pontjaiból.



9. ábra. Dragon és Armadillo modell pontfelhőből rekonstruált izoop-  
tikus felülete  $\alpha = \frac{\pi}{3}$  esetén.



10. ábra. Dragon modell az izooptikus felület különböző pontjaiból ( $\alpha = \frac{\pi}{9}$ ). Megfigyelhető, hogy a modelltől látható rész ugyanakkora területet tölt ki az egyes képeken, azaz minél kevesebb részletet látunk a sárkányból (például a (d) ábrán szemből), annál közelebről szemlélhetjük.



11. ábra. Stanford Bunny modell pontfelhőből rekonstruált izooptikus felületeinek darabjai  $\alpha = \frac{\pi}{6}$  (zöld),  $\alpha = \frac{\pi}{3}$  (kék) és  $\alpha = \frac{\pi}{2}$  (piros) esetben.

### 2.2.1.3. Az algoritmusok sebessége

Az 1. táblázatban láthatóak a bemutatott algoritmusok futási idejei különböző komplexitású modellek esetén (a tesztekhez Intel Core i7-4790 processzort használtunk). Valamennyi módszer, adott  $\alpha = \frac{\pi}{3}$  mellett legalább 900 izooptikus pontot talál meg, amely megfelelő a használt felület-rekonstrukciós eljárásnak [27] az izooptikus felület poligonhálóként történő előállításához. A megadott lépésköz minden esetben 0,05 volt, és egy  $P$  pontot akkor fogadtunk el, ha a kiszámított térköz és a megadott  $\alpha$  abszolút eltérése kevesebb volt mint  $2 \times 10^{-3}$  (hibahatár).

Megfigyelhető, hogy a futási idő lineárisan növekszik a modelleket alkotó lapok számával. Továbbá az izooptikus pontok száma is lineárisan növekszik az eltelt idő függvényében. A második és a harmadik algoritmus legalább kétszer gyorsabb, mint a nyers erő algoritmus, mert ezek kevesebb térbeli pontot vizsgálnak a keresés során. A gömbi keresés esetén a megtalált izooptikus pontok száma pontosan 900, mert a (2.5) egyenlet  $\theta$  és  $\varphi$  paraméterét  $\frac{\pi}{6}$ -ra és  $\frac{\pi}{12}$ -re osztjuk fel, azaz  $30 \times 30 = 900$  sugarat lövünk a modell középpontjából. Ez a módszer a felület-kitöltésnél is gyorsabb lehet az adott felület alakjától és a felosztástól függően, ha a sugarak hamar elérik az izooptikust (például a 9. ábrán az izooptikus felület közelebb van a Dragon modell fejéhez).

### 2.2.2. Alternatív izooptikus felület definiálása

Az előző alfejezetekben az izooptikus felület meghatározására a Csima és Szirmai által megadott definíciót [9] alkalmaztuk. A leírt algoritmusok esetén az általuk megadott térbeli izooptikus meghatározást vettük alapul. Ugyanakkor a térszög számolása helyett alternatív értelmezések is lehetségesek. Mi egy új értelmezést is bevezettünk, amellyel ugyanaz a megadott modell egy másik izooptikus felülete határozható meg. Ha egy kétdimenziós izooptikus pontot a megfelelő

Modell	Nyers erő	Felület-kitöltés	Gömbi keresés
<b>Tetraéder</b> Csúcsok: 4, Lapok: 4	5.6	2.7	1.6
<b>Stanford Bunny</b> Csúcsok: 66, Lapok: 128	163.7	55.2	48.9
<b>Stanford Bunny</b> Csúcsok: 130, Lapok: 256	397.6	123.7	131.7
<b>Dragon</b> Csúcsok: 222, Lapok: 444	910.9	339.5	244.9
<b>Armadillo</b> Csúcsok: 352, Lapok: 700	1390.4	438.1	527.1

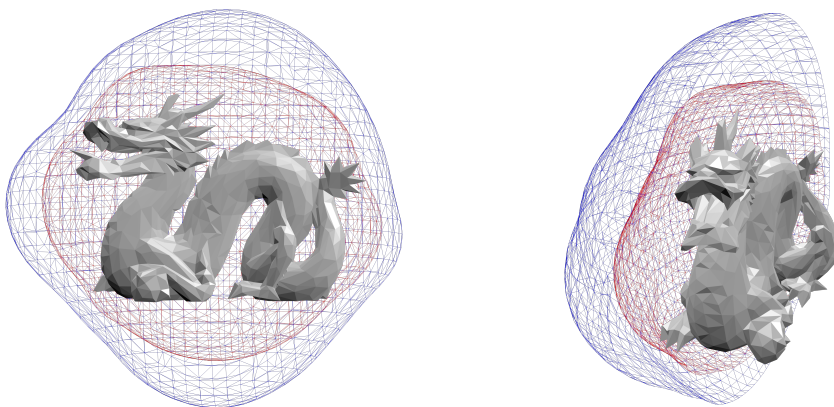
1. táblázat. A különböző keresési módszerek futási ideje (másodpercben). Mindegyik algoritmus legalább 900 pontját találja meg az izooptikus felületnek. Megjegyzés: a tetraéder izooptikus felületének a Csima és Szirmai [9] által leírt analitikus módszerrel történő kiszámolása és megjelenítése 20 percet vett igénybe, Wolfram Mathematica [24] használatával.

érintőegyeneseink metszéspontjaként, azaz a „legszeleesebb nézetként” értelmezzük, akkor a háromdimenziós általánosítása ugyanezen legszeleesebb térbeli nézet lesz, ami az adott alakzat nézőpont körüli egységgömbi vetületének a legszeleesebb átmérője.

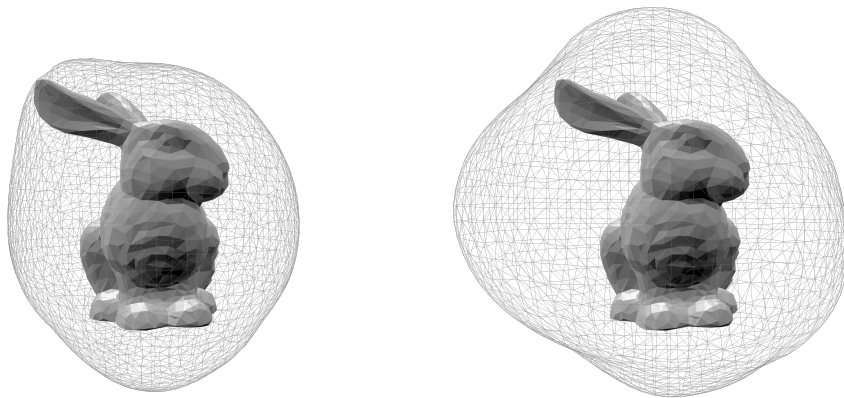
**4. definíció.** *Egy háromdimenziós zárt  $\mathcal{D}$  tartomány  $\mathcal{H}_{\mathcal{D}}^{\alpha}$  izooptikus felülete azon  $P$  pontok halmaza, melyek esetén az egységgömbre vetített  $\mathcal{D}$ -t alkotó pontok legnagyobb gömbi távolsága megegyezik a megadott  $\alpha$  értékkel.*

Az új értelmezés alapján meghatározott izooptikus felület látható

a 12. ábrán. Megfigyelhető, hogy az alternatív definícióval ugyanannak a modellnek megegyező  $\alpha$ -val megadott, mégis jelentősen különböző alakú izooptikus felülete határozható meg (13. ábra). Megjegyezzük azonban, hogy nincs egyértelműen „jó” vagy „rossz” térbeli izooptikus, hiszen az adott problémától függ, hogy melyik definíció írja le jobban a problémát, melyik felületet érdemes alkalmazni. Fontos kiemelni továbbá, hogy az új definíció szerinti izooptikus felület meghatározása nagyságrendekkel gyorsabb, mint a Csima és Szirmai-féle definícióval megadott izooptikus, ugyanazon keresőalgoritmusok segítségével történő előállítás, mely még hatékonyabb lehet, ha a modell vetületének kontúrpointjai között keressük a maximális gömbi távolságot, nem pedig az  $\mathcal{M}$  összes levetített csúcsa között.



12. ábra. A Dragon modell új definíció alapján pontfelhőből rekonstruált izooptikus felületeinek darabjai  $\alpha = \frac{\pi}{2}$  (kék) és  $\alpha = \frac{2}{3}\pi$  (piros) esetben.



13. ábra. Stanford Bunny eredeti (bal oldal) és új (jobb oldal) definíció alapján meghatározott izooptikus felülete  $\alpha = \frac{\pi}{2}$  esetben.

### 2.2.3. Izooptikus felületek keresésének párhuzamosítása

A 2.2.1. fejezetben bemutatott 3D modellek izooptikus felületének meghatározására felírt algoritmusunk a keresés során először a pont körüli egységgömbre vetített kontúrt keresi meg, majd kiszámolja a vetület térszögét. Az 1. táblázat alapján látható, hogy az eljárások, habár sebességben felülmúlják a megelőző módszert [9], összetettebb (több száz, akár több ezer poligonból álló) alakzatok esetén nem elég hatékonyak, ezért további gyorsítások szükségesek. A következőkben egy párhuzamos feldolgozású, GPU-ra tervezett algoritmust ismertetünk, mely a keresés során elsődlegesen a térszög kiszámítására fókuszál, nem pedig a gömbi kontúr meghatározására. A parallel lépéssorozatokat implementálásához az NVIDIA Compute Unified Device Architecture-t (CUDA) használtuk, amely közvetlen hozzáférést biztosít a grafikus kártya virtuális utasításkészletéhez és párhuzamos számítási elemekhez. Az algoritmus lépései speciális, úgynevezett kernel függvényeket hívnak meg. Minden ilyen hívás elindít egy meghatározott számú GPU

szálat (feldolgozási egységet). Ezek a szálak a megadott adatokon hajtják végre a kernel függvény által meghatározott műveleteket.

A térszöget számító eljárás mellett a megfelelő háromdimenziós pontok meghatározására használt keresési módszer is felgyorsítható párhuzamos feldolgozással, ezért az izooptikus felület meghatározásához használható új algoritmus beágyazott kernelhívásokat igényel. Ez a többszintű párhuzamos feldolgozás, amit a CUDA-ban dinamikus párhuzamosságnak hívnak [25], lehetővé teszi a szálak számára új kernelindítás létrehozását és szinkronizálását. A párhuzamosság különböző szintjeit a következő alfejezetekben külön-külön tárgyaljuk.

### 2.2.3.1. Térszögszámítás párhuzamosítása

Tegyük fel, hogy az adott  $\mathcal{M}$  modell a 2.2.1. fejezetben leírtak alapján egy  $\mathcal{F}$  lapokat,  $\mathcal{E}$  éleket és  $V$  csúcsokat tartalmazó halmazokkal adott. Ezen ismeretek alapján a megadott lapok normálvektorai könnyedén meghatározhatóak. Továbbá, az  $\mathcal{E}$  éllista elemeinek struktúrájában helyezzük el a `v_1[3]`, `v_2[3]` végpontok koordinátái mellett a másik lap normálvektorát is, (`other_normal[3]`), amelyhez az adott él tartozik (a  $P \in \mathbb{E}^3$  pontokból látható élek gyorsabb meghatározása érdekében):

---

```

edge = {
    v_1[3]           : float ;
    v_2[3]           : float ;
    other_normal[3] : float ;
}

```

---

1. kódrészlet. Az  $\mathcal{E}$  éleket tartalmazó adatszerkezet elemeinek struktúrája.

Emellett érdemes a már említett koordináta-transzformációt is alkalmazni a további számolások könnyítése érdekében, hogy az origót

az egységgömb középpontjába, azaz a  $P$  pontba helyezzük, és az  $\mathcal{M}$  modell vetületét az origó középpontú egységgömbön számoljuk.

Egy adott  $P$  pont  $\Omega(P)$  térszögének párhuzamos feldolgozású kiszámításához a következő négy lépés szükséges:

### I. Élek vetítése az egységgömbre

Az első lépésben levetítjük az origó középpontú egységsugarú gömbre az  $\mathcal{M}$  modell  $P$  pontból látható lapjainak éleit. Annak elkerülése érdekében, hogy azonos végpontú, de ellentétes irányú vetületek keletkezzenek, csak azokat az éleket vetítjük le, amelyek pontosan egy látható laphoz tartoznak. A megfelelő élek kiválasztása az 1. kódrészletben leírt struktúra `other_normal[3]` elemének felhasználásával hatékonyan történhet. A lépés során a megfelelő levetített élekből egy  $\mathcal{S}$  tömböt töltünk fel, melyek elemei az origóközéppontú egységgömb főköreinek szakaszai lesznek, a következő struktúrával:

---

```
spherical_edge = {
    A[3]           : float ;
    B[3]           : float ;
    dist_from_center : float ;
    intersection_id[max_int] : integer ;
    n_int          : integer ;
}
```

---

2. kódrészlet. Az  $\mathcal{S}$  gömbi szakaszokat tartalmazó tömb elemeinek struktúrája.

Az  $A[3]$  és  $B[3] \in \mathbb{E}^3$  (továbbra is Descartes-féle koordinátákkal adott) pontok a levetített él végpontjai. A `dist_from_center` a  $B[3]$  pont és az alakzat középpontjának vetülete közötti távolság, amelyre az algoritmus következő szakaszában lesz szükség, de az

értékét ebben a lépésben számítjuk ki. Lehet gömbi vagy euklideszi távolság, bár a trigonometrikus függvények elkerülése érdekében az utóbbi javasolt. A struktúra további részeit később ismertetjük.

Az  $\mathcal{S}$  tömb feltöltése párhuzamosan is lehetséges, mivel a modell élei egymástól függetlenül feldolgozhatók. A számolások eredményének tárolásához, azaz a megfelelő mennyiségű memória lefoglalásához szükséges azonban az  $\mathcal{S}$  méretének fentről történő közelítése ( `max_s` a 4. kódrészletben), mely lehet egyszerűen a modellt alkotó lapok száma, azaz  $\mathcal{F}$  mérete, szorozva a lapot alkotó élek számával (ami három, háromszögelt modellek esetén). Ez azonban meglehetősen memóriapazarló, hiszen a sziluettet alkotó élek száma az eredeti élek csupán töredéke, ezért érdemes egy pontosabb becslést használni. A [28] szerzői készítettek egy valószínűségi elemzést a sziluettélek várható számáról egy véletlenszerűen választott kamerapozícióból, amely felhasználható az  $\mathcal{S}$  méretének közelítésére:

$$\sum_{e \in \mathcal{E}} 1 - 2\phi_e, \quad (2.8)$$

$$\phi_e = \frac{1}{2\pi} \arccos \frac{-\vec{n}_{f_i} \cdot \vec{n}_{f_j}}{|\vec{n}_{f_i}| |\vec{n}_{f_j}|}.$$

Az  $\vec{n}_{f_i}$  és  $\vec{n}_{f_j}$  az  $f_i$  és  $f_j \in \mathcal{F}$  lapok normálvektorai. A  $\phi_e$  pedig annak a valószínűsége, hogy az  $e \in \mathcal{E}$  élhez tartozó mindkét  $f_i$  és  $f_j$  lap normálvektora hegyesszöget zár be az élből a  $P$  pontba mutató vektorral, vagy mindkettő tompaszöget.

Ezenkívül az algoritmus további lépéseihez az  $\mathcal{S}$  pontos mérete is szükséges, ezért a feldolgozás során tárolni kell az aktuális  $n_{\mathcal{S}}$  elemszámot is. Mivel a tömb feltöltése több szál által, egyidejűleg történik, az elemek számolása csak úgynevezett atomi művelettel (összeadás) lehetséges, amely kiolvassa az adott értéket a meghatározott memóriacímről, hozzáad egy számot (esetünkben egyet), majd az eredményt visszaírja ugyanarra a címre. Az atomi művelet garantálja, hogy a leírt műveletsor egy lépésként történik meg, az-

az a végrehajtása nem szakítható meg konkurens műveletek által [49].

## II. Gömbi szögek kiszámítása a metszéspontokban

Ebben a lépésben kiszámoljuk az összes egymást fedő  $e$  és  $f \in \mathcal{S}$  élpár gömbi szögét a metszéspontokban, illetve azt, amikor az ellentétes végpontban találkoznak ( $\mathbf{A}[3]_e = \mathbf{B}[3]_f$  vagy  $\mathbf{B}[3]_e = \mathbf{A}[3]_f$ ). A számolást nyers erő módszerével (az összes gömbi szakaszt páronként teszteljük), párhuzamosan végezzük. Két levetített él metszéspontját a 2.2.1. fejezetben leírt (2.6) egyenlettel számolhatjuk ki. Ismeretesek gyorsabb, GPU-ra tervezett szakaszmetező algoritmusok is (például [52]), de figyelembe véve a többszintű párhuzamosságot, az egyszerűbb nyers erő módszerének használata ajánlott.

A gömbi szögek mellett az algoritmus következő lépéseinek további adatokra is szüksége van, ezért ha  $e$  és  $f$  metszik egymást, akkor egy  $\mathcal{I}$  tömbben két elemet állítunk be a következő struktúra alapján:

---

```
intersection = {  
    angle      : float ;  
    other_edge : integer ;  
    dist_from_A : float ;  
}
```

---

3. kódrészlet. Az  $\mathcal{I}$  metszéspontokkal kapcsolatos adatokat tároló tömb elemeinek struktúrája.

Az  $e$ -nek megfelelő elemekben tároljuk a gömbi szöget  $e$  és  $f$  között (`angle`), melynek kiszámolásához a (2.7) egyenlet használandó, melyet ugyancsak a 2.2.1. fejezetben ismertettünk. Az `other_edge` az  $f$  él  $j$  indexe az  $\mathcal{S}$  tömbben. A `dist_from_A` a metszéspont és az

$e$  él első ( `A[3]` ) végpontjának távolsága, mely szintén számolható gömbi, illetve euklideszi távolságként is. A másik,  $f$  élnek megfelelő elemben a `2π - angle` szöveget, az  $e$   $\mathcal{S}$ -beli  $i$  indexét, valamint az  $f$  első pontja és a metszéspont közötti távolságot tároljuk.

Az említett számolások eredményeit párhuzamosan határozzuk meg, egy szálat használva minden egyes  $e$  és  $f$  párhoz. Az  $\mathcal{I}$  struktúrát egy egydimenziós tömbként tároljuk a globális memóriában, melynek mérete a (2.8) egyenlet segítségével megbecsült  $\mathcal{S}$  terjedelmének négyzete. A konkurens feldolgozás során felmerülő írási ütközések az  $\mathcal{S}$ -beli  $i$  és  $j$  indexek segítségével elkerülhetők, ha az  $e$ -nek megfelelő új elem pozíciója  $\mathcal{I}$ -ben  $n_{\mathcal{S}} \cdot i + j$ , és az  $f$ -nek megfelelő elem pozíciója  $n_{\mathcal{S}} \cdot j + i$  lesz.

A kiszámolt  $\mathcal{I}$  tömb elemeinek indexeit tárolnunk kell az  $\mathcal{S}$  gömbi szakaszokat tartalmazó tömb elemeinek struktúrájában is ( `intersection_id[max_int]` a 2. kódrészletben). Mivel egy gömbi szakasz több másikat is keresztezhet, ezért az indexeket ugyancsak tömbként kell tárolni, mely `max_int` méretének egy felső becslését a következőképpen határozhatjuk meg. A [28] 3. fejezetében egy részletes számolást találunk két él metszésének valószínűségére egy véletlenszerűen választott kamerapozícióból. Ezek közül kell a legnagyobb valószínűségűt tekinteni. Emellett, mivel az ellentétes végpontbeli metszések szögeit is kiszámoljuk, meg kell határozni a modellnek azt a csúcspontját, amelyben a legtöbb oldal találkozik. Ezen számértéket kétszer kell hozzáadni a kiszámolt valószínűséghez, hogy az élek mindkét oldalát tekintsük. A felső becslés mellett a pontos elemszám ( `n_int` a 2. kódrészletben) meghatározásához a már ismertett atomi összeadás műveletét használhatjuk.

### III. Kezdő él megtalálása

Az  $\mathcal{S}$  bejárásához meg kell határozni a kezdő  $e$  gömbi szakaszt, amely az  $\mathcal{S}$  tömbnek biztosan egy kontúr éle. A megfelelő elem az lesz, amely első ( `A[3]` ) végpontja a legmesszebb van az  $\mathcal{M}$  mo-

dell középpontjának  $C$  vetületétől. Fontos azonban, hogy az  $A[3]_e$  pontnak megfelelő modelcsúcs a vizsgált  $P$  pontból látható legyen, azaz ne takarja  $\mathcal{M}$  valamelyik lapja. Ez akkor teljesül, ha az  $A[3]_e$  és a  $C$  középpontvetület  $C'$  ellenpólusa által meghatározott  $C'A[3]_e$  gömbi szakaszt nem metszi az  $\mathcal{S}$  éltömb egyetlen eleme sem. Mivel a kritériumnak azok a gömbi élek is megfelelnek, amelyek nem kontúr, hanem belső sziluettélei a modellnek, ezért ezek közül azt kell kiválasztani, amelyiknek a legnagyobb a  $\sphericalangle C'B[3]_eA[3]_e$  gömbi szöge.

Mivel  $\mathcal{I}$ -ben az élek metszéspontjai mellett az egybeeső ellentétes végpontok is tárolásra kerülnek, ezért érdekesebb a  $C$  középpontvetülettől a legtávolabbi  $P$ -ből látható  $B[3]_f$  pontot megkeresni, hiszen így a legnagyobb  $\sphericalangle C'B[3]_eA[3]_e$  szögű  $e$  kezdő él gyorsabban meghatározható az  $\mathcal{S}$  ismételt bejárása nélkül, az  $f$  elemekben lokálisan tárolt `intersection_id[max_int]` (a 2. kódrészlet) metszéspontindexek segítségével. A legtávolabbi  $B[3]_f$  elemet az előző lépésben kiszámolt `dist_from_center` érték segítségével könnyedén megtalálhatjuk.

Ebben a lépésben a láthatóság tesztelése történhet egyidejűleg. A megfelelő  $B[3]_f$  keresésekor az adott  $C'B[3]_f$  gömbi szakasz  $\mathcal{S}$ -beli élekkel történő metszését vizsgálhatjuk párhuzamosan.

#### IV. Térszögek összegének kiszámolása

Az utolsó lépés a megfelelő  $\mathcal{S}$ -beli élpárok gömbi szögösszegének kiszámítása és a térszög meghatározása a (2.2) képlet segítségével. A kiválasztott kezdő éltől indulva végighaladunk az egymást metsző gömbi szakaszok mentén, az  $\mathcal{I}$  tömbben tárolt adatok segítségével. Egy adott szakasz rákövetkező elemét lokálisan az `intersection_id[max_int]`-ben tárolt metszéspont-indexeket bejárva a következőképpen választjuk ki. Először mindig az első végponthoz legközelebbi metszéspontot tekintjük, megvizsgálva

a 3. kódrészlet `dist_from_A` értékét. A következő elem  $\mathcal{S}$ -beli indexét az `other_edge` tartalmazza. Ha a metszéspontok egybeesnek (azonos `dist_from_A` esetén) azt az elemet kell választani, amelyik az aktuális éllel a legnagyobb gömbi szöget zárja be a 3. kódrészlet `angle` értéke alapján.

Egy aktuális  $f$  gömbi él rákövetkezőjének meghatározásakor a távolság vizsgálata során az összehasonlítás nem feltétlenül nulláról indul, hanem egy minimális értéktől, attól függően, hogy az őt megelőző  $e$  szakasz az  $f$  élt hol metszi, azaz az  $f$   $A[3]$  pontjától mekkora távolságra. Mivel egy  $e$  és  $f$  szakasz metszése esetén két elem kerül az  $\mathcal{I}$  tömbbe, melyek esetén a különböző gömbi szakaszok első végpontjától és a metszésponttól való távolságát tároljuk, ezért az említett minimális érték meghatározható a megfelelő  $\mathcal{I}$ -beli elem `dist_from_A` értéke segítségével. Ha az  $e$ -nek megfelelő szakasz  $\mathcal{I}$ -beli indexe  $i$ , akkor a rákövetkező  $f$ -nek megfelelő  $\mathcal{I}$ -beli  $j$  index a következőképpen számítható ki:  $n_{\mathcal{S}}(i \bmod n_e) + (i/n_{\mathcal{S}})$ , ahol  $n_{\mathcal{S}}$  az  $\mathcal{S}$  éltömb mérete.

