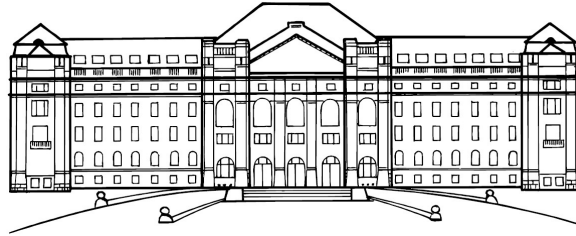


TDK-dolgozat

Bana Kornél

Kruppa Kinga

Debreceni Egyetem



Informatikai Kar
Komputergrafika és Képfeldolgozás Tanszék

**KSpheres – Hatékony algoritmus
gömbsorozatokat interpoláló felületek
csatlakoztatási problémáinak
kiküszöbölésére**

Témavezetők:

Dr.Hoffmann Miklós
főiskolai tanár
Kunkli Roland
egyetemi tanársegéd

Szerzők:

Bana Kornél
III. évfolyam
Kruppa Kinga
III. évfolyam

Debrecen, 2012

Tartalomjegyzék

1	Bevezetés	5
2	Előzmények	7
2.1	Kiindulási módszer	7
2.2	Ismert módszerek	9
2.2.1	Slabaugh-féle módszer	10
2.2.2	Burkolás a Spore TM szoftverben	10
2.2.3	Hiányosságok a felületdarabok illesztésekor	11
3	Szoftveres implementáció	13
4	Az új csatlakoztatási módszerek	19
4.1	Csatlakoztatás két dimenzióban	19
4.2	A csatlakoztatás kezelése szélső gömbökkel	20
4.3	A szomszédos gömbökre érzékeny módszer	21
4.4	A megoldást szolgáltató eljárás	23
4.5	Meglévő módszerekkel való összehasonlítás	29
5	Összefoglalás	35
6	Köszönetnyilvánítás	37
7	Irodalomjegyzék	38
8	Saját munka leírása	39

1. Bevezetés

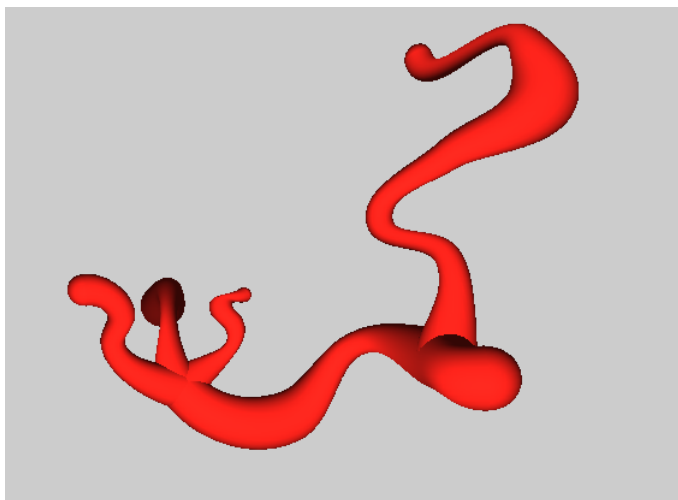
A geometriai adatok interpolációja központi fontosságú a számítógépes geometriai modellezésben. Napjainkban egyre inkább igény mutatkozik arra, hogy valamilyen formában kiterjesszük az eddig ismert, pontokon alapuló módszereket más típusú geometriai objektumok esetére is, például körökre, illetve gömbökre. Egy adott körsorozathoz két olyan görbe megtalálása a cél, melyek a köröket rendre érintik egy-egy pontban, továbbá a görbétől elvárjuk, hogy kellően simák legyenek, vizuálisan kielégítő eredményt szolgáltatassanak. A térre általánosítva pedig egy gömbsorozatot burkoló felület definiálása a cél.

Az ilyen alapokon működő felületmodellezés már megjelent a komputergrafikában, az utóbbi pár évben több folyóiratcikk is született a témában ([3], [4], [5]). Népszerű modellező szoftverekben (pl. ZBrush, SporeTM) szintén tetten érhető a jelenség, melyekben alapeszköz szinten használhatunk az említetteket megvalósítani kívánó megoldásokat. Az ezekben felhasznált eljárások azonban nem minden esetben szolgáltatnak elvárható minőségű eredményt. Sokszor a kapott kimenet nem esztétikus, s már körsorozatok esetén is szembetűnő különbségek vannak a különböző módszerek között.