Az iteráció addig tart, amíg vissza nem érünk a kezdő élhez. Mivel minden egyes lépésben a már kiszámolt gömbi szöget hozzáadjuk a (2.2) képlet  $\theta$  értékéhez, ezért az utolsó szakasz csak szekvenciálisan dolgozható fel. A [28] alapján a kontúréleknek csak néhány metszése van (esetünkben az `n_int` értéke alacsony), ezért nem érdemes az élmetszéseket sorbarendezeni, mivel iteratívan is gyorsan feldolgozhatóak.

A térszögszámoló eljárás (mind a négy lépését összefoglaló) pszeudokódját a Függelékben található 4. kódrészlet tartalmazza.

### 2.2.3.2. Keresési módszerek párhuzamosítása

Az előző alfejezetben a térszög párhuzamos lépésekkel kibővített számolásának négy fő lépését írtuk le. A sebesség mellett a bemutatott algoritmus egyszerűsége is fontos, mivel az új párhuzamos ke-

resési módszerek segítségével egyszerre több  $P \in \mathbb{E}^3$  pontra szeretnénk térszöget számolni. Ez azt jelenti, hogy az izooptikus pontfelhő meghatározására szolgáló összes számítást egyetlen, teljes egészében a grafikus kártyán futó kernelhívásba ágyazzuk. Ez az eljárás azért hatékony, mert így a folyamatot nem szakítják meg különböző memóriakezelési műveletek, hanem az összes szükséges memóriaterület előre lefoglalható, illetve a megfelelő  $\mathcal{M}$  modellt leíró adatok betölthetőek a memóriába a kernelhívás előtt.

A 2.2.1. fejezetben bemutatott keresési módszerek párhuzamos feldolgozása a következő módosításokkal lehetséges:

1. nyers erő (brute-force): az  $\mathcal{M}$  modell körüli, adott lépésközzel bejárando térrész szomszédos  $P \in \mathbb{E}^3$  pontjait egyszerre vizsgáljuk. A térrészre értelmezünk egy kockarácsot, hasonlóan a masírozó kockák (marching cubes) algoritmushoz [34].
2. gömbi keresés: ebben a módszerben, a  $\mathcal{M}$  középpontjából kiindulva, egyidejűleg több irányban meghatározott keresést indítunk el és az egyes irányokat külön GPU szálakban dolgozzuk fel.

Sajnos a felület-kitöltés módszere nem gyorsítható fel a grafikus kártya használatának segítségével. Az előző módszerek esetén a párhuzamosítás könnyen és egyszerűen értelmezhető. A felület-kitöltés nehézsége azonban, a már bejárt pontok nyomon követése. A CPU-ra optimalizált szekvenciális esetben a folyamat bináris keresőfa segítségével felgyorsítható. Annak ellenére, hogy hasonló megoldások GPU-ra is léteznek (például [48]), az új térszögszámító eljárással együtt a folyamat túl sok GPU erőforrást (elsősorban szálát) használ, ezért a felület-kitöltő módszer GPU implementációja hosszabb futási időt eredményez.

A párhuzamosított eljárások nyilvánvalóan ugyanazokat az izooptikus felületeket állítják elő, mint a szekvenciálisan implementált vál-

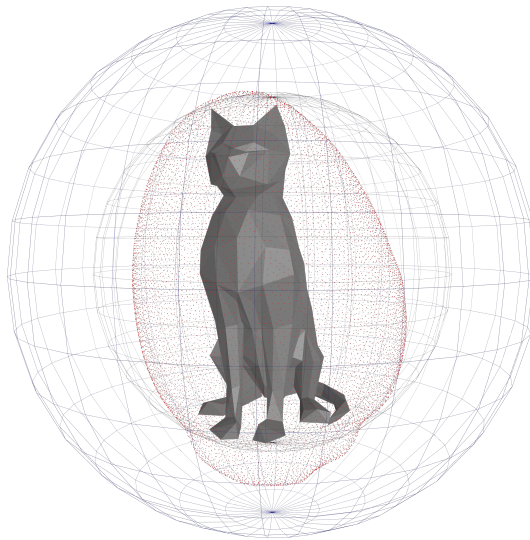
tozataik, ha a keresést azonos paraméterekkel indítjuk. A nyers erő módszerének esetében a modell körül bejárando tér nagyságát meghatározhatjuk az  $\mathcal{M}$  modellt befoglaló minimális gömb segítségével. Ennek, az  $\mathcal{M}$ -et tartalmazó gömbnek az izooptikus felülete is gömb, melynek sugarát egyszerűen meghatározhatjuk, hasonlóan a kétdimenziós kör izooptikus körének sugarát számító módszerhez:

$$R = r / \sin \frac{\alpha}{2}, \quad (2.9)$$

ahol  $r$  a befoglaló gömb sugara. Az  $R$  sugár a legnagyobb távolság, ahol az  $\mathcal{M}$  modell izooptikus pontjai lehetnek (14. ábra), mely a nyers erő módszerén kívül a gömbi keresés esetén is felhasználható a különböző irányokban történő keresés maximalizálására.

### 2.2.3.3. A szekvenciális és párhuzamos algoritmusok sebességének összehasonlítása

A 2. összehasonlító táblázat mutatja be a 2.2.1. fejezetben ismertetett szekvenciális és a GPU-n futó, párhuzamosított izooptikus felületet kereső algoritmusok különböző komplexitású modelleken végrehajtott futási idejét. A tesztekhez Intel Core i7-4790 processzort és GeForce GTX 1050 Ti grafikus kártyát, valamint CUDA 10.1-et használtunk. A számolásokhoz egyszeres, 32 bites pontosságot alkalmaztunk, a következő keresési paraméterekkel:  $\alpha = \frac{\pi}{2}$ , a bejárasi lépésköz 0.1, illetve egy  $P \in \mathbb{E}^3$  pontot akkor fogadtunk el, ha a kiszámított térköz és a megadott  $\alpha$  abszolút eltérése kevesebb volt mint  $2 \times 10^{-3}$  (hibahatár). A modelleket úgy méreteztük, hogy a befoglaló gömb sugara  $r = 5$  legyen, melyet a modell középpontja és a legtávolabbi csúcspont közötti euklideszi távolságként számoltunk ki. A keresésben tehát  $R = 5 / \sin \frac{\pi}{4}$  értéket használtunk. A keresési eredmények a 14. és 15. ábrán, továbbá a B. Függelékben található ábrákon láthatóak. Az előállított izooptikus poligonhálók a megtalált pontfelhőkből felület-rekonstrukciós algoritmus [27] segítségével készültek.



14. ábra. Macska modell izooptikus felületének pontjai (pirossal) a modellt befoglaló gömb izooptikusán belül vannak (modell forrás: [www.turbosquid.com](http://www.turbosquid.com)).

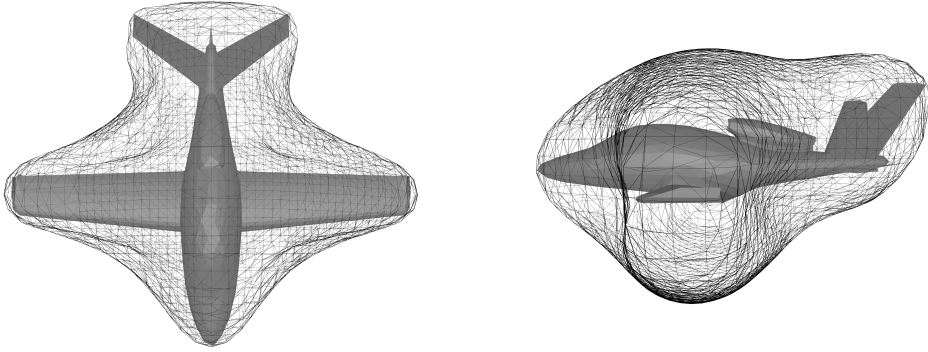
Ahogy az látható a futási idő alapvetően nő a megadott modellek bonyolultságával párhuzamosan. Az izooptikus felületek párhuzamosított algoritmussal történő keresése esetén megfigyelhető azonban, hogy az összetettebb, repülőgép modell izooptikusának előállítására gyorsabb, mint az annál kevesebb lapból és csúcspontból álló szarvas modell izooptikusának keresése. Ennek oka, hogy az  $\mathcal{S}$  levetített élék tömbjének várható méretét (`max_S` a 4. kódrészletben) a (2.8) egyenlet alapján becsüljük meg. Ez a számolás azon a valószínűségen alapul, hogy egy véletlenszerűen választott kamerapozícióból vizsgált élhez tartozó lapok normálvektorai mekkora eséllyel zárnak be egyszerre hegyes vagy tompaszöget a lapokból a kamerapontba mutató vektorral. Mivel a repülőgép modell szárnyait alkotó élék többségéhez tartozó lapok nor-

Modell	Nyers erő		Gömbi keresés	
	CPU	GPU	CPU	GPU
<b>Stanford Bunny</b> Csúcsok: 66, Lapok: 128	499.1	10.6	24.2	0.9
<b>Macska</b> Csúcsok: 216, Lapok: 428	2279.9	36.5	117.4	2.9
<b>Szarvas</b> Csúcsok: 376, Lapok: 747	4233.7	92.2	228.9	8.8
<b>Repülőgép</b> Csúcsok: 529, Lapok: 910	6449.4	63.8	418.3	5.1
<b>Elefánt</b> Csúcsok: 779, Lapok: 1492	9185.6	163.5	453.5	13.6

2. táblázat. Az izooptikus pontokat kereső szekvenciális és párhuzamos algoritmusok futási ideje (másodpercben). Paraméterek:  $\alpha = \frac{\pi}{2}$ , bejárési lépésköz = 0.1, hibahatár =  $2 \times 10^{-3}$ .

málvektorai párhuzamosak (15. ábra), ezért ez a valószínűségi érték kisebb, mint a szarvas modell esetében. Alacsonyabb sziluettélek száma esetén az algoritmus több háromdimenziós pontot vizsgálhat egyidejűleg, ezért a repülőgép modell izooptikus felületének keresése kevesebb időt vesz igénybe.

Amint a 2. táblázatból látható az új, GPU-ra tervezett párhuzamos algoritmusok segítségével történő izooptikus felületek keresése jelentősen felgyorsítható. Összetett háromdimenziós modellek esetén a keresési módszerek párhuzamosítási lehetősége azonban korlátozott, mivel a térszög kiszámítása több GPU erőforrást (memóriaterületet és szálakat) használ fel, tehát az eljárások csak kevesebb pontot tudnak egyszerre tesztelni. Ezért komplex (akár több ezer poligonból álló) modellek izooptikus felületének a parallel algoritmusok segítségével



15. ábra. A repülőgép modell és a pontfelhőből előállított izooptikus felülete ( $\alpha = \frac{\pi}{2}$ , modell forrás: [www.cadnav.com](http://www.cadnav.com)).

történő meghatározása és megjelenítése még mindig időigényes lehet. A sebesség növelésének egy egyszerű módja, ha az izooptikus pontokat az adott modell egyszerűsítése (decimálása) után keressük, mert a poligonháló lapszámának jelentős redukálása is csak minimális méretcsökkenést okoz a modell gömbi vetületében. A kiszámított térszög és az  $\alpha$  közötti elfogadási hibához képest elhanyagolható.

### 3. Log-aesthetic görbék

A geometriai modellezésben a tervezendő objektumoknak gyakran a funkcionális követelményeken felül bizonyos esztétikai elvárásoknak is meg kell felelniük. Mivel az utóbbi feltételeket kielégítő adatok nehezen írhatók le matematikai formulákkal, ezért általában az alakzatot jellemző belső geometriai tulajdonságok (érintő, görbület, torzió stb.) segítségével definiálják azokat.

A síkgörbék esztétika feltételekkel szembeni megfelelését leggyakrabban a görbület elemzésével vizsgálhatjuk. A görbület az egyenestől való eltérésnek a számértéke, azaz az egységnyi útra eső irányváltozás nagysága. Kiszámolható egy adott, kétszeresen folytonosan differenciálható görbe bármely pontjában. Egy görbét esztétikusnak nevezhetünk, ha a görbülete folytonos és csak néhány monoton görbületű részből áll [12]. Ilyen például az iparban (gépgyártás, építészet stb.) is gyakran használt természetes esztétikus görbe a parabola, az arkhimédészi spirál, a logaritmikus spirál vagy az Euler-féle spirál.

Érdeemes továbbá megemlíteni a Class A Bézier-görbék [11, 3, 63] is. A Class A elnevezés (A osztályú) az autóiparból származik, melyet elsősorban az esztétikai szempontból kritikus felületekre (motorház-tető, tető stb.) használnak. Ilyen magas minőségű, monoton görbületfüggvényű görbéket Bézier-görbékből, azok kontrollpoligonjára vonatkozó megszorításokkal lehet előállítani. Az ily módon előállított görbéket hívjuk Class A Bézier-görbéknek.

Emellett fontosak a Pitagorászi hodográf (PH) görbék [13, 14, 15] is. A hodográf egy síkgörbe első deriváltjából előállított görbe. Ha egy polinomiális görbe esetén létezik olyan  $\sigma(t)$  egész polinom, hogy  $\sigma^2(t) = x'^2(t) + y'^2(t)$ , akkor a görbe egy Pitagorászi hodográf görbe. Ez azért jelentős, mert a görbével kapcsolatos számolások egy része (pl. ívhossz, offszet görbe számolása) egyszerűsödik, illetve egzakt marad. Például racionális PH görbe offszetje racionális PH görbe.

Egy érdekes alternatívája a görbék esztétikai vizsgálatának a logaritmikus görbület tanulmányozása [54]. Ez az érték jól kiemeli a vizsgált görbe „egyenletesebb” részeit, mert ezen a kisebb eltérések is a görbe nagyobb változására utalhatnak (a logaritmus függvény használata miatt). Jelen fejezetben egy ilyen vizsgálat eredményeként előállított, az esztétikus modellezésben a legígéretesebbnek tartott [31] log-aesthetic görbével foglalkozunk, melynek a legfontosabb tulajdonságait és egyenleteit a 3.1.1. és a 3.1.2. alfejezetben ismertetjük.

Mivel a geometriai modellezésben gyakrabban használt Bézier, B-spline, illetve NURBS görbék esztétikai jellemzői nem mindig teljesítik a magas ipari elvárásokat, ezért gyakori, hogy az elkészült modell geometriai tulajdonságait valamilyen utófeldolgozással javítják az adott feltételeknek megfelelően (például [53]). Ezzel szemben egy esztétikus görbével történő modellezés hatékonyabb lehet, mert a segítségükkel készített alakzatok könnyebben megfelelhettek a magas ipari követelményeknek is. A 3.1.3. alfejezetben egy ilyen, a Yoshida és Saito által felírt interaktív log-aesthetic görbemodellező eljárást [64] mutatunk be.

A fejezet további részeiben az új eredményeinket ismertetjük. A kutatásunk első szakaszában a log-aesthetic görbe autoizooptikus tulajdonságát vizsgáltuk meg, tehát azt, hogy az izooptikus görbéje is log-aesthetic görbe-e. Ezen eredményeket a 3.2.1. alfejezetben ismertetjük. Emellett továbbfejlesztettük Yoshida és Saito algoritmusát, bővítve a módszerük felhasználási lehetőségét ( $G^1$ , illetve  $G^2$  Hermite-interpolációs problémára) és enyhítve az eljárás geometriai adatainak megadására vonatkozó szigorú feltételeket, a könnyebb felhasználás érdekében. Az új módszert úgynevezett kiterjesztett (két,  $G^2$  folytonosan csatlakoztatott) log-aesthetic görbékre terveztük meg, melyek számolását és az új modellező algoritmus főbb lépéseit a 3.2.2. alfejezetben ismertetjük.

Habár a Yoshida és Saito-féle eljárás meghatározó technikának számít a szakirodalomban, számos további log-aesthetic görbemodellező

eljárást találhatunk, melyek közül kiemelnénk Miura és társai munkáit, melyek között találunk olyan alkalmazást, mely kétdimenziós pontsorozatra interpolál egy inflexiós pontot is tartalmazható görbeszeletet [41], illetve egy újabb megoldásukban [40] a felhasználók a log-aesthetic görbe végpontjainak érintővektorát is szabályozhatják, hasonlóan az általunk fejlesztett eljáráshoz, azzal a különbséggel, hogy Miura és munkatársai az interpolációt három összekapcsolt log-aesthetic görbével (triplets) oldották meg.

## 3.1. Előzmények

### 3.1.1. Log-aesthetic görbe származtatása

A log-aesthetic görbe meghatározása Harada és társai [22, 21] nevéhez fűződik. A szerzők elemezték a természetes esztétikus görbék jellemzőit és felismerték, hogy ezen görbék olyan tulajdonsággal rendelkeznek, hogy a görbületük logaritmikus eloszlási diagramját (logarithmic distribution diagram of curvature – LDDC) egy egyenes vonallal lehet közelíteni. Megállapították, hogy szoros összefüggés van ezen vonal meredeksége és a görbék benyomása, látványa között. Ezzel egy közös, esztétikai minőséggel összefüggő, mérhető értéket rendeltek a különböző görbékhez.

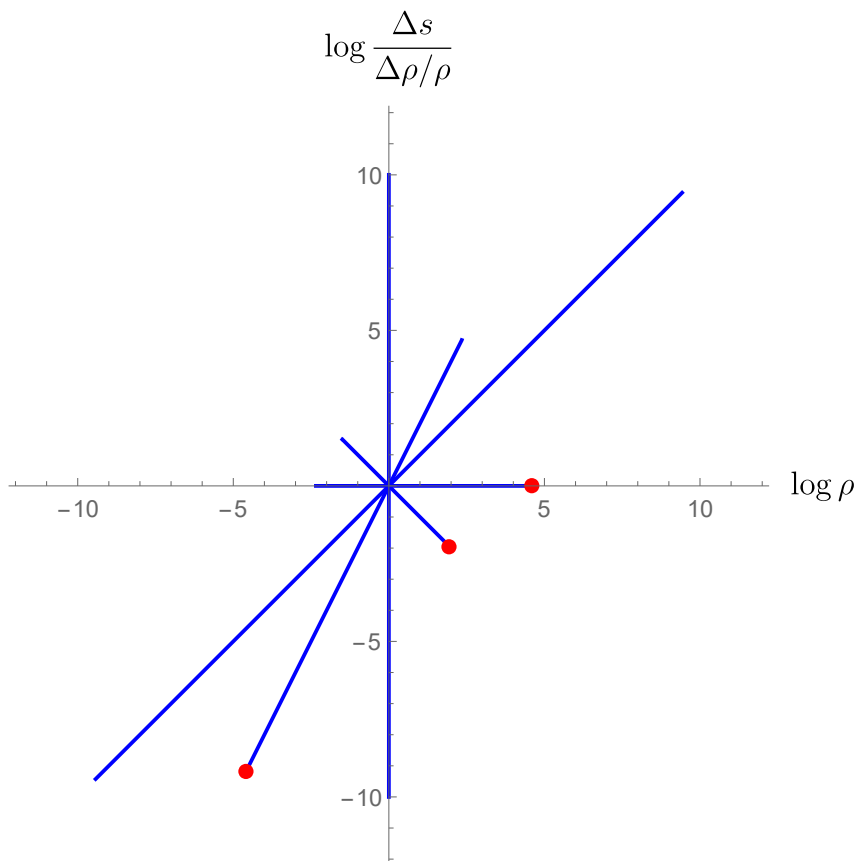
Az említett munka alapján Miura és társai definiálták [39, 38] az LDDC analitikus változatát, az úgynevezett logaritmikus görbületi grafikont (logarithmic curvature graph – LCG) a következőképpen: ha egy  $\rho(s)$  görbületi sugárral adott és az  $s$  húrhosszal paraméterezett görbe paramétertartományát elengedően sűrűn felosztjuk úgy, hogy a  $\Delta\rho/\rho$  érték konstansnak tekinthető legyen, az LCG a  $\rho$  és a  $\Delta s$  kapcsolatát írja le egy kettős logaritmikus grafikonon (16. ábra). Meghatározták továbbá az egyenletét azon görbéknek, melyek egy  $\alpha$  meredekségű

egyeneset határoznak meg ezen a grafikonon:

$$\log \rho \frac{ds}{d\rho} = \alpha \log \rho + c, \quad (3.1)$$

ahol a  $c$  konstans érték. A (3.1) a log-aesthetic görbe alapegyenletének tekinthető.

Az LCG-n eltérő meredekségű egyenesek más-más típusú görbét definiálnak, ezért a log-aesthetic görbe értelmezhető úgy is, mint a különböző  $\alpha$  alakparaméterekkel meghatározott görbecsalád (16. ábra).



16. ábra. Különböző alakparaméterű log-aesthetic görbék logaritmikus görbületi grafikonjai: Euler-féle spirál ( $\alpha = -1$ ), Nielsen spirál ( $\alpha = 0$ ), logaritmikus spirál ( $\alpha = 1$ ), körevolvens ( $\alpha = 2$ ), kör ( $\alpha = \pm\infty$ ). A görbék azonos  $\theta$  intervallumokon láthatóak, melyek az  $\alpha = 1$  és  $\alpha = \pm\infty$  esetek kivételével alulról korlátosak (piros pontok).

### 3.1.2. Log-aesthetic görbe egyenletei

Yoshida és Saito [64] az említett modellező eljárás mellett megvizsgálták a log-aesthetic görbe tulajdonságait és az ívhossz és a görbületi sugár közötti összefüggésből általános, a görbe generálásához (megrajzolásához) használható paraméteres egyenleteket vezettek le. A következőkben az ő munkájuk eredményeit összegezzük.

A standard egyenleteket egy origóban elhelyezett referenciapont segítségével vezették le a következő feltételek mellett: a görbületi sugár értéke 1 és az érintővektor az  $X$  koordinátatengely mentén pozitív irányú a referencia pontban. A formulákat a (3.1) egyenlet átalakításával, a log-aesthetic görbe transzformálásával kapjuk úgy, hogy az említett feltételek teljesüljenek, ahol egy  $\Lambda$  érték (paraméter) felelős a megfelelő transzformációkért. A szerzők munkája [64] alapján egy  $P(\theta)$  görbepont koordinátáit, melynek irányyszöge  $\theta$  (azaz a pontban vett érintőegyenest és az  $X$  tengely hajlásszöge) a következőképpen számolhatjuk ki a komplex síkon:

$$P(\theta) = \begin{cases} \int_0^\theta e^{(1+i)\Lambda\psi} d\psi & \text{ha } \alpha = 1 \\ \int_0^\theta ((\alpha - 1)\Lambda\psi + 1)^{\frac{1}{\alpha-1}} e^{i\psi} d\psi & \text{egyébként,} \end{cases} \quad (3.2)$$

ahol  $\alpha \in \mathbb{R}$  és  $\Lambda \in \mathbb{R}^+$  paraméterek. A  $\theta = 0$  pont az origóban elhelyezkedő referenciapont, melynek irányvektora  $[1 \ 0]^T$ .  $\Lambda$  a (3.1) egyenlet konstans részéből adódik ( $c = -\log \Lambda$ ). Megegyező  $\alpha (\neq 1)$ , de különböző  $\Lambda_1$  és  $\Lambda_2$  paraméterrel adott  $C_1$  és  $C_2$  log-aesthetic görbe esetén létezik olyan hasonlósági transzformáció, mellyel  $C_1$  előállítható  $C_2$ -ből.

A görbületi sugár a következő egyenlettel határozható meg ([64]):

$$\rho(\theta) = \begin{cases} e^{\Lambda\theta} & \text{ha } \alpha = 1 \\ ((\alpha - 1)\Lambda\theta + 1)^{\frac{1}{\alpha-1}} & \text{egyébként.} \end{cases} \quad (3.3)$$

Ha  $\theta = 0$ , akkor  $\rho = 1$ , illetve a függvény monoton növekvő  $\Lambda \neq 0$  esetén. Ha  $\Lambda = 0$ ,  $\rho$  konstans 1 és a log-aesthetic görbe egy kör. Hasonlóan, amikor  $\alpha = \pm\infty$ , mert  $\rho$  tart 1-hez, ha  $\alpha$  tart a  $\pm$  végtelenbe.

A görbe  $\theta$  irányszöge és  $s$  húrhossza között a következő összefüggés érvényes ([64]):

$$\theta(s) = \begin{cases} \frac{1-e^{-\Lambda s}}{\Lambda} & \text{ha } \alpha = 0 \\ \frac{\log(\Lambda s+1)}{\Lambda} & \text{ha } \alpha = 1 \\ \frac{(\Lambda\alpha s+1)^{(1-\frac{1}{\alpha})}-1}{\Lambda(\alpha-1)} & \text{egyébként.} \end{cases} \quad (3.4)$$

Ezért a log-aesthetic görbe pontjainak koordinátái kifejezhetők a komplex síkon az  $s$  húrhossz függvényében is a következő egyenlettel ([64]):

$$C(s) = \begin{cases} \int_0^s \exp(i \frac{1-e^{-\Lambda u}}{\Lambda}) du & \text{ha } \alpha = 0 \\ \int_0^s \exp(i \frac{\log(\Lambda u+1)}{\Lambda}) du & \text{ha } \alpha = 1 \\ \int_0^s \exp(i \frac{(\Lambda\alpha u+1)^{(1-\frac{1}{\alpha})}-1}{\Lambda(\alpha-1)}) du & \text{egyébként.} \end{cases} \quad (3.5)$$

A (3.5) és a (3.2) egyenlet megegyező  $\alpha$  és  $\Lambda$  paraméterekkel ugyanazt a görbét határozza meg.

A (3.4) egyenletet felhasználva a görbületi sugár is kifejezhető a húrhossz függvényében ([64]):

$$\rho(s) = \begin{cases} e^{\Lambda s} & \text{ha } \alpha = 0 \\ (\Lambda\alpha s + 1)^{\frac{1}{\alpha}} & \text{egyébként.} \end{cases} \quad (3.6)$$

Mivel a  $\rho$  a  $[-\infty, +\infty]$  intervallumon értelmezett, a  $\theta$  irányszögnek és az  $s$  húrhossznak lehet alsó, illetve felső korlátja az  $\alpha$  értékétől függően (a felírt egyenletekben előforduló negatív alapú törthatványok miatt):

	Irányszög ( $\theta$ )			Húrhossz ( $s$ )		
	$\alpha < 1$	$\alpha = 1$	$\alpha > 1$	$\alpha < 0$	$\alpha = 0$	$\alpha > 0$
Felső korlát:	$\frac{1}{\Lambda(1-\alpha)}$	-	-	$-\frac{1}{\Lambda\alpha}$	-	-
Alsó korlát:	-	-	$\frac{1}{\Lambda(1-\alpha)}$	-	-	$-\frac{1}{\Lambda\alpha}$

3. táblázat. A  $\theta$  irányszög és az  $s$  húrhossz alsó, illetve felső korlátja.

Megemlítendő, hogy a felírt egyenletek  $\alpha \approx 0$  és  $\alpha \approx 1$ , de  $\alpha \neq 0$  és  $\alpha \neq 1$  esetben instabilak lehetnek [37]. Az algoritmusok során ezeket az  $\alpha$  értékeket érdemes elkerülnünk ( $]0, 0 \pm \epsilon]$ , illetve  $]1, 1 \pm \epsilon]$ ), mely módosítás csak elhanyagolható eltérést okoz a bemutatásra kerülő algoritmusok alapján meghatározott log-aesthetic görbe geometriai tulajdonságaira vonatkozóan.

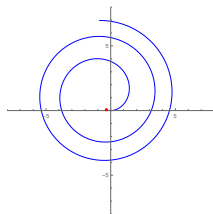
Az említett szerzők [64], a  $\Lambda = 1$  esetet a görbe Standard I formájának nevezik. Nézzük meg a görbét ebben az esetben a különböző  $\alpha$  értékek függvényében:

**$\alpha > 1$ :** A görbe alulról korlátos. Ahogy a  $\theta$  irányszög a végtelenbe tart, úgy a görbe spirálisan divergál a  $\rho = \infty$  pontja felé (ahol a húrhossz is végtelen, 17.(a)–(d) ábra).

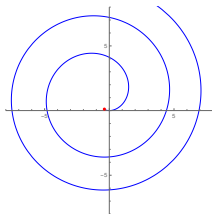
**$\alpha < 1$ :** A görbe felülről korlátos. Ahogy a  $\theta$  irányszög a mínusz végtelenbe tart, úgy a görbe spirálisan konvergál a  $\rho = 0$  pontja felé (ahol a húrhossz is végtelen, 17.(f)–(l) ábra).

**$\alpha = 1$ :** Ahogy a  $\theta$  irányszög a mínusz végtelenbe tart, úgy a görbe spirálisan konvergál a  $\rho = 0$  pontja felé (ahol a húrhossz véges, az alsó korlát:  $-\frac{1}{\Lambda\alpha}$ ). Ahogy a  $\theta$  a végtelenbe tart, úgy a görbe spirálisan divergál a  $\rho = \infty$  pontja felé (ahol a húrhossz is végtelen, 17.(e) ábra).

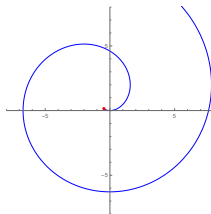
**$\alpha = \pm\infty$ :** A log-aesthetic görbe egy  $[0 \ 1]^T$  középpontú egységkör.



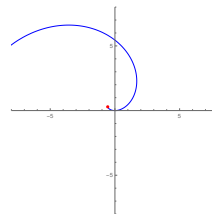
(a)  $\alpha = 3$



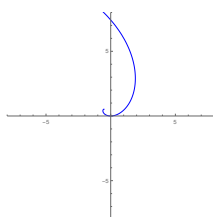
(b)  $\alpha = 2.5$



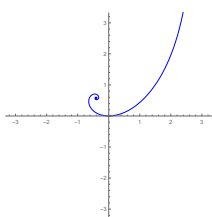
(c)  $\alpha = 2$ ,  
kőrevolvens



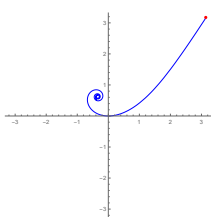
(d)  $\alpha = 1.5$



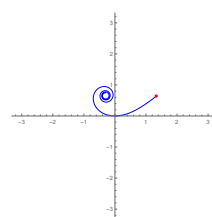
(e)  $\alpha = 1$ ,  
logaritmicus  
spirál



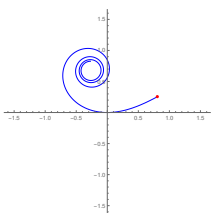
(f)  $\alpha = 0.5$



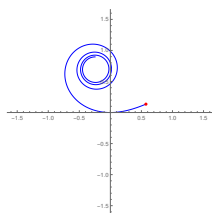
(g)  $\alpha = 0$ ,  
Nielsen spirál



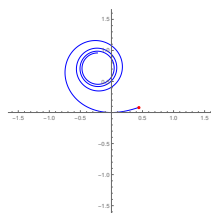
(h)  $\alpha = -0.5$



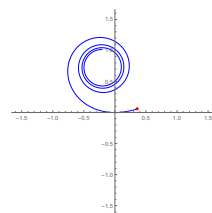
(i)  $\alpha = -1$ ,  
Euler-féle spirál



(j)  $\alpha = -1.5$



(k)  $\alpha = -2$



(l)  $\alpha = -2.5$

17. ábra. Log-aesthetic görbék különböző  $\alpha$  alak- és  $\Lambda = 1$  paraméterrel. A piros pontok a  $\theta$  paraméter korlátjai (3. táblázat).

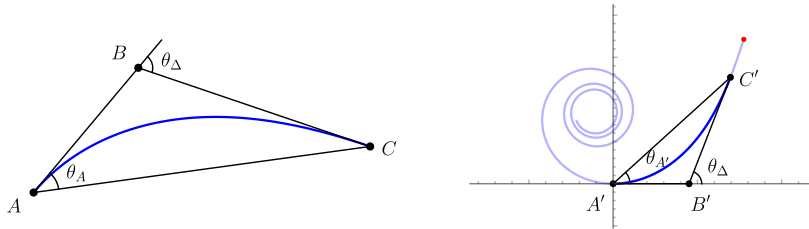
### 3.1.3. Interaktív modellező algoritmus

Yoshida és Saito a log-aesthetic görbe egyenleteinek levezetésén túl kifejlesztett egy interaktív interpolációs eljárást [64] is, melynek főbb lépéseit jelen fejezetben ismertetjük. A megoldásuk az esztétikus görbemodellezésben meghatározó technikának számít és számos új algoritmust fejlesztettek ki az általuk felírt módszer alapján.

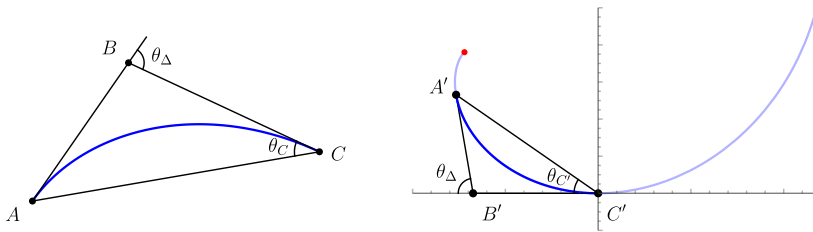
Az algoritmus megadott  $\alpha$  mellett három kontrollpontra interpolál egy log-aesthetic görbeszegmenst. Az eljárás egy megfelelő görberészt keres, amely illeszkedik az  $A$ ,  $B$ ,  $C$  kontrollpontok által meghatározott háromszögbe, felezőmódszert alkalmazva a  $\Lambda$  paraméterre. A görbeszegmenst az  $A$ -tól a  $C$  pontig rajzoljuk, míg a  $B$  pont a végpontok közötti  $\theta_\Delta$  irányszög eltérését határozza meg. Ha  $|AB| \leq |BC|$  nem teljesül a végpontok koordinátáit felcseréljük.

Az algoritmus a 18. ábrán látható módon a következőképpen működik. Az  $A$ -nak megfelelő  $A'$  pontot a log-aesthetic görbe azon pontjaként definiáljuk, amelynek irányszöge 0 (referenciapont). A  $C'$  pont felel meg a  $C$ -nek, ahol az irányszög  $\theta_\Delta$ . A  $B'$  a végpontok érintőegyeneseinek a metszéspontja, amely az  $X$  tengelyen helyezkedik el, hiszen az  $A'$  érintővektora  $[1 \ 0]^T$ . Ha az  $A'B'C'$  háromszög hasonló az  $ABC$  háromszöghöz a görbeszegmens megrajzolható úgy, hogy az  $A'B'C'$ -beli log-aesthetic pontokat transzformáljuk az  $ABC$  háromszögbe. Az algoritmusban az optimális  $\Lambda$  megtalálásához a háromszögek hasonlóságát vizsgáljuk a megfelelő szögek összehasonlításával: ha  $\alpha \leq 1$ , a  $\theta_A$  és  $\theta_{A'}$ , egyébként a  $\theta_B$  és  $\theta_{B'}$  egyezését ellenőrizzük. Egyenlőség esetén a háromszögek hasonlóak, mert a külső szög a  $B$  és  $B'$  pontban egyaránt  $\theta_\Delta$ .

Ha  $\alpha < 1$  az irányszög a  $[0, \theta_\Delta]$  intervallumban van. Mivel ebben az esetben  $\theta$  felső korlátos, ezért  $\theta_\Delta < 1/(\Lambda(1 - \alpha))$ . Ez átalakítva a következő:  $0 < \Lambda < 1/(\theta_\Delta(1 - \alpha))$ . Ezzel szemben, amikor  $\alpha > 1$ ,  $A'$  az a pont, ahol az irányszög  $-\theta_\Delta$  és  $C'$ -ben pedig 0. Az intervallum tehát  $[-\theta_\Delta, 0]$  és  $\theta$  alsó korlátos, ezért  $-\theta_\Delta > 1/(\Lambda(1 - \alpha))$ . Ebben az



(a)  $\alpha \leq 1$



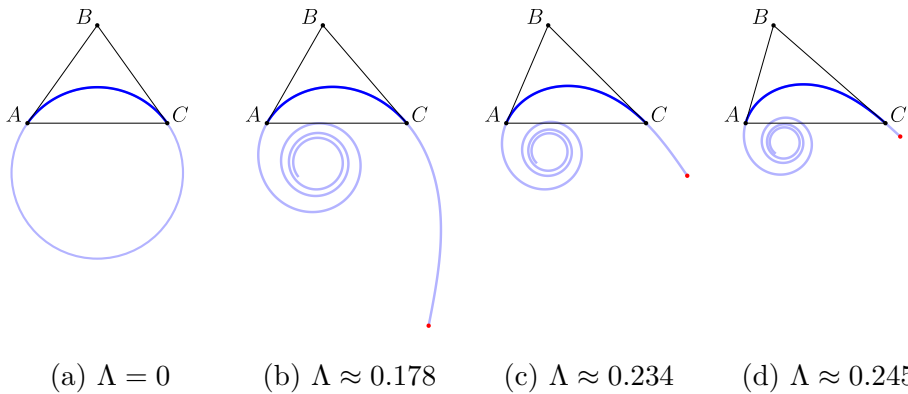
(b)  $\alpha > 1$

18. ábra. Yoshida és Saito algoritmusának működése, a [64] 7. ábrája alapján (a piros pont a log-aesthetic görbe korlátja).

esetben tehát a  $\Lambda$ -t a 0 és  $1/(\theta_{\Delta}(\alpha - 1))$  között keressük. Amikor  $\alpha = 1$ ,  $\theta$  nem korlátos, ezért a felező módszer során  $\Lambda (> 0)$  tetszőlegesen nagy értéket felvehet. A Yoshida és Saito által felírt felező módszer pszeudokódját a [64] A. Függelékében találjuk.

Az algoritmusuk hátránya, hogy a kontrollpontok helyzete és az  $\alpha$  értéke erősen befolyásolja, illetve korlátozza a görbe megrajzolását. A bemutatott eljárással a (húrhossz szerint) legnagyobb log-aesthetic görbeszelet a referenciaponttól a 18. ábrán pirossal látható görbe végpontjáig rajzolható meg, ahol a  $\theta$ , illetve  $s$  korlátja van. Ha a kontrollpontok egy egyenlő szárú háromszöget képeznek úgy, hogy az  $AB$  és  $BC$  oldalak egyenlő hosszúságúak, a log-aesthetic görbeszelet egy körív, azaz  $\Lambda = 0$  (19. ábra). Ha a  $B$  pontot az  $AC$  oldallal párhuzam-

mosan, annak egy végpontja felé mozgatjuk, akkor az a végpont lesz az első pontja a görberésznek, azaz a referenciapont (az említett koordinátacsere miatt). Egy ilyen irányú mozgatással a  $\Lambda$  értéke nő, illetve a másik végpont megközelíti a log-aesthetic görbe határpontját. A görbeszegmens addig határozható meg, amíg ez a másik végpont el nem éri a korlátot, azaz amíg a  $\Lambda$  az előző bekezdésben leírt intervallumon belül van.



19. ábra. A  $B$  pont  $AC$  oldallal párhuzamosan történő mozgása,  $\alpha = -1$  esetén. További módosítás esetén a pirossal jelölt határpont a háromszögön belülré esik és az algoritmusnak nincs megoldása.

Ha  $\alpha$  kisebb, mint 0 vagy nagyobb, mint 1, akkor ez az úgynevezett megrajzolható régió [64, 65, 17] drasztikusan csökken (ahogyan például a 9. ábrán látható a [64]-ben).

## 3.2. Új eredményeink

### 3.2.1. Log-aesthetic görbe és izooptikusának kapcsolata

Jelen alfejezetben a log-aesthetic görbe autoizooptikus tulajdonságát tanulmányozzuk, azaz megvizsgáljuk, hogy a különböző  $\alpha$  alakparaméterrel megadott log-aesthetic görbék izooptikus görbéje log-aesthetic görbe-e. Mivel különböző  $\Lambda$  paraméterrel megadott görbék-ből megfelelő hasonlósági transzformációval kongruens görbe állítható elő, ezért a következőkben külön csak a  $\Lambda = 1$  esettel foglalkozunk.

Az irányszög szerint paraméterezett log-aesthetic görbék ((3.2) egyenlet) izooptikus görbéje az 1. definíció alapján felírható összefüggésből adódóan egyszerűen meghatározható. Megadott  $\gamma$  szögű izooptikus esetén a megfelelő érintőegyeneselek egyenletének felírása a következőképpen adódik: egy tetszőleges  $P(\theta)$  görbepontból húzott érintő a  $P(\theta + (\pi - \gamma))$  ponthoz tartozó érintőegyenessel zár be  $\gamma$  szöget. Továbbá egy  $P(\theta)$  log-aesthetic pont egység hosszú tangensvektora:  $[\cos \theta \ \sin \theta]^T$ . Ezen összefüggés bármely irányszög szerint paraméterezett  $r(t)$  síkgörbére igaz [1]:

$$\frac{r'(t)}{|r'(t)|} = \begin{bmatrix} \cos t \\ \sin t \end{bmatrix}. \quad (3.7)$$

Az egyszerűség kedvéért definiáljuk az izooptikust a  $\delta = \pi - \gamma$  szög segítségével. Így a megfelelő görbepontokból húzott érintőegyeneselek metszéspontját a következő egyenlet segítségével számolhatjuk ki:

$$P(\theta) + t_\theta \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} = P(\theta + \delta) + t_{\theta+\delta} \begin{bmatrix} \cos(\theta + \delta) \\ \sin(\theta + \delta) \end{bmatrix}. \quad (3.8)$$

Az egyenletrendszer megoldva  $t_\theta$ -ra a következő összefüggést kapjuk:

$$t_\theta = \csc(\delta) \left( \vec{V}_x^\delta(\theta) \sin(\theta + \delta) - \vec{V}_y^\delta(\theta) \cos(\theta + \delta) \right), \quad (3.9)$$

ahol  $\vec{V}_x^\delta(\theta)$  és  $\vec{V}_y^\delta(\theta)$  a  $\vec{V}^\delta(\theta) = \overline{P(\theta)P(\theta + \delta)}$  vektor koordinátái, melynek explicit egyenlete a következőképpen adható meg a komplex síkon (a (3.2) összefüggésből):

$$\vec{V}^\delta(\theta) = \begin{cases} \int_{\theta}^{\theta+\delta} e^{(1+i)\Lambda\psi} d\psi & \text{ha } \alpha = 1 \\ \int_{\theta}^{\theta+\delta} ((\alpha - 1)\Lambda\psi + 1)^{\frac{1}{\alpha-1}} e^{i\psi} d\psi & \text{különben.} \end{cases} \quad (3.10)$$

Tehát a log-aesthetic görbe (és bármely irányszög szerint parameterezett síkgörbe)  $\delta (= \pi - \gamma)$  szöggel megadott izooptikus görbéje a következőképpen írható fel:

$$I_\delta(\theta) = P(\theta) + \csc(\delta) \left( \vec{V}_x^\delta(\theta) \sin(\theta + \delta) - \vec{V}_y^\delta(\theta) \cos(\theta + \delta) \right) \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \quad (3.11)$$

A következő módosítások segítségével az izooptikus geometriai összefüggései jobban szemléltethetőek. A (3.9) egyenlet átalakítható a harmonikus összeadás tételének segítségével [58], a következőképpen:

$$t_\theta = \csc(\delta) \operatorname{sign} \left( \vec{V}_x^\delta(\theta) \right) \left\| \vec{V}^\delta(\theta) \right\| \sin \left( \theta + \delta + \arctan \left( \frac{-\vec{V}_y^\delta(\theta)}{\vec{V}_x^\delta(\theta)} \right) \right). \quad (3.12)$$

Továbbá az arctan átírható a jól ismert arctan2 függvényre, mely megadja egy tetszőleges  $\vec{V}$  vektor irányszögét (azaz meghatározza a pozitív  $X$  tengely és a  $[\vec{V}_x \ \vec{V}_y]^T$  pontba mutató vektor szögét a  $[0, \pi]$  intervallumon), ami által a  $\operatorname{sign} \left( \vec{V}_x^\delta(\theta) \right)$  az egyenletből kihagyható:

$$t_\theta = \csc(\delta) \left\| \vec{V}^\delta(\theta) \right\| \sin \left( \theta + \delta - \operatorname{arctan2} \left( \vec{V}_y^\delta(\theta), \vec{V}_x^\delta(\theta) \right) \right). \quad (3.13)$$

Ez azt jelenti, hogy az irányszög szerint paraméterezett görbék izooptikus görbéje megadható a következőképpen is:

$$I_\delta(\theta) = P(\theta) + \csc(\delta) \|\vec{V}^\delta(\theta)\| \sin\left(\theta + \delta - \arctan2\left(\vec{V}_y^\delta(\theta), \vec{V}_x^\delta(\theta)\right)\right) \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}. \quad (3.14)$$

Az autoizooptikus tulajdonság vizsgálatához írjuk fel a log-aesthetic görbe izooptikusának LCG-jét. Tekintsük először az irányszög szerint paraméterezett log-aesthetic görbe LCG függvényét:

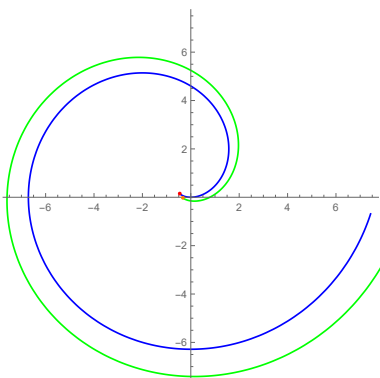
$$P_{LCG}(\theta) = \begin{bmatrix} \log \rho(\theta) \\ \log \frac{\rho(\theta)}{|\rho(s)|} \end{bmatrix}, \quad (3.15)$$

mely a megfelelő görbületi sugár képletek behelyettesítése, illetve az egyszerűsítések után a következő:

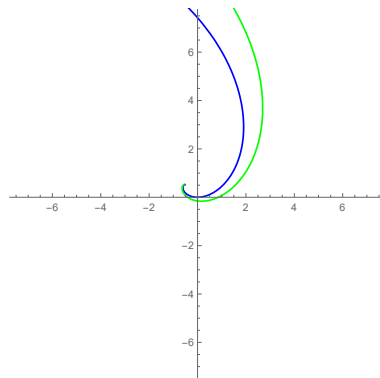
$$P_{LCG}(\theta) = \begin{cases} \begin{bmatrix} \Lambda \\ \theta\Lambda - \log(\Lambda) \end{bmatrix} & \alpha = 1 \\ \begin{bmatrix} \frac{\log((\alpha-1)\theta\Lambda+1)}{\alpha-1} \\ \frac{\alpha \log((\alpha-1)\theta\Lambda+1)}{\alpha-1} - \log(\Lambda) \end{bmatrix} & \text{különben.} \end{cases} \quad (3.16)$$

Ezen összefüggés az LCG-n (melynek tengelyei  $\log \rho$  és  $\log \frac{\Delta s}{\Delta \rho / \rho}$ ) egy  $\alpha$  meredekségű egyenest ír le (16. ábra). A meredekséget meghatározhatjuk a következő összefüggés kiszámításával is (figyelembe véve a 3. táblázat  $\theta$  korlátait):

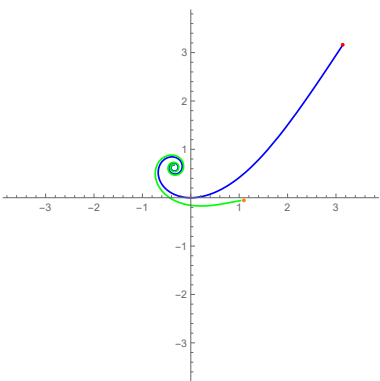
$$\hat{\alpha}(\theta) = \begin{cases} \frac{P_{LCG}(\theta-\phi)_y - P_{LCG}(\theta)_y}{P_{LCG}(\theta-\phi)_x - P_{LCG}(\theta)_x} & \alpha < 1 \\ \frac{P_{LCG}(\theta+\phi)_y - P_{LCG}(\theta)_y}{P_{LCG}(\theta+\phi)_x - P_{LCG}(\theta)_x} & \alpha \geq 1, \end{cases} \quad (3.17)$$



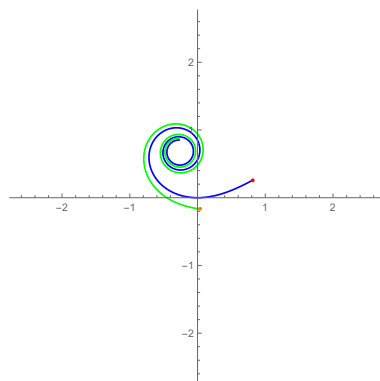
(a)  $\alpha = 2, \Lambda = 1$   
körevolvens



(b)  $\alpha = 1, \Lambda = 1$   
logaritmicus spirál



(c)  $\alpha = 0, \Lambda = 1$   
Nielsen spirál



(d)  $\alpha = -1, \Lambda = 1$   
Euler-féle spirál

20. ábra. Log-aesthetic görbék (kékkel) és azok  $\delta = \frac{\pi}{3}$  szöggel megadott izoptikus görbéi (zölddel). A piros, illetve narancssárga pontok a görbék korlátjai.

ahol  $\phi \in \mathbb{R} \setminus 0$ , illetve a  $P_{LCG}(\theta)_x$  és  $P_{LCG}(\theta)_y$  a  $\theta$  irányszög által meghatározott LCG pont koordinátái. Ha minden  $\alpha \in \mathbb{R}$  alakparaméterű log-aesthetic görbe izooptikusának LCG-je esetén ezen meredekség konstans lenne, akkor az a log-aesthetic görbe autoizooptikus tulajdonságát igazolná.