A meglévő alakzatok egymáshoz való illesztése a felületmodellezés igen fontos része, hiszen ekkor a kiindulási alapként szolgáló objektumoknál bonyolultabb alakzatok jöhetnek létre. Emiatt a tervezők, felhasználók részéről folyamatos igény mutatkozik a minél kifinomultabb és számukra a tervezés folyamatában nagyobb szabadságot adó módszerek irányába. A téma aktualitását jól jelzi az is, hogy másodrendű felületek esetén sem triviális a csatlakoztatás problémájának a megoldása, mellyel egy idén készült cikk is foglalkozik [2].

Kutatásunk témája tehát különböző alakzatok, felületek előállítása megadott gömbök burkolásával és így előállított burkolófelületek összekapcsolásával. Kitűzött célunk, hogy a jelenleg ismerteknél lényegesen jobb algoritmust adjunk olyan értelemben, hogy segítségével topológiaiag bonyolultabb felületeket, élesebb elágazásokat, rendezetlen gömbhal-mazokat is kezelni tudjunk, illetve pótoljuk az illesztés terén tapasztalt hiányosságokat.

Ezen munka első fázisaként az elágaztatás, illetve csatlakoztatás problémáira sikerült hatékonynak bizonyuló módszert kifejlesztenünk, amelyet a továbbiakban részletesen ismertetünk.



1. ábra. Kutatásunk eredménye a térbeli csatlakoztatás megvalósítása

Eredményeink gyakorlatban történő alkalmazásának bemutatására egy saját fejlesztű, projektív geometriai alapokra [1] építkező szoftvert készítettünk. Ezzel a rendszerrel azonban nemcsak a meglévő eredmények reprezentálása volt a célunk, hanem egy olyan interaktív, felhasználóbarát eszközökkel felszerelt modellező szoftver elkészítése, amellyel különböző alakzatokat, felületeket tud a felhasználó előállítani tetszőlegesen megadott gömbök burkolásával és így előállított burkolófelületek összekapcsolásával.

2. Előzmények

2.1. Kiindulási módszer

Kutatásunk célja az interpoláció során nyert különböző burkolófelület-darabok egymáshoz történő csatlakoztatásának algoritmizálása, ehhez viszont elengedhetetlen, hogy foglalkozunk magával a kör- és gömbsorozatok interpolációjával. Ezen alfejezetben a munkánk kiindulási alapjául szolgáló cikket [3] foglaljuk röviden össze, s az ebben használt jelölésrendszert fogjuk a továbbiakban használni.

A jelzett munkában pontosan meg van határozva, milyen körsorozatok tekinthetők elfogadható kiindulási alapnak a probléma szempontjából, mi is ilyen sorozatokat fogunk tekinteni a későbbiekben:

2.1. definíció. Körök egy $C = \{c_1, c_2, c_3, \dots, c_n\}$ ($n \in \mathbb{N}$) sorozatát *elfogadható konfigurációjúnak* nevezzük, ha teljesíti az alábbi feltételeket (d_i a c_i által meghatározott zárt körlemez jelöli):

- $d_i \not\subseteq \bigcup_{j=1, j \neq i}^n d_j, \quad i \in \{1, 2, \dots, n\}$
- $d_i \cap d_j = \emptyset, i, j \in \{1, 2, \dots, n\}, j \notin \{i-2, i-1, i, i+1, i+2\}$
- ha $d_{i-1} \cap d_{i+1} \neq \emptyset$, akkor $d_{i-1} \cap d_{i+1} \subset d_i$

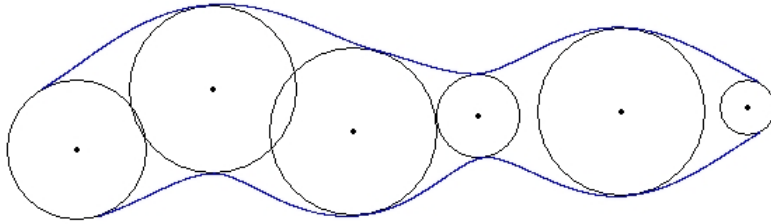
(Teljesen analóg módon vezethetjük be az elfogadható felépítésű gömbsorozat fogalmát is, amivel a térbeli esetben dolgozni fogunk.)

A fenti definíciónak megfelelő körsorozat esetén tehát a feladat két G^1 -folytonos görbe (legyen $s_1(t)$ és $s_2(t)$) konstruálása, melyek a köröket egy-egy pontban érintik. Ehhez a körsorozat minden c_i eleméhez szükséges két érintési pont (p_i és $\tilde{p}_i$), illetve két érintővektor (v_i és $\tilde{v}_i$) előállítás.