Sajnos a (3.11) és (3.14) egyenlettel felírt izooptikus görbe LCG függvényének a (3.16) egyenlethez hasonló egzakt meghatározása komputeralgebrai szoftverekkel is csak bizonyos  $\alpha$  értékekre lehetséges, amikor a log-aesthetic görbe irányszög szerint paraméterezett (3.2) egyenlete felírható trigonometrikus függvények segítségével, azaz [66] alapján, amikor  $\alpha = 2, \frac{3}{2}, \frac{4}{3}, \dots, \frac{k+1}{k}$ ,  $k \in \mathbb{N}^*$ , illetve amikor felírható Fresnel-integrálok segítségével,  $\alpha = -1, 3$  esetben. Továbbá klasszikus geometriai összefüggések alapján egyszerűen belátható, hogy a kör izooptikus görbéje is kör ( $\alpha = \pm\infty$ ), mely esetben az LCG vizsgálat nem indokolt, a log-aesthetic görbe autoizooptikus.

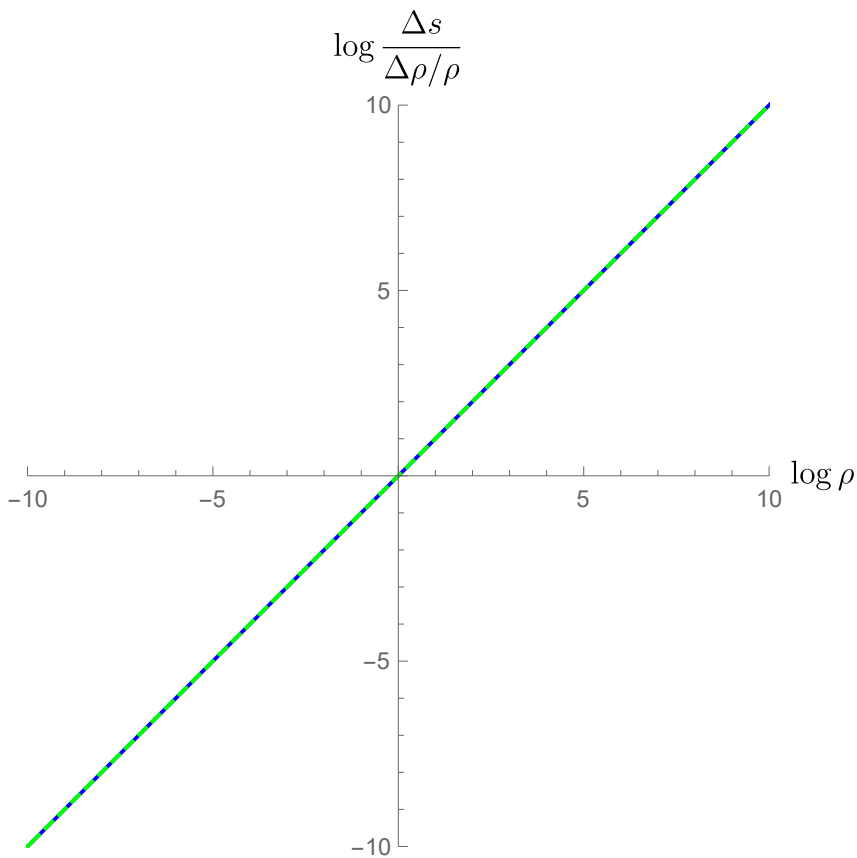
Tekintsünk egy másik, a Romero és társai [51] által bizonyított autoizooptikus esetet, amikor az  $\alpha = 1$ , azaz a görbe a logaritmikus spirál (20.(b) ábra). Ebben az esetben az LCG függvény a következő:

$$P_{LCG}^{\alpha=1}(\theta) = \begin{bmatrix} \theta\Lambda \\ \theta\Lambda - \log(\Lambda) \end{bmatrix}, \quad (3.18)$$

illetve az izooptikus görbe LCG formulája is megadható:

$$I_{LCG}^{\alpha=1}(\theta) = \begin{bmatrix} \frac{1}{2} \log \left( \frac{e^{2\delta\Lambda} - 2e^{\delta\Lambda} \cos(\delta) + 1}{\Lambda^2 + 1} \right) + \log(\csc(\delta)) + \theta\Lambda \\ \log \left( \frac{\csc(\delta) \sqrt{\frac{e^{2\delta\Lambda} - 2e^{\delta\Lambda} \cos(\delta) + 1}{\Lambda^2 + 1}}}{\Lambda} \right) + \theta\Lambda \end{bmatrix}. \quad (3.19)$$

Az izooptikus LCG  $\hat{\alpha}(\theta)$  meredeksége a (3.17) egyenlethez hasonlóan kiszámolható és belátható, hogy konstans 1, bármely  $\theta \in \mathbb{R}$  esetben,



21. ábra.  $\alpha = 1$  és  $\Lambda = 1$  paraméterrel megadott log-aesthetic görbe (kékkel) és annak  $\delta = \frac{2}{3}\pi$  szöggel számolt izooptikus görbéjének (szaggatott zöld vonallal rajzolt) logaritmikus görbületi grafikonja.

azaz az izooptikus is egy  $\alpha = 1$  alakparaméterű log-aesthetic görbe (21. ábra).

Egy másik példaként tekintsük az  $\alpha = 2$  esetet, amikor a log-

aesthetic görbe a körevolvens (20.(a) ábra), ekkor:

$$P_{LCG}^{\alpha=2}(\theta) = \left[ \begin{array}{c} \log(\theta\Lambda + 1) \\ 2\log(\theta\Lambda + 1) - \log(\Lambda) \end{array} \right], \quad (3.20)$$

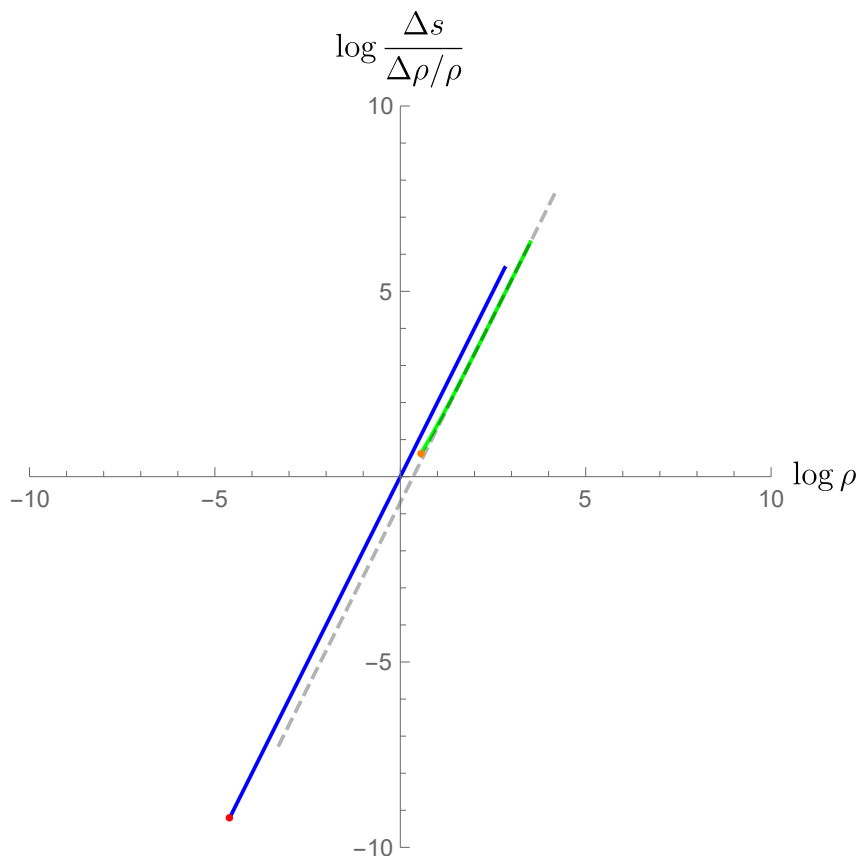
illetve a izooptikus LCG függvénye, megadott  $\delta = \frac{2\pi}{3}$  és  $\Lambda = 1$  paraméterrel (azaz nem tetszőleges  $\Lambda > 0$  és  $0 < \delta < \pi$  formula, a túl hosszú képlet kiírásának elkerülése érdekében):

$$I_{LCG, \delta=2\pi/3}^{\alpha=2, \Lambda=1}(\theta) = \left[ \begin{array}{c} \log \left( \frac{2(-6\pi(-3\theta + \sqrt{3}-3) + 27\theta(\theta+2) + 4\pi^2 + 54)^{3/2}}{27\sqrt{3} | 3\theta(\theta+2) + \pi(2\theta - \sqrt{3} + 2) + \frac{4\pi^2}{9} + 9 |} \right) \\ \log \left( \frac{2(-6\pi(-3\theta + \sqrt{3}-3) + 27\theta(\theta+2) + 4\pi^2 + 54)^{3/2}}{27\sqrt{3}} + \right. \\ \left. + \frac{-9\pi(-2\theta + \sqrt{3}-2) + 27\theta(\theta+2) + 4\pi^2 + 81}{(3\theta + \pi + 3) \left( 3\pi(6\theta - 5\sqrt{3} + 6) + 27(\theta(\theta+2) + 5) + 4\pi^2 \right)} \right) \end{array} \right]. \quad (3.21)$$

A (3.17) egyenlethez hasonlóan számolva az  $\hat{\alpha}(\theta)$  értéket, különböző  $\theta$  (és  $\phi = \pi$ ) paraméterek esetén, különböző meredekséget kapunk:

$$\begin{aligned} \hat{\alpha}(-1) &\approx 1.86204 \quad (\theta \text{ alsó korlátja}) \\ \hat{\alpha}(0) &\approx 1.96486 \\ \hat{\alpha}(\pi) &\approx 1.99523 \\ \hat{\alpha}(3\pi) &\approx 1.99899 \\ \hat{\alpha}(10\pi) &\approx 1.99987 \\ \hat{\alpha}(100\pi) &\approx 1.99999845 \\ \lim_{\theta \rightarrow \infty} \hat{\alpha}(\theta) &= 2. \end{aligned} \quad (3.22)$$

Egyértelműen kijelenthetjük tehát, hogy az  $\alpha = 2$  paraméterrel megadott log-aesthetic görbe (azaz a körevolvens) izooptikus görbéje nem log-aesthetic görbe (22. ábra). Megjegyezzük azonban, hogy az  $\hat{\alpha}(\theta)$  értéke 2, ha  $\theta$ -val a végtelenbe tartunk.



22. ábra.  $\alpha = 2$  és  $\Lambda = 1$  paraméterrel megadott log-aesthetic görbe (kék), annak transzformáltja (szaggatott vonal) és a  $\delta = \frac{2}{3}\pi$  szöggel számolt izooptikus görbe (zöld) logaritmikus görbületi grafikonja.

Felírva más logaritmikus görbületi sugár grafikonokat további  $\alpha > 1$  esetekben, hasonló eredményeket kapunk. Azaz az izooptikus LCG meredeksége nem konstans, de az irányszöggel végtelenbe tartva  $\hat{\alpha}(\theta)$ -ra a vizsgált log-aesthetic görbe  $\alpha$  alakparaméterét kapjuk. Ezért, habár az izooptikus geometriai tulajdonságában is hasonlít az eredeti görbére, a következő következtetést vonhatjuk le: a log-aesthetic görbe  $\alpha > 1$  esetben nem autoizooptikus.

Vizsgáljuk meg továbbá az  $\alpha = -1$  esetet is (amikor a görbe az Euler-féle spirál, 20.(d) ábra). Ekkor a log-aesthetic görbe felírható Fresnel-integrálok segítségével [66], illetve az LCG is könnyedén meghatározható:

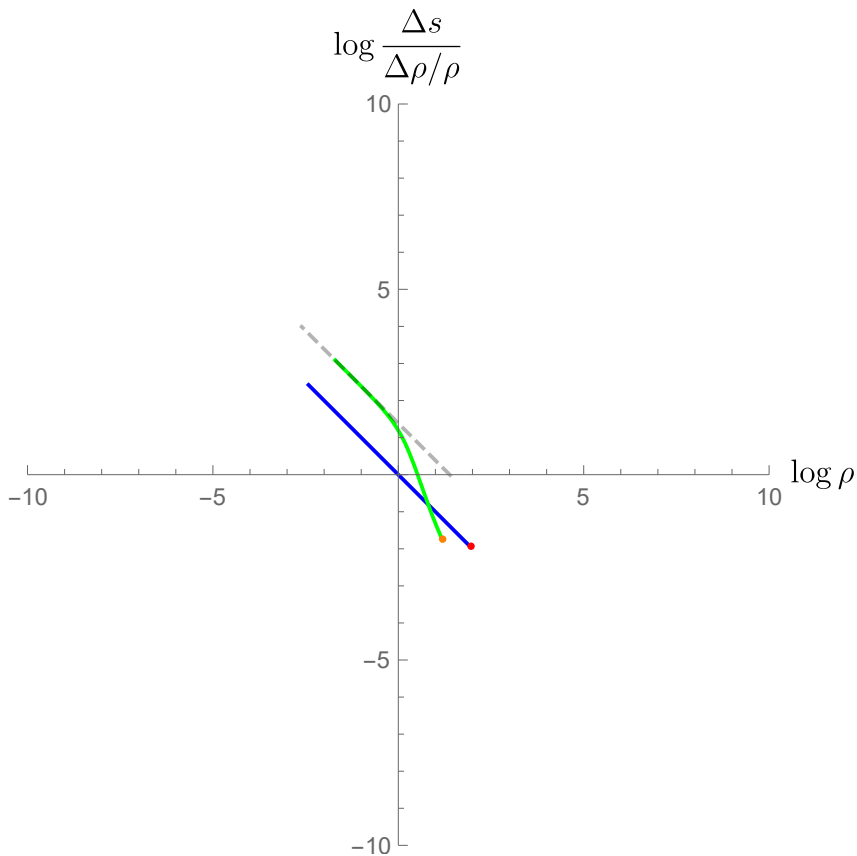
$$P_{LCG}^{\alpha=-1}(\theta) = \left[ \begin{array}{c} -\frac{1}{2} \log(1 - 2\theta\Lambda) \\ \frac{1}{2} \log(1 - 2\theta\Lambda) - \log(\Lambda) \end{array} \right]. \quad (3.23)$$

Továbbá az izooptikus görbe LCG képlete is kiszámolható komputeralgebrai szoftver segítségével, azonban a felírása túl bonyolult és hosszú, még megadott  $\Lambda$  és  $\delta$  paraméterrel is. Kiszámoltuk azonban az  $\hat{\alpha}(\theta)$  meredekséget  $\Lambda = 1$ ,  $\delta = \frac{2}{3}\pi$ ,  $\phi = \pi$ , illetve különböző  $\theta$  paraméterekkel a (3.17) egyenlethez hasonlóan:

$$\begin{aligned} \hat{\alpha}\left(\frac{1}{2} - \delta\right) &\approx -2.23102 \text{ (az izooptikus } \theta \text{ felső korlátja)} \\ \hat{\alpha}(-\pi) &\approx -1.27031 \\ \hat{\alpha}(-2\pi) &\approx -1.0705 \\ \hat{\alpha}(-5\pi) &\approx -1.01251 \\ \hat{\alpha}(-10\pi) &\approx -1.00329 \\ \hat{\alpha}(-100\pi) &\approx -1.00003 \\ \lim_{\theta \rightarrow -\infty} \hat{\alpha}(\theta) &= -1. \end{aligned} \quad (3.24)$$

Belátható tehát, hogy az  $\alpha = -1$ -gyel megadott log-aesthetic görbe, azaz az Euler-féle spirál sem autoizooptikus (23. ábra). Habár, az

$\hat{\alpha}(\theta)$  határ  $-1$ , ha  $\theta$ -val mínusz végtelenbe tartunk. Érdekes azonban az LCG meredekségek nagyfokú eltérése a  $\theta$  felső korlátjának közelében, mely oka az lehet, hogy  $\alpha < 1$  esetben a felső korlát inflexiós pont.



23. ábra.  $\alpha = -1$  és  $\Lambda = 1$  paraméterrel megadott log-aesthetic görbe (kék), annak transzformáltja (szaggatott vonal) és a  $\delta = \frac{2}{3}\pi$  szöggel számolt izooptikus görbe (zöld) logaritmikus görbületi grafikonja.

A különböző alakparaméterrel megadott log-aesthetic görbék izooptikusait meghatározva beláthatjuk, hogy tetszőleges  $\alpha \in \mathbb{R}$  esetén a görbe nem autoizooptikus. Továbbá a (3.13) egyenletet megvizsgálva, úgy gondoljuk, hogy a görbe izooptikusa csak az  $\alpha = 1$  és az  $\alpha = \pm\infty$  esetben log-aesthetic görbe, mert ezekben az esetekben a  $\theta$  és a  $\vec{V}^\delta(\theta)$  vektor irányszöge egyenesen arányos ( $\theta \propto \arctan2(\vec{V}_y^\delta(\theta), \vec{V}_x^\delta(\theta))$ ), azaz a  $\sin(\theta + \delta - \arctan2(\vec{V}_y^\delta(\theta), \vec{V}_x^\delta(\theta)))$  konstans, továbbá a  $\|\vec{V}^\delta(\theta)\|$  a görbületi sugár konstansszorososa. A további esetekben ( $-\infty < \alpha < 1$ , illetve  $1 < \alpha < \infty$ ) a  $\theta$  és a  $\vec{V}^\delta(\theta)$  vektor irányszöge között az arányosság nem áll fenn, így a további összefüggések sem érvényesek. Úgy gondoljuk, hogy ez az oka annak, hogy az izooptikus nem log-aesthetic görbe ezekben az esetekben.

### 3.2.2. Továbbfejlesztett modellező algoritmus

A Yoshida és Saito által fejlesztett modellező algoritmus [64], ahogy azt a 3.1.3. fejezetben láthattuk erősen korlátozott a görbe  $s$  és  $\theta$  paraméterének alsó, illetve felső korlátja miatt. A log-aesthetic görbemodellező technikák többségénél hasonló nehézségekbe ütközünk, mert a felhasználó által megadott geometriai adatoknak gyakran szigorú feltételeknek kell megfelelniük.

Az ötletünk, hogy a bemutatott eljárást továbbfejlesszük és ezáltal a log-aesthetic görbével történő modellezési lehetőségeket kibővítsük, a megrajzolható régió megnövelésével. A megoldásunkhoz bevezetjük az úgynevezett kiterjesztett log-aesthetic görbét, melyet a következő 3.2.2.1 alfejezetben mutatunk be [47]. Ezen újszerű görbére alkalmazzuk a Yoshida-Saito-féle [64] algoritmust, mely megfelelő módosításait a 3.2.2.2 alfejezetben ismertetjük.

A fejezet utolsó szakaszában az ismertett módszer lehetőségeit tovább bővítjük. Az új algoritmusunk egy kezdőpont és az abban megadott érintővektor, illetve egy végpontbeli érintőegyenestől meg-

tározott kiterjesztett log-aesthetic görbeszegmenst generál úgy, hogy a  $\Lambda$  paraméteren túl az  $\alpha$  értékét is automatikusan határozza meg. Ez több szempontból is fontos. Harada és társai alapján [22, 21] az  $\alpha$  összefüggésben van a generált görbe látványával. A gyakorlatban azonban sokszor nehézséget okoz az optimális paraméter kiválasztása, ezért általában egy rögzített értéket használnak a modellezés során. Az új módszerünkben a felhasználó ezen  $\alpha$  értéket a görbe geometriai tulajdonsága (a kezdőpontbeli érintővektor) megadásával implicit módon definiálja.

Emellett, ezen megadás az első pont görbületét is szabályozza, ami lehetővé teszi log-aesthetic görbék  $G^2$  folytonos csatlakoztatását is, mert a görbe utolsó pontjában a tangens vektor kiszámítható a görbe deriváltjából, melynek hossza megegyezik a görbületi sugár értékével. Tehát ha a csatlakozási pontban az érintővektor iránya és hossza is megegyezik, akkor az illeszkedő log-aesthetic görbék görbülete is. Az új algoritmus ezen lépését, melyben az optimális  $\alpha$  paramétert határozzuk meg, a szakasz utolsó 3.2.2.3 alfejezetében ismertetjük. Megjegyezzük, hogy külön nem kezeljük azon speciális esetet, amikor a megadott kezdő vektor párhuzamos az utolsó pontban megadott érintőegyenestől irányával.

### 3.2.2.1. Kiterjesztett log-aesthetic görbék

A log-aesthetic görbe kiterjesztését az  $s$  és  $\theta$  paraméter korlátjain túli értelmezéssel érjük el úgy, hogy a görbét tükrözzük a határpontokban. Az  $\alpha$  értékétől függően négy esetet különböztethetünk meg: amikor  $\alpha > 1$ ,  $\alpha = 1$ ,  $0 \leq \alpha < 1$  és  $\alpha < 0$ . A különböző esetekben figyelembe vesszük a bemutatott algoritmus irányszögének intervallumait, hiszen célunk ezen eljárás továbbfejlesztése.

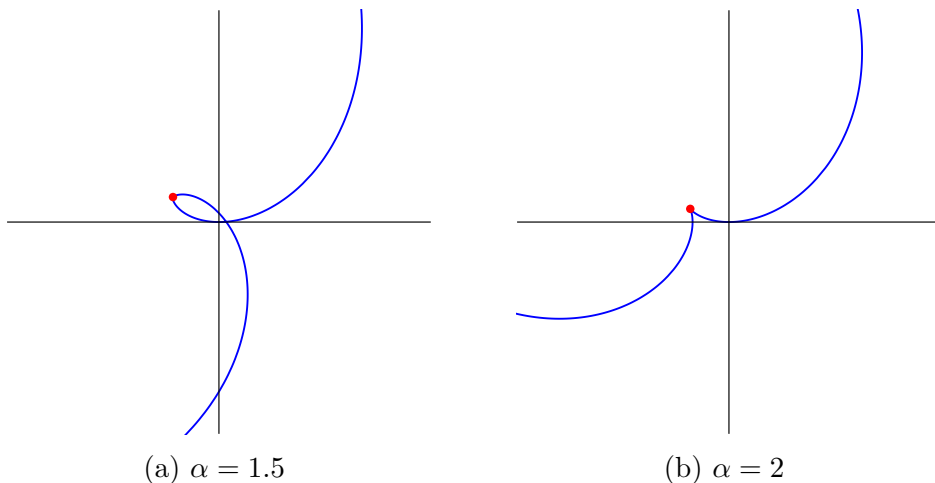
Amikor  $\alpha > 1$ , az irányszög  $0$  és  $-\theta_\Delta$  között lehet, de a log-aesthetic görbének alsó korlátja van  $\theta = \frac{1}{\Lambda(1-\alpha)}$ -ban, ahol az irányvektor eltűnik és  $\rho = 0$ . Ez tehát a görbe egy szinguláris pontja. Ahhoz, hogy az algo-

ritmus megrajzolható régióját megnöveljük, definiálni kell a függvényt ezen határponton túl is. A legtöbb esetben azonban nem számolhatók ki a görbe további pontjai a negatív alapú törthatványok miatt, ezért a 3. táblázatban írt megszorítást alkalmazzuk. Ennek ellenére előfordulnak olyan speciális esetek, amikor a görbe mégis meghatározható a teljes  $\theta$  tartományon, például  $\alpha = 1.5$  esetén (24.(a) ábra). Ekkor a (3.2) egyenlet a következő:

$$P(\theta) = \int_0^\theta \left( \frac{\psi\Lambda}{2} + 1 \right)^2 e^{i\psi} d\psi. \quad (3.25)$$

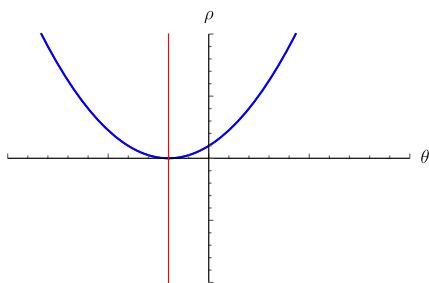
Továbbá, amikor  $\alpha = 2$ , a log-aesthetic görbe a körevolvens (24.(b) ábra) és a (3.2) egyenlet a következő:

$$P(\theta) = \int_0^\theta (\psi\Lambda + 1e^{i\psi}) d\psi. \quad (3.26)$$

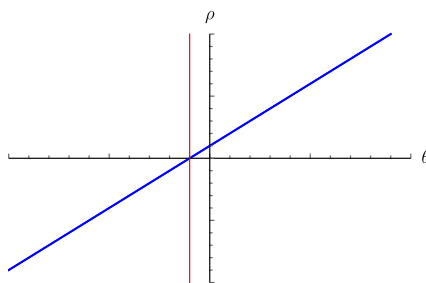


24. ábra. Log-aesthetic görbék a  $\theta$ -ra értelmezett megszorítás nélkül. A továbbfejlesztett algoritmus megoldásához a (b) ábra az előnyösebb.