Az érintési pontok meghatározásához tekintsünk három, a körsorozatban egymást követő kört, melyekre vonatkozóan szeretnénk megoldani az Apollóniusz-problémát, azaz keressük azt a cikkben említett típusú két kört, amelyek érintik a tekintett körök mind-egyikét. A három sorozatbeli kör közül a középső (c_i) esetén az a két pont lesz a keresett p_i és $\tilde{p}_i$, melyben az adott kört a két említett, Apollóniusz-feladatra megoldást adó kör

érinti. Ezen érintési pontok fogják meghatározni a burkológörbét, azonban nem mindegy, hogy a görbepár melyik görbét definiálja egy-egy ilyen pont, s emiatt szükséges az érintési pontok csoportosítása. A szeparálás alapja egy kör esetén az azt megelőző és a rákövetkező kör középpontja, illetve a három kör hatványpontja. Mivel az érintési pontok és a hatványpont kollineárisak minden egyes körre vonatkozóan, így a csoportosítás egyértelműen megadható.

Az i . érintővektorok iránya az érintési pontból a c_i középpontjába mutató vektor elforgatottjával határozható meg, nagyságuk pedig az érintési pontoknak az i . és az $i + 1$. kör hatványvonalától mért távolságának függvénye. Így tehát a görbepár leírható a fenti módon meghatározott alapadatokból származtatott, klasszikus Hermite-ívek (2 pont, 2 érintővektor) csatlakoztatásaként, a módszer eredményeként pl. az alábbi kimenet kapható meg:

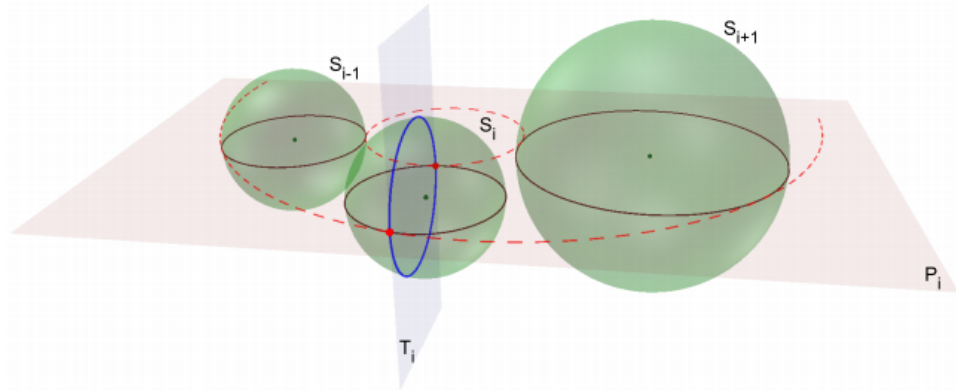


2. ábra. Kétdimenziós esetben egy körsorozat interpolációja

A módszer általánosítása gömbökre, amikor egy $S = s_1, s_2, \dots, s_n$ gömbsorozat esetén a gömböket burkoló felület megtalálása a cél.

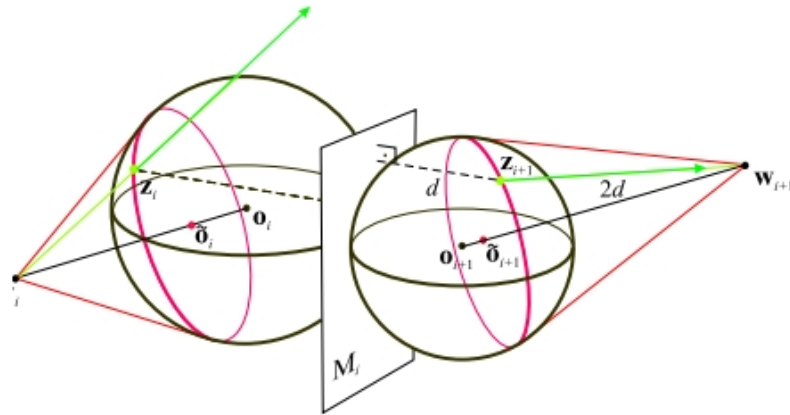
Ekkor a gömbökön köröket határozunk meg, ahol szeretnénk, hogy a felület érintse őket. Ezen érintési körök $(c_1, c_2, \dots, c_n)$, ahol c_i egy $\tilde{O}_i$ középpontú $\tilde{R}_i$ sugarú kör) előállításához tekintsünk rendre a gömbsorozatból három egymást követő elemet. A középpontjaik által felfeszített P_i síkkal elmetszve őket három kört kapunk, így visszavezettük a problémát a síkbeli esetre, s a már tárgyalt módon előállítjuk a középső körhöz a két érintési pontot. Ekkor létezik egy olyan T_i sík, amelyen ez a két érintési pont rajta van, és merőleges a P_i -re. Ezen sík és az s_i gömb metszete adja a keresett érintési kört a gömbön.