Mivel a görbét generáló (3.2) egyenlet tartalmazza a  $\rho$  (3.3) egyenletét, definiáljuk a kiterjesztést a görbületi sugár segítségével, ami az említett  $\alpha = 1.5$  esetben  $\rho = \left(\frac{\theta\Lambda}{2} + 1\right)^2$  (25.(a) ábra), illetve  $\rho = \theta\Lambda + 1$ , amikor  $\alpha = 2$  (25.(b) ábra).



(a)  $\alpha = 1.5$



(b)  $\alpha = 2$

25. ábra. A görbületi sugár a  $\theta$ -ra értelmezett megszorítás nélkül. A piros vonal az eredeti korlátot jelöli. A továbbfejlesztett algoritmus megoldásához a (b) ábra az előnyösebb.

A kétértelműség megszüntetése végett definiáljuk  $\alpha > 1$  esetben a log-aesthetic görbét a teljes  $\theta$  tartományon úgy, hogy a 24.(b) ábrához hasonló eredményt kapjunk, azaz tükrözzük a görbületi sugarát a határvonalon és a  $\theta$  tengelyen is a 25.(b) ábrának megfelelően. Így a kiterjesztett görbületi sugár  $\alpha > 1$  esetben a következőképpen adható meg:

$$\rho_{ext}^{\alpha>1}(\theta) = \begin{cases} \left((\alpha - 1)\Lambda\theta + 1\right)^{\frac{1}{\alpha-1}} & \theta > B_\theta \\ -\left((1 - \alpha)\theta\Lambda - 1\right)^{\frac{1}{\alpha-1}} & \theta \leq B_\theta, \end{cases} \quad (3.27)$$

ahol a  $B_\theta = \frac{1}{\Lambda(1-\alpha)}$ , a  $\theta$  korábbi korlátja. Ezek alapján egy  $\theta$  irányszöggel megadott  $P(\theta)$  kiterjesztett log-aesthetic görbepont koordinátáit a következőképpen számolhatjuk ki a komplex síkon:

$$P_{ext}^{\alpha>1}(\theta) = \begin{cases} \int_0^\theta ((\alpha - 1)\Lambda\psi + 1)^{\frac{1}{\alpha-1}} e^{i\psi} d\psi & \theta > B_\theta \\ \int_0^\theta -((1 - \alpha)\Lambda\psi - 1)^{\frac{1}{\alpha-1}} e^{i\psi} d\psi & \theta \leq B_\theta. \end{cases}$$

A  $\theta = \frac{1}{\Lambda(1-\alpha)}$  pontban a kiterjesztett log-aesthetic görbének csúcspontja (cusp) van.

$\alpha = 1$  esetben a Yoshida és Saito algoritmus [64] szerint a log-aesthetic görbe irányszöge a  $[0, \theta_\Delta]$  intervallumban van és nincs alsó, illetve felső korlátja  $\theta$ -nak és  $s$ -nek sem, ezért nincs szükség a log-aesthetic görbe kiterjesztésére.

Amikor  $\alpha < 1$  az irányszög intervalluma  $[0, \theta_\Delta]$  és  $\theta$ -nak felső korlátja van  $\frac{1}{1-\alpha}$ -ban. Azonban ez nem probléma, amíg  $\alpha \geq 0$ , mert ebben a  $\theta = \frac{1}{1-\alpha}$  pontban  $s$  végtelen. Az említett szerzők is a tesztek alapján nagy rajzolható régióról számolnak be ebben az esetben [64], ezért nincs szükség a log-aesthetic görbe kiterjesztésére, ha  $0 \leq \alpha < 1$ .

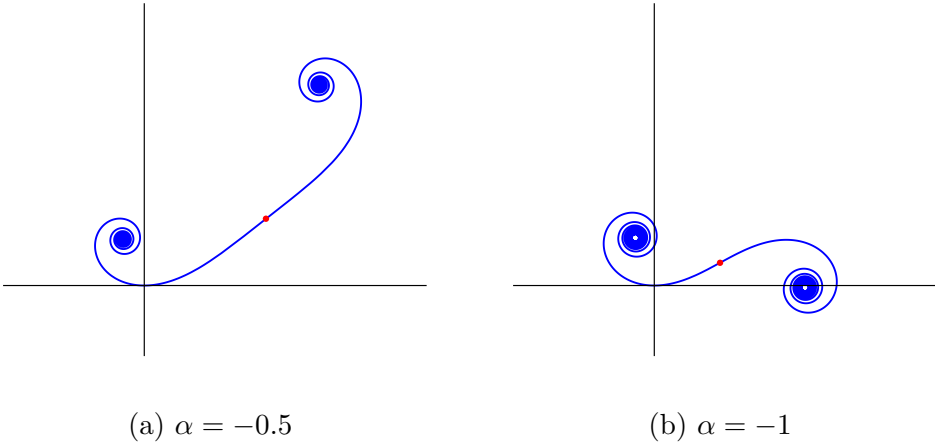
Ezzel szemben  $\alpha < 0$  esetében nem csak  $\theta$ , de az  $s$  húrhossz is korlátos  $-\frac{1}{\Lambda\alpha}$ -ban. Az irányszög továbbra is a  $[0, \theta_\Delta]$  intervallumban van, tehát a log-aesthetic görbe kiterjesztése szükséges, mert a megrajzolható régió méretét a határpont korlátozza. Yoshida és Saito szerint [64]  $\alpha < 0$  esetben a görbének inflexiós pontja van a határpontban, azaz a görbületi sugár végtelen (27. ábra), ezért az  $s$  húrhossz szerinti paraméterezés használata célszerűbb, az irányszög szerinti egyenlettel szemben. Azonban a (3.5) görbét generáló egyenletre is érvényes a 3. táblázat korlátozása a negatív alapú törthatványok kiküszöbölésére. Ennek ellenére itt is találunk olyan  $\alpha$  értéket, melyek esetén a log-aesthetic görbe a teljes  $s$  tartományon meghatározható. Ilyen például,

amikor  $\alpha = -0.5$  (26.(a) ábra), ekkor a (3.5) egyenlet a következő:

$$C(s) = \int_0^s \exp\left(i \frac{1}{12} \Lambda u^2 (\Lambda u - 6) + u\right) du, \quad (3.28)$$

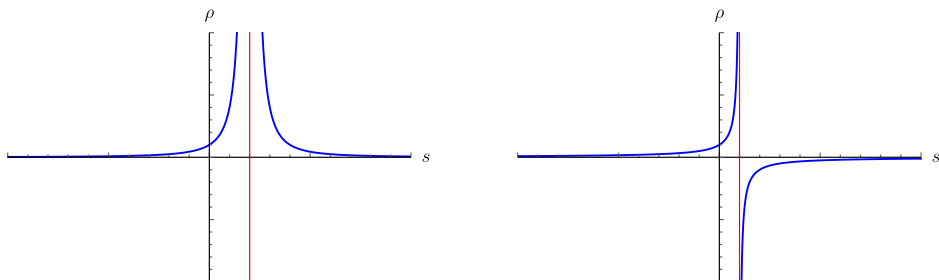
illetve az  $\alpha = -1$  eset, amikor a görbe az Euler-féle spirál (26.(b) ábra):

$$C(s) = \int_0^s \exp\left(iu - \frac{\Lambda u^2}{2}\right) du. \quad (3.29)$$



26. ábra. Log-aesthetic görbék az  $s$  húrhosszra értelmezett megszorítás nélkül. A továbbfejlesztett algoritmus megoldásához a (b) ábra az előnyösebb.

Az  $\alpha$  értékétől függő különbségek láthatóak a görbületi sugár képén is. Ugyanis amikor  $\alpha = -0.5$ , a  $\rho = \frac{4}{(\Lambda s - 2)^2}$  (27.(a) ábra), illetve  $\alpha = -1$  esetén  $\rho = \frac{1}{1 - \Lambda s}$  (27.(b) ábra).



(a)  $\alpha = -0.5$

(b)  $\alpha = -1$

27. ábra. A görbületi sugár az  $s$  húrhosszra értelmezett megszorítás nélkül. A piros vonal az eredeti korlátot jelöli. A továbbfejlesztett algoritmus megoldásához a (b) ábra az előnyösebb.

A célunk a kiterjesztéssel, hogy a log-aesthetic görbe megrajzolható régióját növeljük a modellezési algoritmusok esetében, ezért úgy érdekes a görbét meghosszabbítani, hogy az inflexiós pontot tartalmazzon (ahogyan az  $\alpha = -1$  esetben látható). A tükrözést a görbületi sugáron alkalmazzuk, hasonlóan az előző esethez. Azaz  $\rho$ -t a görbe korlátja által definiált egyenesen, illetve az  $s$  tengelyen tükrözzük (27(b). ábra). Így, amikor  $\alpha < 0$  a görbületi sugár függvénye a következő:

$$\rho_{ext}^{\alpha < 0}(s) = \begin{cases} (\alpha\Lambda s + 1)^{\frac{1}{\alpha}} & s < B_s \\ -(-\alpha\Lambda s - 1)^{\frac{1}{\alpha}} & s \geq B_s, \end{cases} \quad (3.30)$$

ahol  $B_s = -\frac{1}{\alpha\Lambda}$  a húrhossz korábbi korlátja.

Ezek alapján egy kiterjesztett log-aesthetic görbe  $s$  húrhosszal megadott pontjának koordinátáit a következő egyenlettel számolhatjuk ki

a komplex síkon:

$$C_{ext}^{\alpha < 0}(s) = \begin{cases} \int_0^s \exp\left(i \frac{(\alpha\Lambda u + 1)^{\frac{\alpha-1}{\alpha}} - 1}{(\alpha-1)\Lambda}\right) du & s \leq B_s \\ \int_0^s \exp\left(i \frac{(\alpha\Lambda(-u) - 1)^{1-\frac{1}{\alpha}} - 1}{(\alpha-1)\Lambda}\right) du & s > B_s. \end{cases} \quad (3.31)$$

Az  $s = -\frac{1}{\alpha\Lambda}$ -ban a kiterjesztett log-aesthetic görbének inflexiós pontja van.

A bemutatott kiterjesztések két különböző módon tükrözik az eredeti log-aesthetic görbét irányyszög, illetve húr hossz szerint. Az új egyenletek különböző görbéket határoznak meg, melyekre a 3. táblázat megszorításai nem vonatkoznak. Ezek közül  $\alpha > 1$  esetben a (3.28),  $\alpha < 0$  esetben a (3.31) egyenletet kell használni. Egyébként, amikor  $0 \leq \alpha \leq 1$ , a log-aesthetic görbe az eredeti képletek bármelyikével generálható, de a megfelelő megszorítások figyelembevételével: felső korlátja van  $\theta$ -nak, illetve alsó korlátja  $s$ -nek. Ezért érdemes a (3.2) húr hossz szerinti paraméterezést használni, ha  $s \leq 0$  (illetve  $\theta \leq 0$ ) és a (3.5) irányyszög szerinti paraméterezést alkalmazni  $s > 0$  (és  $\theta > 0$ ) esetben.

A Yoshida és Saito algoritmusának továbbfejlesztéséhez azonban szükség van a húr hossz szerinti egyenletek esetén az  $s$  meghatározására a megadott  $\theta_\Delta$  irányszögből. Ha  $0 \leq \alpha \leq 1$ , ez kiszámolható a következőképpen (a (3.4) összefüggésből):

$$\theta(s)^{-1} = S(\theta) = \begin{cases} -\frac{\log(1-\theta\Lambda)}{\Lambda} & \alpha = 0 \\ \frac{e^{\theta\Lambda} - 1}{\Lambda} & \alpha = 1 \\ \frac{((\alpha-1)\theta\Lambda + 1)^{\frac{\alpha}{\alpha-1}} - 1}{\alpha\Lambda} & 0 < \alpha < 1. \end{cases} \quad (3.32)$$

Az  $S(\theta)$  megadása  $\alpha < 0$  esetben azonban nem egyértelmű. A (3.32) egyenletben a  $\theta$  irányyszög felülről korlátos ( $\alpha = 1$  kivételével), az  $s$  húr hossz pedig a végtelenbe tart (mivel  $S(\frac{1}{\Lambda(1-\alpha)}) = \infty$ ). A

bemutatott tükrözéssel a húrhossz  $\alpha < 0$  esetben is végtelenbe tart, viszont az irányszög növekszik az inflexiós pont előtt, de csökken utána (27.(b) ábra). Ez azt jelenti, hogy  $\theta$  továbbra is korlátos. Tehát az  $S(\theta)$  nem kölcsönösen egyértelmű leképezés, ezért definiálható a következőképpen:

$$S(\theta)_{within}^{\alpha < 0} = \frac{((\alpha - 1)\theta\Lambda + 1)^{\frac{\alpha}{\alpha-1}} - 1}{\alpha\Lambda}, \quad (3.33)$$

vagy az alábbi módon:

$$S(\theta)_{beyond}^{\alpha < 0} = -\frac{((\alpha - 1)\theta\Lambda + 1)^{\frac{\alpha}{\alpha-1}} + 1}{\alpha\Lambda}, \quad (3.34)$$

attól függően, hogy az adott pont az inflexiós pont előtt vagy után helyezkedik el. Mivel a log-aesthetic görbe ezen részei egymás tükörképei, ezért a következő összefüggés teljesül:

$$S(\theta)_{within}^{\alpha < 0} = S(2B_\theta - \theta)_{beyond}^{\alpha < 0}, \quad (3.35)$$

ahol  $B_\theta = \frac{1}{(1-\alpha)\Lambda}$ , a  $\theta$  felső korlátja, a görbe inflexiós pontja.

### 3.2.2.2. Kibővített interpolációs eljárás

Az új eljárás a 3.1.3. szakaszban bemutatott Yoshida-féle [64] algoritmushoz hasonlóan egy felező módszert alkalmaz a  $\Lambda$  paraméterre és egy kiterjesztett log-aesthetic görbeszegmenst illeszt a megadott  $ABC$  háromszögbe. Azonban az előállítandó görberész megadása új módon, a két végpontja ( $A$  és  $C$ ), illetve a  $\vec{v}_A$  és  $\vec{v}_C$  vektorok segítségével történik. Előbbi vektor a kezdőpont iránytangense, utóbbi pedig a végpontbeli érintőegyenes irányát határozza meg. Továbbá az algoritmushoz szükséges  $\theta_\Delta$  irányszöveget az említett vektorok által bezárt szög segítségével határozzuk meg.

A felező módszer (ahogyan [64]-ben is) egy ciklusban folyamatosan felezi a  $\Lambda$  intervallumot úgy, hogy az összehasonlításban vizsgált szögek

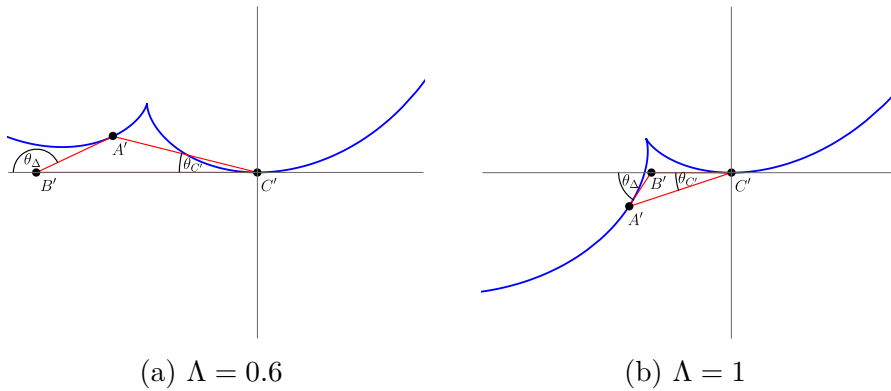
abszolút eltérése minden lépésben egyre kevesebb legyen. Ezen szögek a [64] alapján  $\alpha \leq 1$  esetben  $\theta_A$  és  $\theta_{A'}$ , illetve  $\theta_C$  és  $\theta_{C'}$ , ha  $\alpha > 1$ . Emellett, ahogyan az előbbi algoritmusban is, az  $A$  és  $C$  pont koordinátáit fel kell cserélni, ha az  $|AB| \leq |BC|$  összefüggés nem áll fenn. Ezen koordinátacserét az új algoritmusban egy flag változóban is tároljuk a további lépések könnyítése végett.

Mivel a log-aesthetic görbét a  $0 \leq \alpha \leq 1$  esetben nem bővítettük, ezért ekkor a Yoshida-féle algoritmus módosítás nélkül alkalmazható. A további esetekben azonban a felező módszer végrehajtása közben néhány változtatás szükséges.

Amikor  $\alpha < 0$ , az irányszög a referenciaponttól az inflexiós pontig nő, ahol  $\theta = \frac{1}{\Lambda(1-\alpha)}$  és csökken ezután. Ez azt jelenti, hogy a  $\Lambda$  intervallum továbbra is  $0$  és  $1/(\theta_\Delta(1-\alpha))$  között lesz, de a ciklus egyes lépéseiben meg kell határozni, hogy a  $C'$  pont az inflexiós mely oldalán van, mert a következő lépés megfelelő  $\Lambda$  intervalluma ettől függően választandó. Ettől függ továbbá, hogy az algoritmusban mely egyenletet, a (3.33) vagy a (3.34) kell használnunk a húr hossz meghatározására.

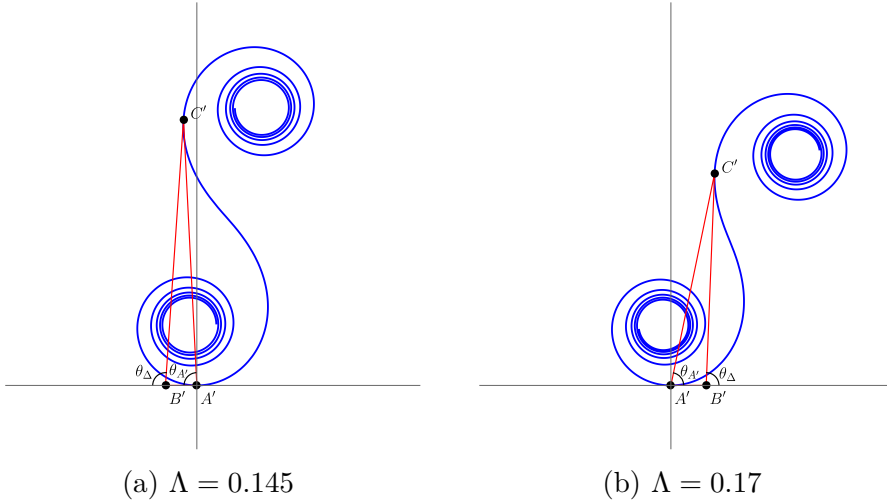
Ezzel szemben, amikor  $\alpha > 1$ , az irányszög csökken a referenciaponttól, viszont nem csak a  $\theta = \frac{1}{\Lambda(1-\alpha)}$  csúcspontig, hanem azon túl is. Ezért ebben az esetben a log-aesthetic görbe bővítése a  $\Lambda$  intervallum megnagyobbítását is megköveteli, hogy a csúcsponton túlra is kiterjesszük a keresést. Az új intervallum tehát a  $[0, 2/(\theta_\Delta(1-\alpha))]$  lesz.

Továbbra is az  $\alpha > 1$  esetet vizsgálva megállapíthatjuk, hogy a Yoshida-féle algoritmusban, ha a  $\Lambda$  értéke nő, azzal párhuzamosan a  $\theta_{C'}$  szög értéke is. A kiterjesztett görbét interpoláló felező módszer egyes ciklusiterációja esetén előfordulhat azonban, hogy a  $\theta_{C'}$  csökken (amikor az  $A'$  pont az  $X$  tengely alá esik), mert az  $A'B'C'$  háromszög pontjainak sorrendje megváltozik ( $A'C'B'$ -re, 28. ábra). Ezért az eredeti algoritmusához képest ellentétesen kell a következő lépés  $\Lambda$  intervallumát választani, az  $A'$  pont függvényében.



28. ábra. Az  $A'$  pont helyzetétől függően előforduló, különböző orientációjú  $A'B'C'$  háromszögek a  $\Lambda$  felező módszer közben ( $\alpha = 2$  esetben).

Amikor  $\alpha < 0$  az említett orientációváltozáson túl más eset is előfordulhat. Az eredeti Yoshida-féle algoritmusban az  $A'$  és  $C'$  pontok érintőegyenesei mindig az  $Y$  tengely jobb oldalán metszik egymást. Az új felező módszer esetén a  $B'$  metszéspont ezen tengely bal oldalára is eshet bizonyos cikluslépésekben a kiterjesztés miatt, ami ugyancsak az  $A'B'C'$  háromszögcsúcsok sorrendjének felcserélődését okozza, ahogyan azt a 29. ábrán láthatjuk. Ebben az esetben is az eredeti algoritmushoz képest ellentétesen kell választani a következő ciklusiteráció  $\Lambda$  intervallumát.



29. ábra. A  $B'$  pont helyzetétől függően előforduló, különböző orientációjú  $A'B'C'$  háromszögek a  $\Lambda$  felező módszer közben ( $\alpha = 2$  esetben).

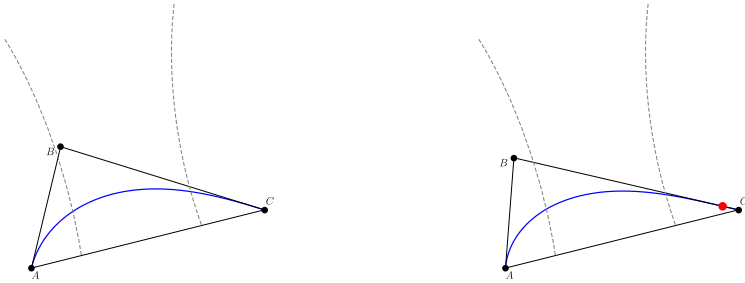
Az új, kiterjesztett log-aesthetic görbét interpoláló  $\Lambda$  felező módszer pszeudokódját, mely megoldja az említett orientációs problémákat a Függelékben található 5. kódrészlet tartalmazza. A Yoshida-féle algoritmus [64] megfelelő módosításait piros színnel emeltük ki. Az új algoritmus log-aesthetic görbék korlátlan megrajzolható régióval történő modellezését teszi lehetővé, a  $0 \leq \alpha < 1$  esetet kivéve, ahol a [64] szerzői szerint a kontrollpontok elhelyezésének kis korlátozását valószínűleg a nagy integrációs tartomány számítási hibája okozza.

A hagyományos görbemodellezésben az inflexiós és csúcspontok szokatlanok lehetnek és a különböző interpolációs feladatok esetén legtöbbször igyekszünk ezeket elkerülni. Mégis, bizonyos esetekben, például automatikus robot eszközök pályagörbéinek tervezésénél az inflexiós pontok elkerülhetetlenek.

Továbbá Zhonggui Chen és társai [4] az említett görbejellemzők direkt kontrollálására adnak lehetőséget, az általuk FPC-nek (Fea-

ture Points Controlled interpolatory) nevezett görbe konstruálásával. Az említett cikkben nem csak inflexiós pontokra, hanem csúcspontok használatára is láthatunk példákat. Log-aesthetic görbék esetén pedig Miura és társai a csúcspontok fontosságát a görbe inflexiós pontbeli hirtelen irányváltoztatása esetén fellépő „ugró” viselkedésének elkerülésében látják [42].

Az általunk felírt módszerben is a felhasználó döntése az inflexiós, illetve csúcspontok használata, illetve ezen pontok pozícióját is szabályozhatja, melyhez segítségként a modellezés közben szemléltethetjük az úgynevezett megrajzolható régiót, a [65]-ben leírt algoritmus alapján, ahogyan például a 30. ábrán is láthatjuk.



(a) A  $B$  pont belül van

(b) A  $B$  pont kívül van

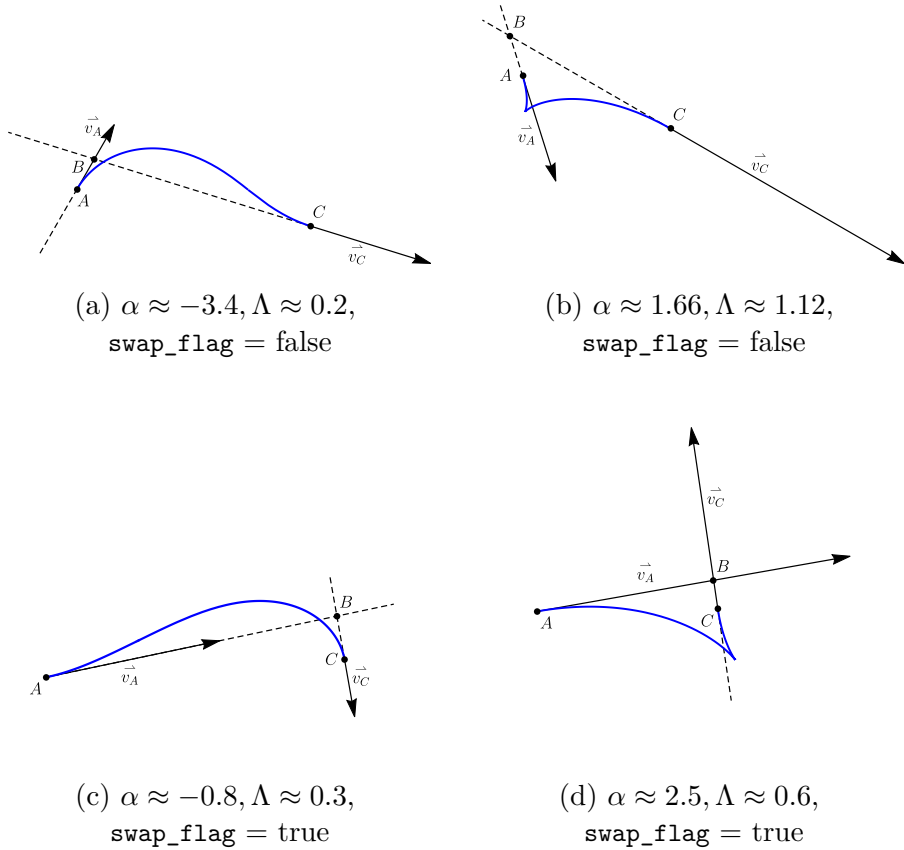
30. ábra. A megrajzolható régió (szaggatott vonalak közötti térrész) szemléltetése.  $\alpha < 1$  esetben inflexiós pont jelenik meg a  $C$  pont közelében, ha a görbe végpontjainak  $B$  metszéspontja a megrajzolható régió bal oldalán kívülre esik.

### 3.2.2.3. Kezdőpont görbületének szabályozása

A modellező algoritmus  $\Lambda$  felező módszere a megadott geometriai adatok segítségével ( $A$  és  $C$  pont, illetve  $\vec{v}_A$  és  $\vec{v}_C$  vektor) interpo-

lál egy kiterjesztett log-aesthetic görbeszegmenst tetszőleges  $\alpha \in \mathbb{R}$  értékkel. Ebben az alfejezetben ezen algoritmust egészítjük ki egy új lépéssel, melyben meghatározzuk a megfelelő  $\alpha$  paramétert úgy, hogy az előállított görbe kezdőpontbeli érintőjének hossza megegyezzen a megadott  $\vec{v}_A$  vektor hosszával. Ezáltal az eljárás felhasználási lehetősége is bővül, hiszen a felhasználó szabályozhatja a kezdőpont görbületét is, amely log-aesthetic görbék  $G^2$  folytonos csatlakozását teszi lehetővé. Mivel azonos geometriai adatokkal, de különböző  $\alpha$ -val megadott görbeszegmens meghatározásához eltérő  $\Lambda$  paraméter szükséges, ezért az  $\alpha$  egzakt számolása nem lehetséges, tehát egy új felező módszer alkalmazunk az  $\alpha$  értékének meghatározására.

Az algoritmusban két esetet különböztetünk meg az alapján, hogy a  $B$  mely végponthoz ( $A$  vagy  $C$ ) van közelebb. Ezeket az előző fejezetben említett `swap_flag` változóban tároljuk, mely értéke hamis, ha  $|AB| \leq |BC|$ , illetve igaz egyébként (mert a végpontok koordinátáinak felcserélése szükséges). Hamis esetben, tehát ha a  $B$  pont az  $A$ -hoz van közelebb és a  $\vec{v}_A$  vektor a  $B$  pontba mutat, az illesztett log-aesthetic görbe inflexiós pontot tartalmazhat, illetve ha a  $\vec{v}_A$  ellentétes irányú, akkor csúcspontot (31.(a) és 31.(b) ábra). Amennyiben a  $B$  pont a  $C$  végponthoz van közelebb (azaz a `swap_flag` értéke igaz), a log-aesthetic görbeszegmenst ezen végponttól az  $A$  pontig generáljuk, a koordináták felcserélése miatt. Ezért ebben az esetben a feltételezettel ellentétes irányú érintővektorok keletkeznek, melyek kiküszöbölése érdekében érdemes fordított paraméterezést alkalmazni, hogy a 31.(c), illetve 31.(d) ábrán látható eredményeket kapjuk. Így, ha a  $\vec{v}_C$  vektor a  $B$  pont felé mutat, akkor a görbeszegmens csúcspontot, egyébként inflexiós pontot tartalmazhat.



31. ábra. Az  $\alpha$  felező módszer példa eredményei.

Először tekintsük azt az esetet, amikor nem hajtunk végre koordinátacserét, tehát a `swap_flag` hamis, azaz a  $B$  pont közelebb van az  $A$ -hoz ( $|AB| \leq |CB|$ ). Ekkor, ha  $\alpha \leq 1$ , az  $A'$  a referencia pont, ahol a log-aesthetic görbe érintővektora egységnyi hosszúságú (?? ábra). Ez igaz minden  $\Lambda$  felező módszer végrehajtása után. Ezt az érintővektort kell megfeleltetni a felhasználó által megadott  $A$ -beli  $\vec{v}_A$  vektorral, melynek hossza a görbeszegmens meghatározása után alkalmazott

transzformációban szereplő skálázás mértéke lesz. Ezen érték kiszámolható a végpontok távolságának arányából:  $|AC|/|A'C'|$ . Azonos  $\theta_\Delta$  esetén (tehát, ha a geometriai adatok nem változnak), az  $\alpha$  paraméter csökkenésekor az  $A'$  és  $C'$  pontok egymáshoz közelebb lesznek, mert ha  $\alpha < 1$ , akkor a log-aesthetic görbe spirálisan konvergál a  $\rho = 0$  pontjához, ahogy a  $\theta$  értékkel közeledünk  $-\infty$ -hez [64]. Alacsonyabb  $\alpha$  érték esetén ez a konvergencia gyorsabb, tehát az  $|A'C'|$  érték a legkisebb akkor, amikor  $\alpha = -\infty$  (vagyis a log-aesthetic görbe egy kör). Ebben az esetben a legnagyobb a transzformációban szereplő skálázás értéke, tehát a  $v_A$  hossza is.

Amikor  $\alpha > 1$  (és  $|AB| \leq |CB|$  továbbra is fennáll), a megadott  $A$  pont nem a referenciapontnak felel meg, hanem azon pontnak, ahol az irányszög  $-\theta_\Delta$  (?? ábra). Itt az érintővektor nem egység hosszú, tehát a  $v_A$  nem csak a transzformációban szereplő skálázástól függ, hanem az érintővektor hosszától is, amelyet a derivált segítségével számolhatunk ki.

A  $\Lambda$  felező módszerben  $\alpha > 1$  esetben az irányszög a  $[-\theta_\Delta, 0]$  intervallumban van, ahol a görbületi sugár a referenciaponttól csökken (és mivel egyenes arányosságban vannak, ezért az érintővektor hossza is). A  $\theta = \frac{1}{\Lambda(1-\alpha)}$  csúcspontban a  $\rho$  és az érintővektor is 0. Viszont a kiterjesztett log-aesthetic görbét használjuk, ezért a  $\Lambda$  intervallum is bővített, tehát az  $A'$  pont ezen csúcspont után is lehet, ahol a görbületi sugár előjele (és az érintővektor iránya is) megváltozik és a csúcspont után tovább csökken, amíg az érintővektor hossza  $-1$  nem lesz (a  $\theta = \frac{2}{\Lambda(1-\alpha)}$  pontban). A görbületi sugár ezen változását a 25.(b) ábrán figyelhetjük meg. Összefoglalva, a  $\Lambda$  felező módszerben, ha  $\alpha > 1$ , vagyis míg  $0 < \Lambda < 2/(\theta_\Delta(1-\alpha))$  az  $A'$  pontbeli érintővektor hossza a  $[-1, 1]$  intervallumban van. A  $v_A$  nagysága pedig, hasonlóan az előző esethez, ezen vektor és a transzformációban szereplő skálázás szorzata lesz. Azzal a különbséggel, hogy a log-aesthetic görbe  $\alpha > 1$  esetben spirálisan divergál, amíg  $\rho = \infty$  [64], és az  $\alpha$  paraméter növelése esetén

gyorsul a divergencia, egészen  $\alpha = +\infty$ -ig, ahol a log-aesthetic görbe egy kör és az érintővektora egység hosszú lesz megint.

Ezek alapján megállapíthatjuk, hogy a `swap_flag` hamis értéke esetén a  $\vec{v}_A$  érintővektor hossza egy jól definiált tartományban van, ahol az  $\alpha$  paraméter monoton változtatása ezen vektorhossz monoton módosulását vonja maga után. Egy bizonyos  $\alpha$  paraméter esetén az  $A$  kezdőpont a csúcspont lesz és a  $\vec{v}_A$  iránya megváltozik. Ezen  $\alpha$  érték alatt a log-aesthetic görbe tartalmazhat inflexiós pontot és felette a csúcspontot. Kiszámolva az említett  $\vec{v}_A$  vektorhossztartományt egy új felező módszert alkalmazhatunk  $\alpha$ -ra, hogy meghatározzuk azon értékét, ahol az  $A$  pontbeli érintővektor nagysága az előre megadott hosszal lesz egyenlő (amennyiben a kiszámolt intervallumba esik). A maximum hosszúságot  $\alpha = -\infty$  esetén számoljuk, amikor a kiterjesztett log-aesthetic görbe két egymást érintő  $[0 \ 1]^T$  és  $[0 \ -1]^T$  középpontú egységkör. Ebben az esetben a megadott  $A$  és  $C$  végpontok két ismeretlen középpontú, de egyenlő  $r$  sugarú érintőkörön helyezkednek el. A sugár megegyezik a transzformációban használandó skálázás értékével, illetve a  $\vec{v}_A$  és  $\vec{v}_C$  vektor hosszával (32.(a) ábra). Legyenek a körök középpontjai  $O_1$  és  $O_2$ . Ekkor az  $A$  és  $O_1$ , illetve a  $C$  és  $O_2$  pontokkal definiált egyenesek paraméteres egyenlete segítségével a következő összefüggés írható fel:

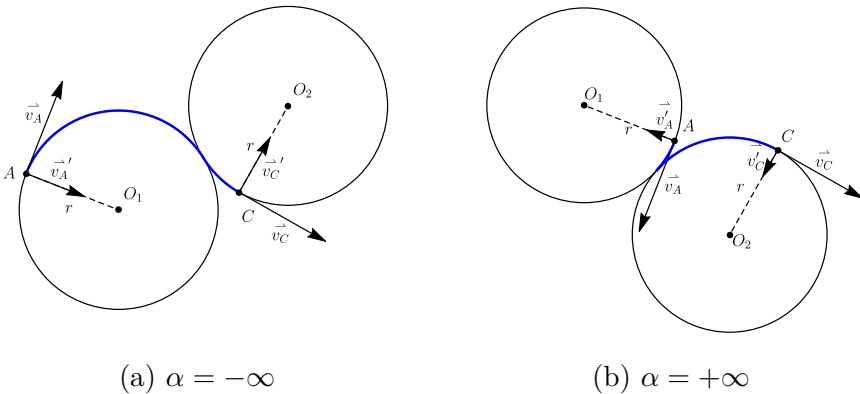
$$\begin{aligned} O_1 &= A + r \cdot \vec{v}_A' \\ O_2 &= C + r \cdot \vec{v}_C' \\ |O_2 - O_1| &= 2r, \end{aligned} \tag{3.36}$$

Ahol a  $\vec{v}_A'$  és a  $\vec{v}_C'$  az  $\overline{AO_1}$  és  $\overline{CO_2}$  irányítású egységvektorok (32.(a) ábra).

Mivel  $A$  és  $C$  adottak, illetve  $\vec{v}_A'$  és  $\vec{v}_C'$  kiszámolható, a (3.36) egyenletrendszer megoldható  $r$ -re, ami az  $A$  pontbeli érintővektor maximum hosszát adja  $\alpha \leq 1$  esetben. Fontos megjegyezni azonban, hogy

$\alpha = -\infty$  esetén a log-aesthetic görbe két fix középpontú kör a  $\Lambda$  paraméter értékétől függetlenül. Ez azt jelenti, hogy a  $\Lambda$  felező módszernek csak  $\alpha > -\infty$  esetben van megoldása, tehát az így számolt  $r$  érték egy elméleti maximum hosszúság, amelyet a  $\vec{v}_A$  soha nem ér el.

Ezzel szemben az  $A$  pontbeli érintővektor ellentétes irányú maximális hosszát az  $\alpha = \infty$  esetben számolhatjuk ki. A log-aesthetic görbe itt is két egymást érintő egységkör, ezért ezt is a (3.36) egyenletrendszerhez hasonlóan számoljuk ki, azzal a különbséggel, hogy a  $\vec{v}_A$  és  $\vec{v}_C$  irányát a 32.(b) ábrán látható módon definiáljuk. Az így meghatározott  $r$ , ebben az esetben is egy elméleti maximum hosszát adja a  $\vec{v}_A$  vektornak.



32. ábra. A  $\vec{v}_A$  és  $\vec{v}_C$  vektorok maximális hosszának számolása.

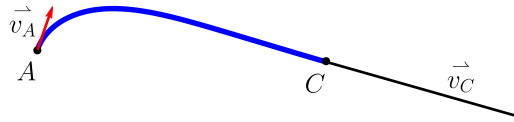
Hamis `swap_flag` érték esetén, amikor a  $B$  pont közelebb van a kezdőponthoz, mindig találunk olyan  $\alpha (> 1)$  értéket, amikor az  $A$  pont a csúcspont. Ezen érték felett és alatt az  $A'$  (és természetesen az  $A$ ) pont érintővektora ellentétes irányú. Ezért mindig egyedi  $\vec{v}_A$ -t kapunk a különböző  $\alpha$  értékek esetén. Ezzel szemben, amikor  $|AB| > |CB|$  és a `swap_flag` igaz, mindig találunk olyan  $\alpha (< 1)$  értéket, amikor az  $A$

pont az inflexiós pont, ahol a  $\vec{v}_A$  végtelen hosszú, és ezen  $\alpha$  paraméter alatt és felett az érintővektor hossza csökken, de a vektorirány azonos. A  $\vec{v}_A$  hossza tehát felülről nem korlátos, de egy minimum hossza van, ami attól függ, hogy az  $\alpha$  paraméter  $-\infty$ -t vagy  $+\infty$ -t közelíti. Ettől függően a kiterjesztett log-aesthetic görbeszelet csúcspontot, illetve inflexiós pontot tartalmazhat. A kezdőpont érintővektorának minimum értékét ugyancsak a (3.36) egyenletrendszerrel, illetve a 32. ábra segítségével számolhatjuk ki.

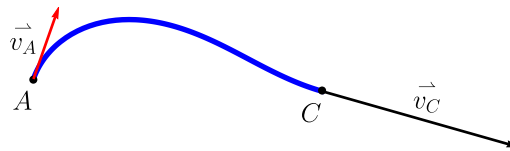
Mivel az algoritmus az  $\alpha$  paraméter értékét határozza meg, ezért a két esetet más módon kell megkülönböztetnünk. Megfigyelhetjük, hogy a  $C$  pont a görbén a csúcspont ellentétes oldalán található, amikor  $\alpha$ -val a  $+\infty$ , illetve a  $-\infty$  felé közeledünk, ami azt jelenti, hogy a  $\vec{v}_C$  vektor iránya a két esetben ellentétes. Ezért az algoritmusban a felhasználó ezen vektor irányának meghatározásával dönti el, hogy melyik lehetőséget választja, és ettől függően, ha a  $C$  pont érintővektora a  $B$  pont felé mutat a görbeszegmens csúcspontot tartalmazhat, vagy inflexiós pontot egyébként.

Koordinátacsere esetén, azaz igazra állított `swap_flag`-gel történő felező módszerben az egyes lépések megfelelő  $\alpha$  intervallumát az inflexiós pont helyzetéből következtethetjük, azaz a következő intervallum választásánál a  $\Lambda$  felező módszerben (5. kódrészlet) használt `beyond_inf_point` változó segítségével. Továbbá az inflexiós pontot is magában foglaló esetben (amikor a  $\vec{v}_C$  a  $B$  ponttal ellentétes irányba mutat) a rákövetkező intervallum meghatározásánál ellentétesen kell választani (mint amikor a  $\vec{v}_C$  a  $B$  pont felé mutat), hiszen alacsonyabb  $\alpha$  érték szükséges, hogy az  $A$  ponttal közeledjünk az inflexiós ponthoz és kapjunk hosszabb  $\vec{v}_A$  érintővektort.

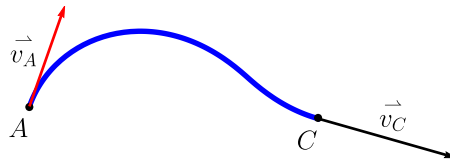
A bemutatott eljárás pszeudokódját a Függelékben található 6. kódrészlet tartalmazza. Az új felező módszer meghatározza a megfelelő  $\alpha$  értéket úgy, hogy az előállított kiterjesztett log-aesthetic görbeszelet kezdőpontbeli érintővektora ne csak irányában, hanem a hosszával is megegyezzen az előre megadott  $\vec{v}_A$  vektorral (33. ábra).



(a)  $|\vec{v}_A| = 15$



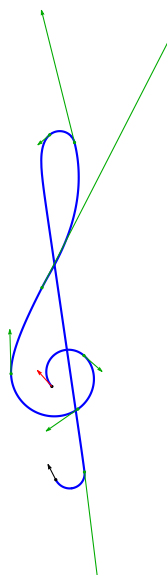
(b)  $|\vec{v}_A| = 25$



(c)  $|\vec{v}_A| = 35$

33. ábra. Különböző  $v_A$  hosszal meghatározott log-aesthetic görbék. Ezen konfiguráció első vektorának elméleti maximum hosszúsága  $\approx 40.8$  egység.

Az új modellezési algoritmusra egy példát láthatunk a 34. ábrán. Levien megfigyelései alapján [32], betűtípusok log-aesthetic görbékkel történő tervezése jobb eredményt hozhat, mint az általános szabadformájú görbék (Bézier, B-spline) használata, mert előfordulhat, például kubikus Bézier-görbék interpolációja esetén, hogy a különböző betűtípus-variációk létrehozása esetén a görbék a  $G^1$  folytonosságot sem őrzik meg.



34. ábra. Violinkulcs tervezése a bemutatott algoritmussal. A kiterjesztett log-aesthetic görbeszegmensek  $G^2$  folytonosan csatlakoznak a zöld pontokban (az érintővektor és a görbület is megegyezik). A görbeszeletek az ábrán látható pontok szerkesztésével, az azokból induló irányvektorok (fekete, zöld, piros nyilak) módosításával, illetve az első irányvektor (piros nyíl) hosszának megadásával szabályozhatóak.

## Összefoglalás

A dolgozatban az izooptikus görbékkel és felületekkel kapcsolatos kutatásainkat összegeztük. A disszertáció 1. fejezetében egy adott felülethez rendelt izooptikus görbével foglalkoztunk, mely a síkbeli izooptikus problémák térbeli általánosítása felé megtett első lépés. Az 1.2. alfejezetben részletesen kifejtettük egy speciális helyzetben adott Bézier-felület izooptikus görbéjének meghatározásának módját. A számolások három különböző lehetőségét írtuk le. Ezen új eredményeket [45]-ben publikáltuk.

A 2. fejezetben izooptikus felületekkel kapcsolatos eredményeinket mutattuk be. A 2.2.1. alfejezetben a Csima és Szirmai [9] által felírt, a látószög térbeli kiterjesztésével definiált izooptikus felületet kereső algoritmusainkat részleteztük. Három új eljárást (nyers erő, felületkitöltés, gömbi keresés) ismertettünk, melyek akár konkáv háromdimenziós modellek izooptikus felületeit is képesek hatékonyan meghatározni. Az alfejezet egy, a kamera pozíciójának kezelésére jól használható módszert is tartalmaz az izooptikus felület gömbi kereséssel történő bejárásával. Ezenfelül egy alternatív izooptikus felület definíciót ismertettünk a 2.2.2. alfejezetben, mely térbeli általánosítás alapja a látószög, mint a legszélesebb nézet értelmezése, azaz térben az adott alakzat egy nézőpontja körüli egységgömbi vetületének a legszélesebb átmérője. Ezen új eredményeket [46]-ban publikáltuk.

A 2.2.3. alfejezetben pedig bemutattuk az új, a Csima és Szirmai-féle definíción alapuló algoritmusaink GPU-ra optimalizált változatát, mely akár 100-szoros sebességnövekedést eredményez a dinamikus párhuzamosítási eljárások alkalmazása révén. Az új eredményeket [43]-ban publikáltuk.

A dolgozat utolsó részében, a 3. fejezetben a log-aesthetic görbe vizsgálatával ismét síkbeli izooptikus problémákkal foglalkoztunk. A 3.2.1. alfejezetben megadtuk az irányszöggel paraméterezett gör-

bék izooptikus görbéjének explicit alakját, továbbá megvizsgáltuk a log-aesthetic görbe autoizooptikus tulajdonságát. Ezen eredményeket tudományos folyóiratba közlésre benyújtottuk [44].

Ezen felül a 3.2.2. alfejezetben továbbfejlesztettük a Yoshida és Saito által felírt, a log-aesthetic görbével történő modellezés egyik alapvető algoritmusát, bővítve annak felhasználási lehetőségeit. Az új módszer a  $G^1$  Hermite-féle interpoláción felül képes log-aesthetic görbék  $G^2$  folytonos csatlakoztatására is. Az eredményeket tudományos folyóiratba közlésre benyújtottuk [47], a publikáció a dolgozat írásakor megjelenés alatt van.

## Summary

In the dissertation, we have summarized our research about isoptic curves and surfaces. In Chapter 1, we have presented our result about isoptic space curves, which are the very first step to extend isoptics to the three-dimension space. In Section 1 we have proposed our algorithm to find the isoptic curve in a special case for a Bézier surface through three different calculations. We have published the results in [45].