A burkolófelületet leíró Hermite-ívek előállításához szükséges megadni azok kezdő- és végpontjait, illetve érintővektorait. Ezen pontokat az i . és $i + 1$. gömb esetén z_i és



3. ábra. Gömbök esetén a burkolófelület meghatározásához szükséges érintési köröket az Apollóniusz-problémára vezetjük vissza [3]

z_{i+1} fogja jelölni. A felület ezen kezdeti érintési pontok meghatározása után adott szögű elforgatásokkal fog előállni, az érintővektorok nagyságát pedig a pontoknak a gömbök hatványsíkjától mért távolsága fogja befolyásolni.



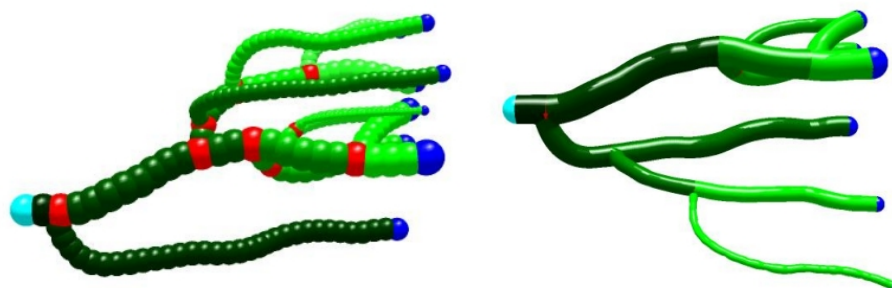
4. ábra. Az érintési pontok és érintővektorok meghatározása [3]

2.2. Ismert módszerek

A bevezetésben említett szoftverekben bár sok esetben kielégítő eredményt adnak gömb-sorozatok burkolásának megoldására, a gömbök speciálisabb elhelyezése esetén már kifejezetten feltűnő a kimenet pontatlansága.

2.2.1. Slabaugh-féle módszer

A Slabaugh és munkatársai által kifejlesztett elmélet [5], illetve érmodellezésre használt módszer [4] esetén is szembetűnőek a gömbsorozatok interpolációjának hiányosságai. Jól látható, hogy sok gömb elhelyezése szükséges ahhoz, hogy jó kimenetet adjon, amelyben ennek ellenére is pontatlanságok tapasztalhatók. Továbbá az sem elhanyagolható, hogy az általa biztosított algoritmus kifejezetten számolásigényes.



5. ábra. Gömbsorozatok burkolása Slabaugh módszerével [5]

2.2.2. Burkolás a Spore™ szoftverben

Az előzőekhez hasonló módon ez a modellező, animációs karakter tervező szoftver is nehézségekbe ütközik a pontos burkolás megvalósításában, már a gömbök interpolációjánál sem nyújt megfelelő eredményt, nem ad pontos kimenetet.



6. ábra. Jól látható, hogy a Spore™ szoftverben a gömbök burkolása nem elég sima, a kapott felület nem szép, nem pontos [7]

2.2.3. Hiányosságok a felületdarabok illesztésekor

A meglévő eljárások az olyan esetekben, amikor az interpoláció során nyert különböző burkolófelület-darabok egymáshoz történő csatlakoztatása a cél, a vártnál még kevésbé elfogadható minőségű kimenetet adnak. Jól látható az alábbi képen is, hogy az egymáshoz csatlakoztatott felületdarabok illesztése nem szolgáltat kellően szép, egy modellező szoftvertől elvárható minőségű eredményt.