In Chapter 2, we have introduced the isoptic surfaces based on the definition of Csima and Szirmai [9]. In Section 2.2.1, we have provided new algorithms to find isoptic surfaces with three searching methods (brute-force, flood-fill and spherical search), that are able to find the isoptic surface for concave meshes as well. We have also proposed a camera tracking technique using isoptics with the third spherical search method. Furthermore, an alternative isoptic surface definition is also presented in Section 2.2.2, by considering the isoptic point as the „widest view”, i.e. the widest diameter of the projection on the unit circle around the point. We have published these results in [46].

In Section 2.2.3 we have shown a new algorithm that is optimized for the GPU to find isoptic surface up to 100 times faster than the previous versions. The new parallel approach is published in [46].

In Chapter 3 we have dealt with planar isoptics again by examining the log-aesthetic curve. In Section 3.2.1 we have provided a formula to compute the isoptic curves of tangential angle parametrized curves. We have also analyzed the autoisoptic property of the log-aesthetic curves. These results are submitted for publication [44].

In a 3.2.2. we have developed a new algorithm to design with log-aesthetic curves based on the fundamental technique of Yoshida and Saito [64], by extending its capability. The new approach solves the  $G^1$  Hermite interpolation problem and also capable of  $G^2$  joining log-

aesthetic curve segments. The results are accepted for publication [47].

## Köszönetnyilvánítás

Ezúton szeretném kifejezni hálás köszönetemet mindazoknak, akik segítettek az egyetemi tanulmányaimat és ezen dolgozat elkészítését.

Mindenekelőtt köszönöm Dr. Hoffmann Miklósnak a témavezetői iránymutatását, bizalmát és mindazt, amit az évek során tőle tanultam. Szeretnék köszönetet mondani Dr. Kunkli Rolandnak, aki az egyetemi és doktoranduszi éveimet is végig segítette.

Szeretném hálámat kifejezni a Komputergrafika és Képfeldolgozás Tanszék munkatársainak, illetve az Informatikai Kar tanárainak, mindazoknak akik bármilyen módon segítettek és támogatták az egyetemi, illetve doktori képzésem sikeres befejezését.

Továbbá szeretném megköszönni családomnak a kitartó türelmüket, támogatásukat és a sok bátorítást, amit a tanulmányaim során tőlük kaptam.

## Irodalomjegyzék

- [1] Elsa Abbena, Simon Salamon és Alfred Gray. *Modern differential geometry of curves and surfaces with Mathematica*. Third Edition. CRC press, 2017.
- [2] Jon Louis Bentley és Thomas A Ottmann. *Algorithms for reporting and counting geometric intersections*. IEEE Transactions on Computers C-28.9 (1979), 643–647. old.  
DOI: <https://doi.org/10.1109/TC.1979.1675432>.
- [3] Juan Cao és Guozhao Wang. *A note on Class A Bézier curves*. Computer Aided Geometric Design 25.7 (2008), 523–528. old.  
DOI: <https://doi.org/10.1016/j.cagd.2007.10.001>.
- [4] Zhonggui Chen és tsai. *Interpolatory curve modeling with feature points control*. Computer-Aided Design 114 (2019), 155–163. old.
- [5] Marc Christie és tsai. *Virtual Camera Planning: A Survey*. Smart Graphics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, 40–52. old.  
DOI: [https://doi.org/10.1007/11536482\\_4](https://doi.org/10.1007/11536482_4).
- [6] Waldemar Cieślak, Andrzej Miernowski és Witold Mozgawa. *Isoptics of a closed strictly convex curve*. Global Differential Geometry and Global Analysis. Springer Berlin Heidelberg, 1991, 28–35. old. ISBN: 978-3-540-46445-7.  
DOI: <https://doi.org/10.1007/BFb0083625>.
- [7] Waldemar Cieślak, Andrzej Miernowski és Witold Mozgawa. *Isoptics of a closed strictly convex curve.-II*. Rendiconti del Seminario Matematico della Università di Padova 96 (1996), 37–49. old.  
URL: [http://www.numdam.org/item/RSMUP\\_1996\\_\\_96\\_\\_37\\_0](http://www.numdam.org/item/RSMUP_1996__96__37_0).

- [8] Géza Csima és Szirmai Jenő. *Isoptic curves of conic sections in constant curvature geometries*. *Mathematical Communications* 19.2 (2014), 277–290. old.  
URL: <https://hrcak.srce.hr/129576>.
- [9] Géza Csima és Jenő Szirmai. *Isoptic surfaces of polyhedra*. *Computer Aided Geometric Design* 47 (2016), 55–60. old.  
DOI: <https://doi.org/10.1016/j.cagd.2016.03.001>.
- [10] Géza Csima és Jenő Szirmai. *On the isoptic hypersurfaces in the  $n$ -dimensional Euclidean space*. *KoG (Scientific and professional journal of Croatian Society for Geometry and Graphics)* 17.17. (2013), 53–57. old.  
URL: <https://hrcak.srce.hr/114277>.
- [11] Gerald Farin. *Class a Bézier curves*. *Computer Aided Geometric Design* 23.7 (2006), 573–581. old.  
DOI: <https://doi.org/10.1016/j.cagd.2006.03.004>.
- [12] Gerald E Farin és Gerald Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002.  
DOI: <https://doi.org/10.1.1.84.9260>.
- [13] R. T. Farouki és T. Sakkalis. *Pythagorean hodographs*. *IBM Journal of Research and Development* 34.5 (1990), 736–752. old.  
DOI: <https://doi.org/10.1147/rd.345.0736>.
- [14] Rida T. Farouki. *1. Pythagorean—Hodograph Curves in Practical Use*. *Geometry Processing for Design and Manufacturing*, 3–33. old.  
DOI: <https://doi.org/10.1137/1.9781611971668.ch1>.
- [15] Rida T Farouki. *Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable*. Springer, 2008.  
DOI: <https://doi.org/10.1007/978-3-540-73398-0>.

- [16] RP Gardner és K Verghese. *On the solid angle subtended by a circular disc*. Nuclear Instruments and Methods 93.1 (1971), 163–167. old.  
DOI: [https://doi.org/10.1016/0029-554X\(71\)90155-8](https://doi.org/10.1016/0029-554X(71)90155-8).
- [17] RU Gobithaasan, R Karpagavalli és Kenjiro T Miura. *Drawable region of the generalized log aesthetic curves*. Journal of Applied Mathematics 2013 (2013).  
DOI: 10.1155/2013/732457.
- [18] H. Gotoh és H. Yagi. *Solid angle subtended by a rectangular slit*. Nuclear Instruments and Methods 96.3 (1971), 485–486. old.  
DOI: [https://doi.org/10.1016/0029-554X\(71\)90624-0](https://doi.org/10.1016/0029-554X(71)90624-0).
- [19] Günther Greiner és Kai Hormann. *Efficient Clipping of Arbitrary Polygons*. ACM Trans. Graph. 17.2 (1998), 71–83.  
DOI: <https://doi.org/10.1145/274363.274364>.
- [20] Csima Géza. *Isoptic curves and surfaces*. Dissz. Budapest University of Technology és Economics, Mathematical Institute, 2017.
- [21] T. Harada, F. Yoshimoto és M. Moriyama. *An aesthetic curve in the field of Industrial design*. Proceedings 1999 IEEE Symposium on Visual Languages. 1999, 38–47. old.  
DOI: <https://doi.org/10.1109/VL.1999.795873>.
- [22] Toshinobu Harada. *Study of quantitative analysis of the characteristics of a curve*. Forma 12.1 (1997), 55–63. old.  
URL: <https://ci.nii.ac.jp/naid/10002200152/>.
- [23] John F. Hughes és tsai. *Computer graphics: principles and practice (3rd ed.)* Addison-Wesley Professional, 2013, 280–283. old.  
ISBN: 0321399528.

- [24] Wolfram Research, Inc. *Mathematica, Version 12.1*. Champaign, IL, 2015.  
URL: <https://www.wolfram.com/mathematica>.
- [25] Stephen Jones. *Introduction to dynamic parallelism*. Nvidia GPU Technology Conference (GTC). 2012. máj.
- [26] Tao Ju és tsai. *Dual Contouring of Hermite Data*. Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '02. 2002, 339–346.  
DOI: <https://doi.org/10.1145/566570.566586>.
- [27] Michael Kazhdan és Hugues Hoppe. *Screened Poisson Surface Reconstruction*. ACM Transactions on Graphics (TOG) 32.3 (2013), 29. old.  
DOI: <https://doi.org/10.1145/2487228.2487237>.
- [28] Lutz Kettner és Emo Welzl\*. *Contour Edge Analysis for Polyhedron Projections*. Geometric Modeling: Theory and Practice: The State of the Art. Szerk. Wolfgang Strasser, Reinhard Klein és René Rau. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, 379–394. old.  
DOI: [https://doi.org/10.1007/978-3-642-60607-6\\_25](https://doi.org/10.1007/978-3-642-60607-6_25).
- [29] Roland Kunkli, Ildikó Papp és Miklós Hoffmann. *Isoptics of Bézier curves*. Computer Aided Geometric Design 30.1 (2013), 78–84. old.  
DOI: <https://doi.org/10.1016/j.cagd.2012.05.002>.
- [30] Árpád Kurusa. *Is a convex plane body determined by an isoptic?*: Beiträge zur Algebra und Geometrie/Contributions to Algebra and Geometry 53.1 (2012), 281–294. old.  
DOI: <https://doi.org/10.1007/s13366-011-0074-2>.

- [31] Raph Levien és Carlo H. Séquin. *Interpolating splines: which is the fairest of them all?:* Computer-Aided Design and Applications 6.1 (2009), 91–102. old.  
DOI: <https://doi.org/10.3722/cadaps.2009.91-102>.
- [32] Raphael Linus Levien. *From spiral to spline: Optimal techniques in interactive curve design.* Dissz. UC Berkeley, 2009.
- [33] R. E. Levin. *Luminance — A Tutorial Paper.* Journal of the SMPTE 77.10 (1968), 1005–1011. old.  
DOI: <https://doi.org/10.5594/J13629>.
- [34] William E. Lorensen és Harvey E. Cline. *Marching Cubes: A High Resolution 3D Surface Construction Algorithm.* SIGGRAPH Computer Graphics 21.4 (1987), 163—169.  
DOI: <https://doi.org/10.1145/37402.37422>.
- [35] Gino Loria. *Spezielle algebraische und transzendente ebene kurven: theorie und geschichte.* 1. köt. Leipzig-Berlin: B.G. Teubner, 1911.
- [36] Francisco Martínez és tsai. *A simple algorithm for Boolean operations on polygons.* Advances in Engineering Software 64 (2013), 11–19. old.  
DOI: <https://doi.org/10.1016/j.advengsoft.2013.04.004>.
- [37] D.S. Meek és tsai. *Planar two-point  $G^1$  Hermite interpolating log-aesthetic spirals.* Journal of Computational and Applied Mathematics 236.17 (2012), 4485–4493. old.  
DOI: <https://doi.org/10.1016/j.cam.2012.04.021>.
- [38] Kenjiro T. Miura. *A General Equation of Aesthetic Curves and its Self-Affinity.* Computer-Aided Design and Applications 3.1–4 (2006), 457–464. old.  
DOI: <https://doi.org/10.1080/16864360.2006.10738484>.

- [39] Kenjiro T Miura és tsai. *Derivation of a general formula of aesthetic curves*. Proceedings of the Eighth International Conference on Humans and Computers (HC2005). 2005, 166–171. old.
- [40] Kenjiro T. Miura és tsai. *Designing Log-aesthetic Splines with  $G^2$  Continuity*. Computer-Aided Design and Applications 10.6 (2013), 1021–1032. old.  
DOI: <https://doi.org/10.3722/cadaps.2013.1021-1032>.
- [41] Kenjiro T. Miura és tsai. *Input of log-aesthetic curve segments with inflection end points and generation of log-aesthetic curves with  $G^2$  continuity*. Computer-Aided Design and Applications 5.1–4 (2008), 77–85. old.  
DOI: [10.3722/cadaps.2008.77-85](https://doi.org/10.3722/cadaps.2008.77-85).
- [42] Kenjiro T Miura és tsai.  *$\tau$ -curve: introduction of cusps to aesthetic curves*. Journal of Computational Design and Engineering 7.2 (2020. ápr.), 155–164. old.  
DOI: <https://doi.org/10.1093/jcde/qwaa014>.
- [43] Ferenc Nagy. *Efficiently parallelised algorithm to find isoptic surface of polyhedral meshes*. Annales Mathematicae et Informaticae 52 (2020), 167–181. old.  
DOI: <https://doi.org/10.33039/ami.2020.05.002>.
- [44] Ferenc Nagy. *Isoptics of log-aesthetic curves*. (közlésre benyújtva). 2022.  
URL: <https://arxiv.org/abs/2104.11327>.
- [45] Ferenc Nagy és Roland Kunkli. *Method for computing angle constrained isoptic curves for surfaces*. Annales Mathematicae et Informaticae 42 (2013), 65–70. old.  
URL: <http://publikacio.uni-eszterhazy.hu/2915/>.

- [46] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. *New algorithm to find isoptic surfaces of polyhedral meshes*. Computer Aided Geometric Design 64 (2018), 90–99. old.  
DOI: <https://doi.org/10.1016/j.cagd.2018.04.001>.
- [47] Ferenc Nagy, Norimasa Yoshida és Miklós Hoffmann. *Interactive  $G^1$  and  $G^2$  Hermite interpolation using coupled log-aesthetic curves*. Computer-Aided Design and Applications (2022). (megjelenés alatt).  
URL: <https://arxiv.org/abs/2105.09762>.
- [48] Naohito Nakasato. *Implementation of a parallel tree method on a GPU*. Journal of Computational Science 3.3 (2012), 132–141. old.  
DOI: <https://doi.org/10.1016/j.jocs.2011.01.006>.
- [49] NVIDIA. *NVIDIA CUDA C programming guide*. 2019.
- [50] Boris Odehnal. *Examples of Autoisoptic Curves*. ICGG 2018 - Proceedings of the 18th International Conference on Geometry and Graphics. Szerk. Luigi Cocchiarella. Cham: Springer International Publishing, 2019, 350–361. old.  
DOI: [https://doi.org/10.1007/978-3-319-95588-9\\_28](https://doi.org/10.1007/978-3-319-95588-9_28).
- [51] Francisco Puerta Romero és tsai. *The logarithmic spiral, autoisoptic curve*. Anales de Ingeniería Gráfica 21/201 (2010), 41–45. old.  
URL: <http://oa.upm.es/12402/>.
- [52] Christine Rüb. *Line-segment intersection reporting in parallel*. Algorithmica 8.1-6 (1992), 119–144. old.  
DOI: <https://doi.org/10.1007/BF01758839>.
- [53] P. Salvi, H. Suzuki és T. Várady. *Fast and Local Fairing of B-Spline Curves and Surfaces*. Advances in Geometric Modeling and Processing. Springer Berlin Heidelberg, 2008, 155–163. old.  
DOI: [https://doi.org/10.1007/978-3-540-79246-8\\_12](https://doi.org/10.1007/978-3-540-79246-8_12).

- [54] Nickolas S Sapidis. *Designing fair curves and surfaces: Shape quality in geometric modeling and computer-aided design*. SIAM, 1994.  
DOI: <https://doi.org/10.1137/1.9781611971521>.
- [55] Paul Siebeck. *Ueber eine Gattung von Curven vierten Grades, welche mit den elliptischen Functionen zusammenhängen*. Journal für die reine und angewandte Mathematik 1860.57 (1860), 359–370. old.  
DOI: <https://doi.org/10.1515/crll.1860.57.359>.
- [56] C. M. Taylor. *Note of a theory of orthoptic and isoptic loci*. Proceedings of the Royal Society of London 37.232–234 (1884), 138–141. old.  
DOI: <https://doi.org/10.1098/rsp1.1884.0024>.
- [57] Bala R. Vatti. *A Generic Solution to Polygon Clipping*. Commun. ACM 35.7 (1992. júl.), 56–63.  
DOI: <https://doi.org/10.1145/129902.129906>.
- [58] Eric W. Weisstein. *Harmonic Addition Theorem*. From *MathWorld—A Wolfram Web Resource*. Megtekintve: 2021. 01. 12.  
URL: <https://mathworld.wolfram.com/HarmonicAdditionTheorem.html>.
- [59] W. Wunderlich. *Contributions to the geometry of cam mechanisms with oscillating followers*. Journal of Mechanisms 6.1 (1971), 1–20. old.  
DOI: [https://doi.org/10.1016/0022-2569\(71\)90002-4](https://doi.org/10.1016/0022-2569(71)90002-4).
- [60] Walter Wunderlich. *Kurven mit isoptischem Kreis*. Aequationes mathematicae 6.1 (1971), 71–81. old.  
DOI: <https://doi.org/10.1007/BF01833240>.

- [61] Walter Wunderlich. *Kurven mit isoptischer Ellipse*. Monatshefte für Mathematik 75.4 (1972), 346–362. old.  
DOI: <https://doi.org/10.1007/BF01303569>.
- [62] R. C. Yates. *A Handbook on Curves and their Properties*. Ann Arbor, J.W. Edwards, 1947, 138–140. old.
- [63] Norimasa Yoshida, Tomoyuki Hiraiwa és Takafumi Saito. *Interactive Control of Planar Class A Bézier Curves using Logarithmic Curvature Graphs*. Computer-Aided Design and Applications 5.1-4 (2008), 121–130. old.  
DOI: <https://doi.org/10.3722/cadaps.2008.121-130>.
- [64] Norimasa Yoshida és Takafumi Saito. *Interactive aesthetic curve segments*. The Visual Computer 22.9 (2006), 896–905. old.  
DOI: <https://doi.org/10.1007/s00371-006-0076-5>.
- [65] Norimasa Yoshida és Takafumi Saito. *The Evolutes of Log-Aesthetic Planar Curves and the Drawable Boundaries of the Curve Segments*. Computer-Aided Design and Applications 9.5 (2012), 721–731. old.  
DOI: <https://doi.org/10.3722/cadaps.2012.721-731>.
- [66] Rushan Ziatdinov, Norimasa Yoshida és Tae wan Kim. *Analytic parametric equations of log-aesthetic curves in terms of incomplete gamma functions*. Computer Aided Geometric Design 29.2 (2012), 129–140. old.  
DOI: <https://doi.org/10.1016/j.cagd.2011.11.003>.

# Publikációs lista

## Referált folyóiratcikkek/Refereed journal papers

- [1] Ferenc Nagy. *Isoptics of log-aesthetic curves*. (közlésre benyújtva). 2022.  
URL: <https://arxiv.org/abs/2104.11327>.
- [2] Ferenc Nagy, Norimasa Yoshida és Miklós Hoffmann. „Interactive  $G^1$  and  $G^2$  Hermite interpolation using coupled log-aesthetic curves”. *Computer-Aided Design and Applications* (2022). (megjelenés alatt).  
URL: <https://arxiv.org/abs/2105.09762>.
- [3] Ferenc Nagy. „Efficiently parallelised algorithm to find isoptic surface of polyhedral meshes”. *Annales Mathematicae et Informaticae* 52 (2020), 167–181. old.  
DOI: <https://doi.org/10.33039/ami.2020.05.002>.
- [4] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. „New algorithm to find isoptic surfaces of polyhedral meshes”. *Computer Aided Geometric Design* 64 (2018), 90–99. old.  
DOI: <https://doi.org/10.1016/j.cagd.2018.04.001>.
- [5] Ferenc Nagy és Roland Kunkli. „Method for computing angle constrained isoptic curves for surfaces”. *Annales Mathematicae et Informaticae* 42 (2013), 65–70. old.  
URL: <http://publikacio.uni-eszterhazy.hu/2915>.

## Könyvrészletek/Conference papers

- [1] Ferenc Nagy, Norimasa Yoshida és Miklós Hoffmann. „Interactive  $G^1$  and  $G^2$  Hermite interpolation using coupled log-aesthetic

curves”. *Proceedings of CAD’21*. Szerk. Roberto Raffaeli és Les Piegl. Barcelona: CAD Solutions LLC, 2021, 329–333. old.  
DOI: <https://doi.org/10.14733/cadconfP.2021.329-333>.

- [2] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. „Isoptics - Camera orbit around convex meshes”. *VIII. Magyar Számítógépes Grafika és Geometria Konferencia*. Szerk. László Szirmay-Kalos és Gábor Renner. Budapest: Neumann János Számítógéptudományi Társaság, 2016, 48–52. old. ISBN: 9786155036118.

## Konferenciaelőadások/Conference talks

- [1] Ferenc Nagy és Norimasa Yoshida. „Interactive modeling with log-aesthetic spirals”. *11th International Conference on Applied Informatics (ICAI 2020)*. Eger, 2020.
- [2] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. „Improved algorithm to find isoptic surfaces for three-dimensional objects”. *10th International Conference on Applied Informatics (ICAI 2017)*. Eger, 2017.
- [3] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. „New algorithms to find isoptic surfaces for three-dimensional objects”. *Conference on Geometry – Theory and Applications (CGTA 2017)*. Pilzen, Csehország, 2017.
- [4] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. „Térbeli alakzat körüli izooptikus felületet kereső algoritmus hatékony implementációja grafikus kártyán”. *XXXIII. OTDK Informatika Tudományi Szekció*. Veszprém, 2017.
- [5] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. „Fast Algorithm for Finding Special Isoptic Curve of Bézier Surfaces”. *17th Scientific-Professional Colloquium on Geometry and Graphics*. Rastoke, Horvátország, 2013.

- [6] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. „Izooptikus görbéken alapuló módszer kamera mozgatási pályájának optimalizálására”. *XXXI. OTDK Informatika Tudományi Szekció*. Budapest, 2013.
- [7] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. „Method for optimization of camera movement path based on isoptic curves”. *Conference on Geometry – Theory and Applications (CG-TA 2013)*. Ljubljana, Szlovénia, 2013.

## **Poszterprezentációk/Poster presentations**

- [1] Ferenc Nagy. „Simple and efficiently parallelised algorithm to find isoptic surface of polyhedral meshes”. *Graphics and Application : the 12th Asian Forum on Graphic Science (AFGS 2019)*. Kunming, Kína, 2019.
- [2] Ferenc Nagy, Roland Kunkli és Miklós Hoffmann. „Algorithms to find isoptic surfaces of polyhedral meshes”. *5th Winter School of PhD Students in Informatics and Mathematics*. Debrecen, 2018.



Nyilvántartási szám: DEENK/51/2022.PL  
Tárgy: PhD Publikációs Lista

Jelölt: Nagy Ferenc  
Doktori Iskola: Informatikai Tudományok Doktori Iskola  
MTMT azonosító: 10058591

## A PhD értekezés alapjául szolgáló közlemények

### Idegen nyelvű tudományos közlemények hazai folyóiratban (2)

1. **Nagy, F.:** Efficiently parallelised algorithm to find isoptic surface of polyhedral meshes.  
*Ann. Math. Inform.* 51, 1-15, 2020. ISSN: 1787-5021.  
DOI: <http://dx.doi.org/10.33039/ami.2020.05.002>
2. **Nagy, F., Kunkli, R.:** Method for computing angle constrained isoptic curves for surfaces.  
*Ann. Math. et Inf.* 42, 65-70, 2013. ISSN: 1787-5021.

### Idegen nyelvű tudományos közlemények külföldi folyóiratban (2)

3. **Nagy, F., Yoshida, N., Hoffmann, M.:** Interactive G1 and G2 Hermite Interpolation Using Coupled Log-aesthetic Curves.  
*Comput Aided Des Appl. "Accepted by Publisher"* (-), 1-20-, 2022. ISSN: 1686-4360.
4. **Nagy, F., Kunkli, R., Hoffmann, M.:** New algorithm to find isoptic surfaces of polyhedral meshes.  
*Comput. Aided Geom. Des.* 64, 90-99, 2018. ISSN: 0167-8396.  
DOI: <http://dx.doi.org/10.1016/j.cagd.2018.04.001>  
IF: 1.421

### Idegen nyelvű konferencia közlemények (2)

5. **Nagy, F., Yoshida, N., Hoffmann, M.:** Interactive G1 and G2 Hermite Interpolation Using Coupled Log-aesthetic Curves.  
In: Proceedings of CAD'21. Ed.: Roberto Raffaeli, Les Piegl, CAD Solutions LLC, Barcelona, 329-333, 2021, (ISSN 2769-8440)
6. **Nagy, F., Kunkli, R., Hoffmann, M.:** Isoptics - Camera orbit around convex meshes.  
In: VIII. Magyar Számítógépes Grafika és Geometria Konferencia. Szerk.: Szirmai-Kalós László, Renner Gábor, Neumann János Számítógép-tudományi Társaság, Budapest, 48-52, 2016. ISBN: 9786155036118





## Magyar nyelvű absztrakt kiadványok (2)

7. **Nagy, F.:** Térbeli alakzat körüli izooptikus felületet kereső algoritmus hatékony implementációja grafikus kártyán.  
In: XXXIII. OTDK Informatika Tudományi Szekció. Szerk.: Bálint Roland, Vassányi István, Pannon Egyetem Műszaki Informatikai Kar, Veszprém, 74, 2017. ISBN: 789633960981
8. **Nagy, F.:** Izooptikus görbéken alapuló módszer kamera mozgatható pályájának optimalizálására.  
In: XXXI. OTDK Informatika Tudományi Szekció : Konferencia kiadvány. Szerk.: Bognár Géza, Gábor Dénes Főiskola, Budapest, 113, 2013. ISBN: 9789630859004

## Idegen nyelvű absztrakt kiadványok (7)

9. **Nagy, F., Yoshida, N.:** Interactive modeling with Log-aesthetic spirals.  
In: 11th International Conference on Applied Informatics (ICAI 2020), CEUR, Eger, 1, 2020.
10. **Nagy, F.:** Simple and Efficiently Parallelised Algorithm to Find Isoptic Surface of Polyhedral Meshes.  
In: Graphics and Application : the 12th Asian Forum on Graphic Science (AFGS 2019). Ed.: Baoling Han, Xiao Luo, Hongliang Fan, Beijing Institute of Technology Press, Beijing, 133-134, 2019. ISBN: 9787893910319
11. **Nagy, F., Kunkli, R., Hoffmann, M.:** Algorithms to find isoptic surfaces of polyhedral meshes.  
In: 5th Winter School of PhD Students in Informatics and Mathematics. Ed.: Hudoba Péter, Doktoranduszok Országos Szövetsége, Budapest, 36, 2018. ISBN: 9786155586231
12. **Nagy, F., Kunkli, R., Hoffmann, M.:** Improved algorithm to find isoptic surface for three-dimensional objects.  
In: ICAI 2017: 10th International Conference on Applied Informatics, Eszterházy Károly Egyetem, Eger, 1, 2017.
13. **Nagy, F., Kunkli, R., Hoffmann, M.:** New algorithms to find isoptic surfaces for three-dimensional objects.  
In: Conference on Geometry: Theory and Applications 2017. Book of abstracts. Ed.: M. Lávička, Vydavatelský servis, Pilsen, 58, 2017. ISBN: 9788086843568
14. **Nagy, F., Kunkli, R., Hoffmann, M.:** A fast algorithm for finding special isoptic curve of Bézier surfaces.  
In: 17th Scientific-Professional Colloquium on Geometry and Graphic : Abstracts. Ed.: Tomislav Doslic, Ema Jurkin, Croatian Society for Geometry and Graphics, Zagreb, 23-23, 2013.





15. Hoffmann, M., Kunkli, R., **Nagy, F.**: Method for Optimization of Camera Movement Path Based on Isoptic Curves.

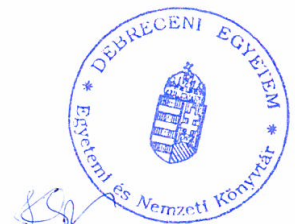
In: Conference on Geometry : Theory and Applications : Book of Abstracts. Eds.: B. Kovac, T. Kanduc, E. Zagar, University of Ljubljana, Ljubljana, 35-36, 2013.

**A közlő folyóiratok összesített impakt faktora: 1,421**

**A közlő folyóiratok összesített impakt faktora (az értekezés alapjául szolgáló közleményekre):  
1,421**

A DEENK a Jelölt által az iDEa Tudóstérbe feltöltött adatok bibliográfiai és tudományometriai ellenőrzését a tudományos adatbázisok és a Journal Citation Reports Impact Factor lista alapján elvégezte.

Debrecen, 2022.01.20.





Registry number: DEENK/51/2022.PL  
Subject: PhD Publication List

Candidate: Ferenc Nagy  
Doctoral School: Doctoral School of Informatics  
MTMT ID: 10058591

### List of publications related to the dissertation

#### Foreign language scientific articles in Hungarian journals (2)

1. **Nagy, F.:** Efficiently parallelised algorithm to find isoptic surface of polyhedral meshes.  
*Ann. Math. Inform. 51*, 1-15, 2020. ISSN: 1787-5021.  
DOI: <http://dx.doi.org/10.33039/ami.2020.05.002>
2. **Nagy, F., Kunkli, R.:** Method for computing angle constrained isoptic curves for surfaces.  
*Ann. Math. et Inf. 42*, 65-70, 2013. ISSN: 1787-5021.

#### Foreign language scientific articles in international journals (2)

3. **Nagy, F., Yoshida, N., Hoffmann, M.:** Interactive G1 and G2 Hermite Interpolation Using Coupled Log-aesthetic Curves.  
*Comput Aided Des Appl. "Accepted by Publisher" (-)*, 1-20-, 2022. ISSN: 1686-4360.
4. **Nagy, F., Kunkli, R., Hoffmann, M.:** New algorithm to find isoptic surfaces of polyhedral meshes.  
*Comput. Aided Geom. Des. 64*, 90-99, 2018. ISSN: 0167-8396.  
DOI: <http://dx.doi.org/10.1016/j.cagd.2018.04.001>  
IF: 1.421

#### Foreign language conference proceedings (2)

5. **Nagy, F., Yoshida, N., Hoffmann, M.:** Interactive G1 and G2 Hermite Interpolation Using Coupled Log-aesthetic Curves.  
In: Proceedings of CAD'21. Ed.: Roberto Raffaeli, Les Piegl, CAD Solutions LLC, Barcelona, 329-333, 2021, (ISSN 2769-8440)
6. **Nagy, F., Kunkli, R., Hoffmann, M.:** Isoptics - Camera orbit around convex meshes.  
In: VIII. Magyar Számítógépes Grafika és Geometria Konferencia. Szerk.: Szirmay-Kalos László, Renner Gábor, Neumann János Számítógép-tudományi Társaság, Budapest, 48-52, 2016. ISBN: 9786155036118





Hungarian abstracts (2)

7. **Nagy, F.**: Térbeli alakzat körüli izooptikus felületet kereső algoritmus hatékony implementációja grafikus kártyán.  
In: XXXIII. OTDK Informatika Tudományi Szekció. Szerk.: Bálint Roland, Vassányi István, Pannon Egyetem Műszaki Informatikai Kar, Veszprém, 74, 2017. ISBN: 789633960981
8. **Nagy, F.**: Izooptikus görbéken alapuló módszer kamera mozgatási pályájának optimalizálására.  
In: XXXI. OTDK Informatika Tudományi Szekció : Konferencia kiadvány. Szerk.: Bognár Géza, Gábor Dénes Főiskola, Budapest, 113, 2013. ISBN: 9789630859004

Foreign language abstracts (7)

9. **Nagy, F.**, Yoshida, N.: Interactive modeling with Log-aesthetic spirals.  
In: 11th International Conference on Applied Informatics (ICAI 2020), CEUR, Eger, 1, 2020.
10. **Nagy, F.**: Simple and Efficiently Parallelised Algorithm to Find Isoptic Surface of Polyhedral Meshes.  
In: Graphics and Application : the 12th Asian Forum on Graphic Science (AFGS 2019). Ed.: Baoling Han, Xiao Luo, Hongliang Fan, Beijing Institute of Technology Press, Beijing, 133-134, 2019. ISBN: 9787893910319
11. **Nagy, F.**, Kunkli, R., Hoffmann, M.: Algorithms to find isoptic surfaces of polyhedral meshes.  
In: 5th Winter School of PhD Students in Informatics and Mathematics. Ed.: Hudoba Péter, Doktoranduszok Országos Szövetsége, Budapest, 36, 2018. ISBN: 9786155586231
12. **Nagy, F.**, Kunkli, R., Hoffmann, M.: Improved algorithm to find isoptic surface for three-dimensional objects.  
In: ICAI 2017: 10th International Conference on Applied Informatics, Eszterházy Károly Egyetem, Eger, 1, 2017.
13. **Nagy, F.**, Kunkli, R., Hoffmann, M.: New algorithms to find isoptic surfaces for three-dimensional objects.  
In: Conference on Geometry: Theory and Applications 2017. Book of abstracts. Ed.: M. Lávička, Vydavatelský servis, Pilsen, 58, 2017. ISBN: 9788086843568
14. **Nagy, F.**, Kunkli, R., Hoffmann, M.: A fast algorithm for finding special isoptic curve of Bézier surfaces.  
In: 17th Scientific-Professional Colloquium on Geometry and Graphic : Abstracts. Ed.: Tomislav Doslic, Ema Jurkin, Croatian Society for Geometry and Graphics, Zagreb, 23-23, 2013.





15. Hoffmann, M., Kunkli, R., **Nagy, F.**: Method for Optimization of Camera Movement Path Based on Isoptic Curves.

In: Conference on Geometry : Theory and Applications : Book of Abstracts. Eds.: B. Kovac, T. Kanduc, E. Zagar, University of Ljubljana, Ljubljana, 35-36, 2013.

**Total IF of journals (all publications): 1,421**

**Total IF of journals (publications related to the dissertation): 1,421**

The Candidate's publication data submitted to the iDEa Tudóstér have been validated by DEENK on the basis of the Journal Citation Report (Impact Factor) database.

20 January, 2022



# Függelék

## A. Kódrészletek

---

```
float CALCULATE_SOLID_ANGLE(P[3], faces)
  spherical_edge S[max_S] //becsült max_S
  /*I. Élek vetítése az egységömbre.*/
  n_S = PROJECT_EDGES(*S, faces)
  /*II. Metszéspontok meghatározása és gömbi szögek kiszámítása*/
  CALCULATE_INTERSECTIONS(S, n_S)
  /*III. Kezdő él megtalálása*/
  float min = float_max
  spherical_edge f, first
  foreach spherical_edge e of S do
    if (e.dist_from_center < min AND IS_VISIBLE(e))
      f = e
      min = e.dist_from_center
    endif
  endforeach
  min = 0
  spherical_edge C_fB //from the antipode of C to f.B
  foreach index i of f.intersection_id do
    if ((I[i].dist_from_a=length(f)) AND // it is at f.B
        (spherical_angle(Cf_B, S[I[i].other_edge])>min))
      first = S[I[i].other_edge]
      min = spherical_angle(Cf_B, S[I[i].other_edge])
    endif
  endforeach
  /*IV. Térszögek összegének kiszámolása*/
  spherical_edge current_edge = first
  integer n = 0 //number of traversed edges
  float theta, min_dist = 0
  repeat
    integer int_id
    float min_angle = 0, max_dist = 2 // Euclidean distance
```

```

foreach index i of current_edge.intersection_id do
  if (((I[i].dist_from_a > min_dist) OR
      ((I[i].dist_from_a = min_dist) AND
       (I[i].angle > min_angle))) AND
      (I[i].dist_from_a < max_dist))
    int_id = i
    max_dist = I[i].dist_from_a
    min_angle = I[i].angle
  endif
endforeach
theta = theta + I[i].angle
min_dist = I[n_S * (int_id%n_S) + (int_id/n_S)].dist
current_edge = S[I[int_id].other_edge]
n = n + 1
until (current_edge != first)
return (theta - ((n - 2) * PI))
end CALCULATE_SOLID_ANGLE

```

---

4. kódrészlet. GPU-ra optimalizált, további párhuzamos függvényhívásokat tartalmazó térszögszámoló eljárás.

---

```

float lambda_bisection( double alpha, int max_iteration ) {
  double lmin = 0 , lmax = 1, lambda, diff, theta_compare ;
  int i = 0 ;
  bool enlarge, beyond_inf_point = false ;
  double point[2] ; /* 2D pont A' és B' tárolására */
  if ( ( alpha <= 1 && !swap_flag ) || ( alpha >= 1 && swap_flag
  ) ) {
    theta_compare = calculate_theta_A_dash(lambda) ;
  } else {
    theta_compare = calculate_theta_C_dash(lambda) ;
  }
  if ( alpha == 1 ) enlarge = true ;
  else if ( alpha <= 1 ) {
    lmax = 1 / ( theta_Δ * (1-alpha) ) ;

```

```

        if ( theta_A < calculate_theta_A_dash(lmax) )
beyond_inf_point = true ;
        point[2] = calculate_intersection_point(theta_Δ) ; /*B'*/
    } else {
        lmax = 2 / ( -theta_Δ * (1-alpha) ) ;
        point[2] = calculate_curve_point(-theta_Δ) ; /*B'*/
    }
lambda = ( lmin + lmax ) * 0.5 ;
do {
    if ( alpha <= 1 )
        diff = theta_compare - calculate_theta_A_dash(lambda) ;
    else
        diff = theta_compare - calculate_theta_C_dash(lambda) ;
    if ( diff < EPS ) return lambda ; /* sikeres visszatérés */
    if ( ( 0 <= alpha && alpha <= 1 ) ||
        ( alpha < 1 && !beyond_inf_point ) ) {
        if ( diff < 0 ) {
            if ( enlarge ) lmax = lmax * 10 ;
            lmin = lambda ;
            lambda += (lmax - lambda) * 0.5 ;
        } else {
            enlarge = false ;
            lmax = lambda ;
            lambda -= (lambda - lmin) * 0.5 ;
        }
    }
} else if ( alpha < 1 && beyond_inf_point ) {
    if ( diff < 0 && point[0] > 0 ) {
        lmax = lambda ;
        lambda -= (lambda - lmin) * 0.5 ;
    } else {
        lmin = lambda
        lambda += (lmax - lambda) * 0.5 ;
    }
} else { // ( 1 < alpha )
    if ( diff < 0 || point[1] < 0 ) {
        lmax = lambda ;
        lambda -= (lambda - lmin) * 0.5 ;
    } else {

```

```

        lmin = lambda ;
        lambda += (lmax - lambda) * 0.5 ;
    }
}
++i ;
} while ( i < max_iteration ) ;
return -1 ; /* nincs találat */
}

```

---

5. kódrészlet. Az új, kiterjesztett log-aesthetic görbét interpoláló  $\Lambda$  felező módszer. A Yoshida-féle algoritmus [64] megfelelő módosításait piros színnel emeltük ki. A szögek helyett skaláris szorzattal számoltunk.

---

```

float alpha_bisection( double length, int max_iteration ) {
    double amin = -999 , amax = 999; /* alpha intervallum: */
    double alpha = ( amin + amax ) * 0.5 ;
    int i = 0 ;
    do {
        lambda_bisection( alpha, max_iteration ) ;
        double diff = length - Calculate_actual_length() ;
        if ( diff < EPS ) return alpha ; /* sikeres visszatérés */
        if ( !swap_flag ) {
            if ( diff < 0 ) {
                amin = alpha ;
                alpha += ( ( amax - alpha ) * 0.5 ) ;
            } else {
                amax = alpha;
                alpha -= ( ( alpha - amin ) * 0.5 ) ;
            }
        } else {
            if ( instance_1 ) {
                /* a görbeszegmens inflexiós pontot tartalmazhat */
                if ( alpha < 1 + EPSILON && beyond_inf_point ) {
                    /* fordított intervallumválasztás */

```

```

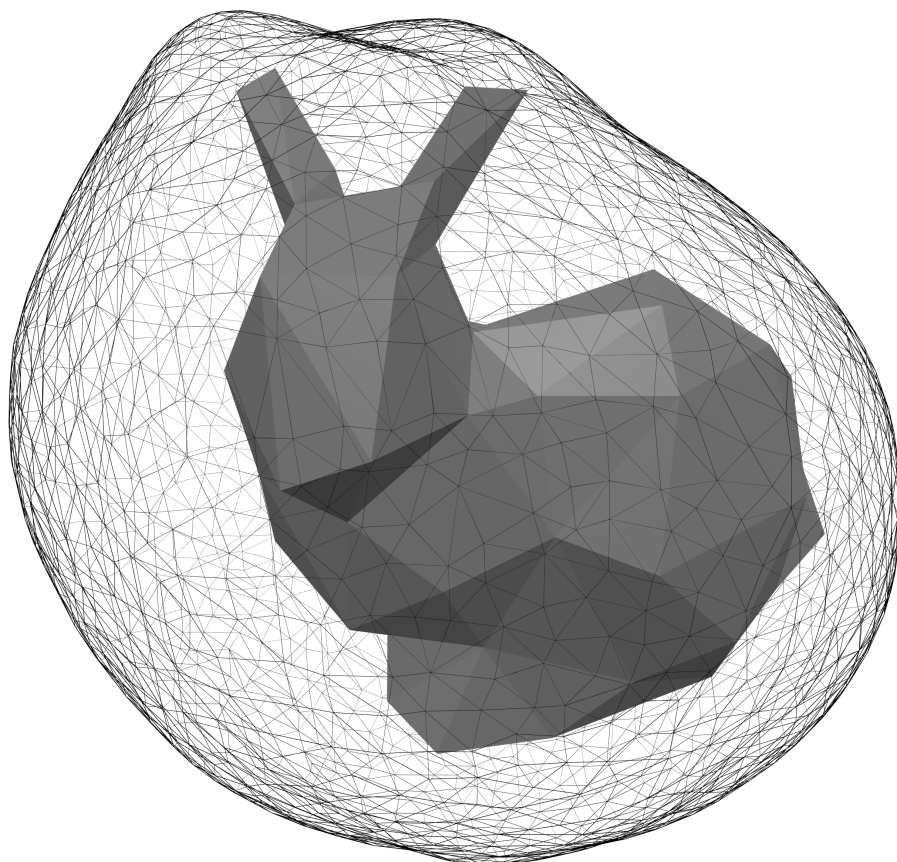
        if ( diff > 0 ) {
            amin = alpha ;
            alpha += ( ( amax - alpha ) * 0.5 ) ;
        } else {
            amax = alpha;
            alpha -= ( ( alpha - amin ) * 0.5 ) ;
        } continue ;
    } else { /* felesleges intervallumok kihagyása */
        amax = alpha;
        alpha -= ( ( alpha - amin ) * 0.5 ) ;
        continue ;
    }
}
if ( instance_2 ) {
    /* görbeszegmens csúcspontot tartalmazhat */
    if ( alpha < 1 + EPSILON && beyond_inf_point ) {
        /* felesleges intervallumok kihagyása */
        amin = alpha ;
        alpha += ( ( amax - alpha ) * 0.5 ) ;
        continue ;
    }
}
}
++i ;
} while ( i < max_iteration ) ;
return 0 ; /* nincs találat */
}

```

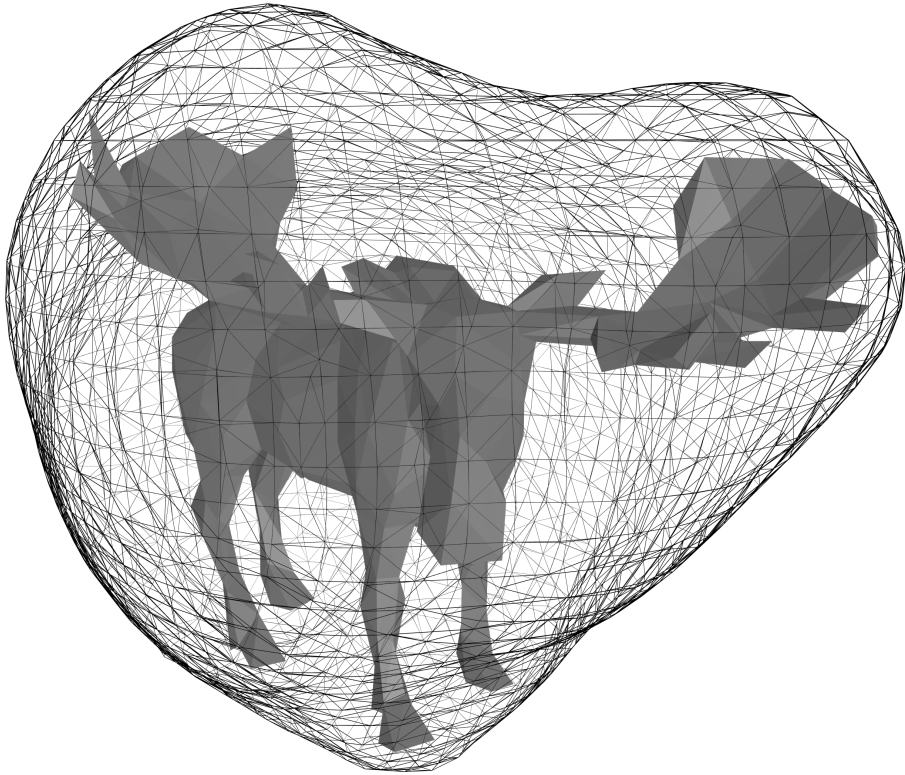
---

6. kódrészlet. Az  $\alpha$  felező módszer, mely meghatározza a megfelelő paramétert, hogy a generált kiterjesztett log-aesthetic görbe kezdőpontjának irányvektora illeszkedjen a megadott  $\vec{v}_A$  vektorra.

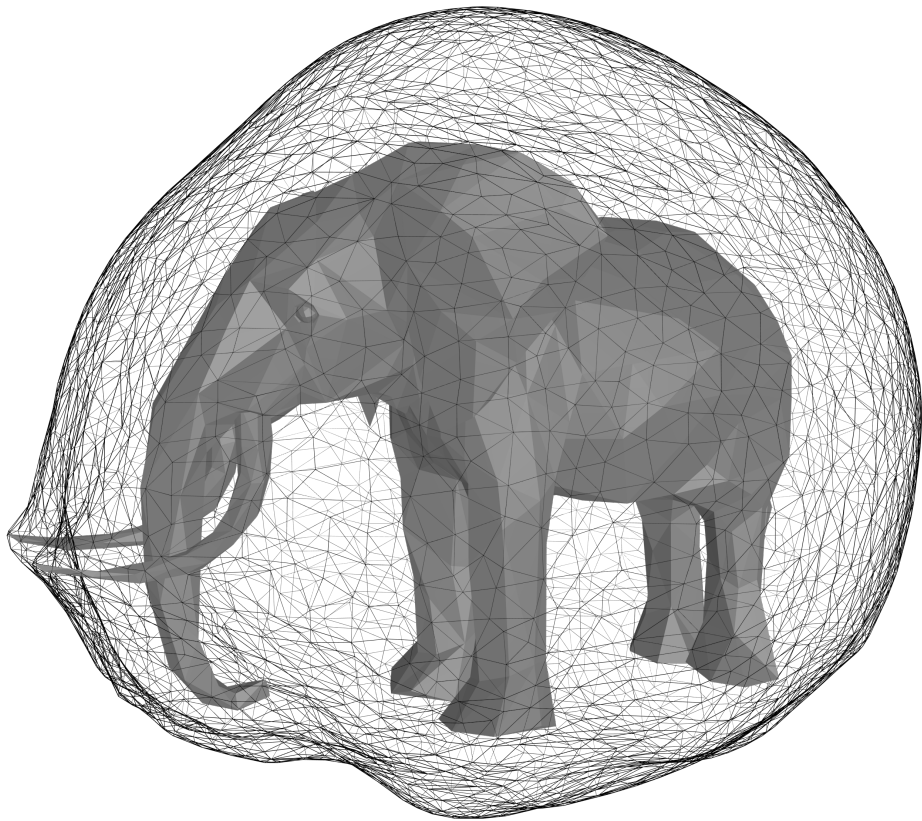
## B. Különböző részletességű modellek izooptikus felületei



35. ábra. Stanford Bunny modell (128 lap) és a pontfelhőből előállított izooptikus felülete ( $\alpha = \frac{\pi}{2}$ , modell forrás: [graphics.stanford.edu/data/3Dscanrep](http://graphics.stanford.edu/data/3Dscanrep)).



36. ábra. A szarvas modell (747 lap) és a pontfelhőből előállított izo-  
optikus felülete ( $\alpha = \frac{\pi}{2}$ , modell forrás: [www.cadnav.com](http://www.cadnav.com)).



37. ábra. Az elefánt modell (1492 lap) és a pontfelhőből előállított izooptikus felülete ( $\alpha = \frac{\pi}{2}$ , modell forrás: [graphics.stanford.edu/data/3Dscanrep](http://graphics.stanford.edu/data/3Dscanrep), [www.cadnav.com](http://www.cadnav.com)).