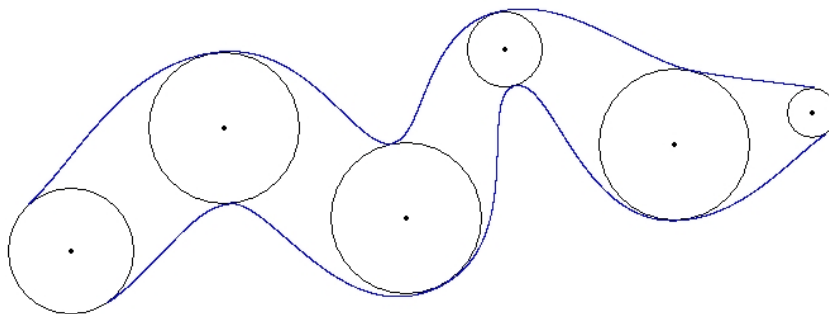
7. ábra. A csatlakoztatás helye kifejezetten feltűnő és pontatlan a SporeTM szoftverben [7]

3. Szoftveres implementáció

Az ismertetett interpolációs módszer eredményeinek reprezentálására kezdtük el fejleszteni a *KSpheres* nevet viselő szoftverünket, melynek készítésekor a minél jobb performanciájú rendszer kialakítására, a számítások egyszerűsítésére törekedtünk.

Az Apollóniusz-feladat megoldását először klasszikus algebrai úton közelítettük meg. A három kör középpontját és sugarát használva egy egyenletrendszert írhatunk fel, melyet megoldva megkaptuk a keresett két, a feladatra megoldást adó kör középpontját, illetve sugarát, az érintési pontot pedig az érintőkör és az adott kör hatványvonalának, illetve a középpontjaikat összekötő egyenes metszéspontjaként kaptuk meg. Azonban ez a módszer nem nyújt megoldást arra az esetre, amikor a három kör középpontja kollineáris, azaz egy egyenesre illeszkednek. Ennek a speciális esetnek a kezelése viszont kifejezetten számolásigényes, ugyanis a középpontok egyenesét metszük mindhárom körrel, s ezeknek a metszéspontoknak a segítségével még további egyeneseket, pontokat és távolságokat kell meghatározni annak érdekében, hogy megkapjuk a keresett érintőköröket és érintési pontokat.

Érdemes viszont megjegyezni, hogy maga a kollinearitás felismerése kézenfekvő feladat a megoldás módszeréből adódóan, nem igényel többlépcsős számolást.



8. ábra. Szoftverünk kimenete kétdimenziós esetben

A hatékonyabb processzor- és memóriahasználatra törekedve implementáltunk egy másik, az Apollóniusz-problémát megoldó, ciklografikus alapokon nyugvó módszert is, mely megoldást a körök pólusai és hatványpontja alapján szolgáltat. Fontos kiemelni, hogy míg a fenti, hagyományos algebrai módszernél az érintőkörök meghatározása után további számítások elvégzésével kapjuk csak meg az érintési pontokat, ez utóbbi módszer jóval

kevesebb számolással adja ugyanazt az eredményt, még az érintőkörök középpontjának és sugarának meghatározása előtt.

A szélső körök esetét is külön kell kezelni, hiszen mindkét tárgyalt módszerünk egy körhármasnál a középső körhöz határozza meg az érintési pontokat, a körsorozat kezdő és záró tagjánál ezek a módszerek nyilvánvalóan nem alkalmazhatók. Ebben az esetben az első és utolsó két körhöz húzott közös külső érintőegyenesek segítségével állítjuk elő a keresett pontokat.

A *KSpheres* tervezése során a kezdetektől fogva egy interaktív, felhasználóbarát szoftver fejlesztése volt a célunk, mellyel már a síkbeli esetben is tetszőlegesen kialakított körsorozatra előállítható a kívánt tulajdonságokkal rendelkező görbe.

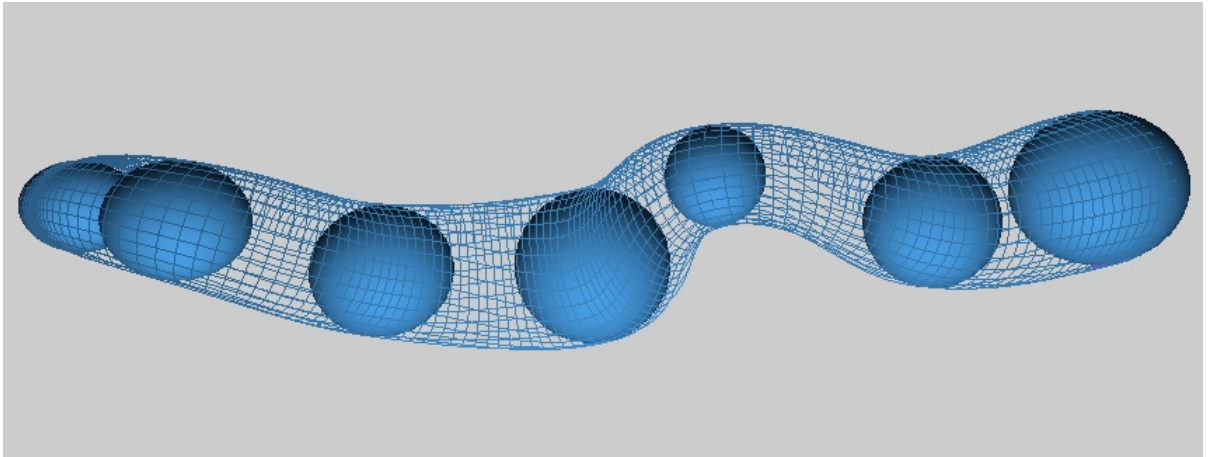
A síkbeli számítások, eredmények térbeli kiterjesztésekor a *BGI* környezet már kényelmetlennek és elavultnak bizonyult, így áttértünk *OpenGL* alapú keretrendszer használatára. Ehhez szükséges volt a meglévő kétdimenziós rendszerünk injektálására az új környezetbe, hiszen a síkbeli rendszer segítségével megkapott érintési pontokat kell felhasználni három dimenzióban a gömbök burkolófelületének meghatározásához.

A gömbsorozat három egymást követő tagja esetén a már tárgyalt módon állítjuk elő az érintési pontokat, s ennek implementálásához koordinátatranszformációt használunk. Ekkor a gömbök eredeti koordinátarendszeréből áttérünk egy új, a középpontok által meghatározott koordinátarendszerre, amelyben már a síkbeli megoldásokat tudjuk alkalmazni. Az így kiszámolt érintési pontok eredeti rendszerbeli koordinátáit a tranzformáció segítségével kapjuk vissza. A síkbeli esethez hasonlóan külön kell kezelni a szélső gömböket, analóg módon itt a szomszédjaikkal vett közös érintőkúppal meghatározott kört tekintjük.

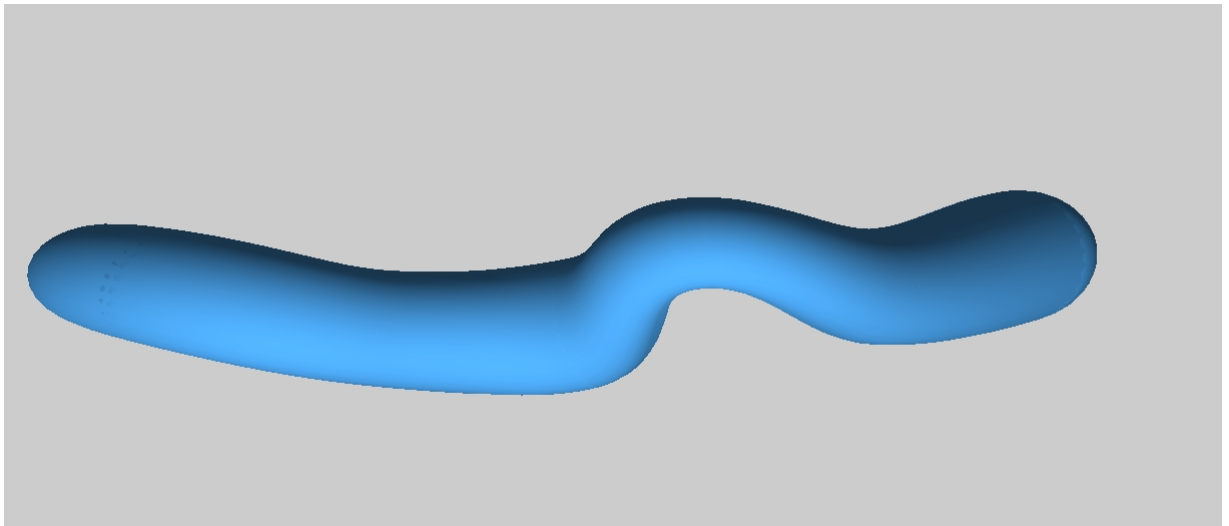
A z_i kiszámításához meg kell határozni az s_i gömb pólusát. Ennek felírása először klasszikus geometriai módszerek segítségével történt, viszont a későbbiekben itt is a projektív geometriai megoldás bizonyult gyorsabbnak és kényelmesebbnek.

A fejlesztés jelenlegi állapotában a burkolófelületet háromszöglapok által határolva jelenítettük meg, melyek csúcsait a paramétervonalak egyes pontjai szolgáltatták, de további lehetőségek vizsgálata is lehetséges, mint például egy (vagy akár több) NURBS felület meghatározása a kiszámolt érintési pontokon keresztül.

Az interakció növelésének érdekében a gömbsorozat tetszőlegesen alakítható: a billentyűzet és az egér segítségével új gömböket szűrhatunk be, változtathatjuk a meglévőket

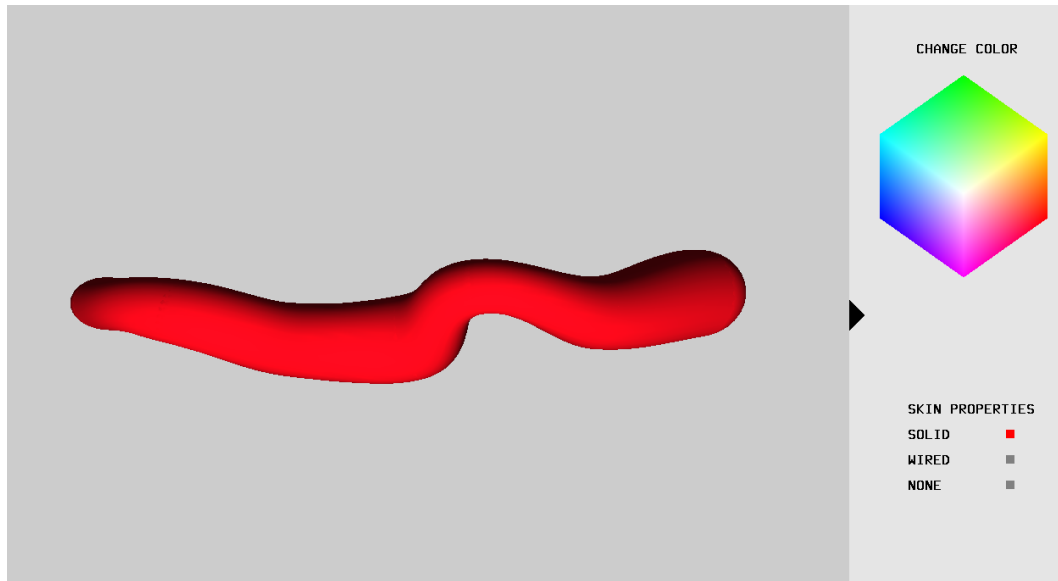


9. ábra. A burkolófelület drótváza



10. ábra. Az elkészült burkolófelület

elhelyezkedését, nagyságát, illetve a navigációt is segítik. A felhasználóbarát kezelőfelület érdekében beépítettünk egy oldalsó panelt is szoftverünkbe, mellyel módosítható a gömbsorozat és a felület megjelenése.



11. ábra. A fejlesztéskor törekedtünk egy felhasználóbarát eszközökkel kibővített szoftver kialakítására

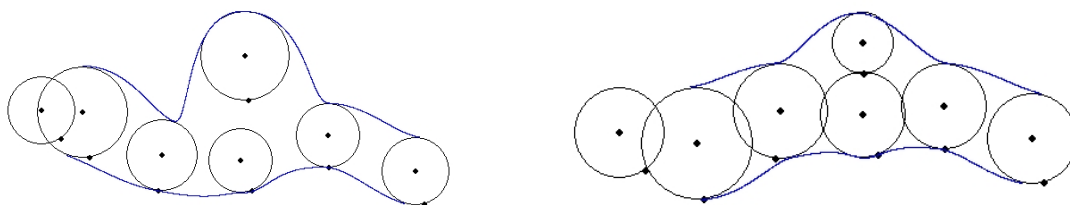
4. Az új csatlakoztatási módszerek

A burkolófelületek csatlakoztatása ugyan már megjelent a komputergrafikában, azonban a felhasznált eljárások, amint azt az előző részben láttuk, nem minden esetben szolgáltatnak elfogadható eredményt. Így szükségét éreztük egy olyan algoritmus kifejlesztésének, ami hatékonyabb, gyorsabb, illetve szebb, vizuálisan is megfelelő minőségű csatlakoztatást biztosít.

A csatlakoztatás megoldásának lehetőségeit már a síkbeli rendszer elkészültével elkezdtek tanulmányozni, s a felmerült ötletek közül némelyeket két dimenzióban implementáltunk.

4.1. Csatlakoztatás két dimenzióban

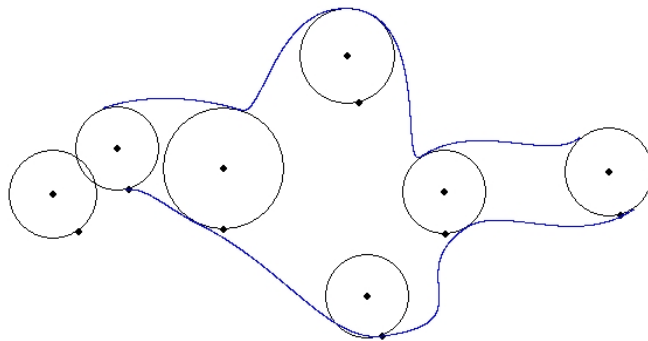
Ezzel a módszerünkkel olyan érintési pontokat akartunk elhagyni, amelyekre már nem lesz szükség a csatlakoztatás után, s helyettük további új érintési pontokat gyártunk az új csatlakoztatandó kör és környezete segítségével. Az ötletünk lényegi része annak meghatározása, hogy mely pontokat hagyjuk el, és helyette mit határozzunk meg. Ez matematikailag pontosan leírható, és sok esetben szép eredményt is szolgáltat, viszont a görbepár a középső kört (amelyhez csatlakoztatunk) csak egy pontban érinti.



12. ábra. A csatlakoztatás két dimenzióban – sokszor szép eredményt kaptunk

Habár bizonyos esetekben ez a módszer is jó eredményt tudott adni, a térbeli kiterjesztése sokkal több nehézséget okozott volna, s a kimenetben felmerülő pontatlanságok azt eredményezték, hogy a csatlakoztatás megoldására ezt a megközelítést elvetettük.

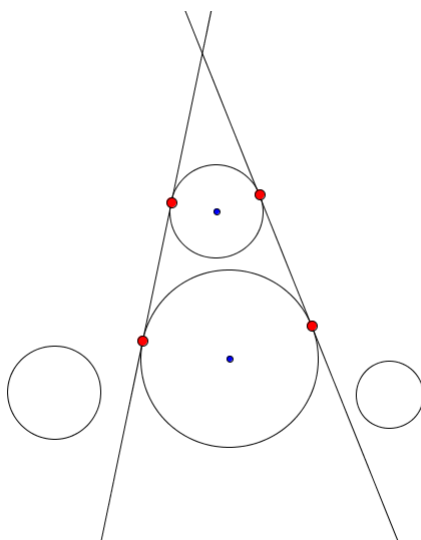
Azonban azok az ötletek, amelyek már a térbeli implementációt ismerve fogalmazódtak meg bennünk, használhatóbbnak bizonyultak. A háromdimenziós rendszer elkészültével lehetőségünk nyílt ezeknek a kipróbálására, implementálására.



13. ábra. A kétdimenziós csatlakoztatás hibája, jelentős torzulás

A megvalósításhoz szükséges struktúra létrehozásához elkülönítjük az eredetileg meglévő gömbök folyamát, s minden gömbnél figyeljük, van-e csatlakoztatott „gyermek” gömb. A továbbiakban nevezzünk folyamnak egy olyan gömbsorozatot, amelyben nincs csatlakoztatás. Módszereink leírásához használjuk a következő jelölésrendszert: jelölje s azt a gömböt, amelyhez csatlakoztatunk, p a csatlakoztatandó, illetve s_1 az s -t megelőző, s_2 pedig az s -t követő gömb legyen. Ha s a folyamban szélső, akkor azt nem csatlakoztatásnak tekintjük, hanem az adott folyam egyszerűen egy új elemmel bővül.

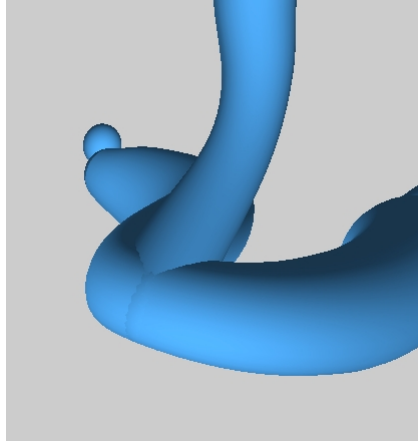
4.2. A csatlakoztatás kezelése szélső gömbökkel



14. ábra. Az érintőkúp segítségével határozzuk meg az új érintési pontot.

A gömbsorozatok folyamként való megjelenítése és kezelése miatt kézenfekvő megol-

dásnak tűnhet a csatlakoztatásra az, hogy szélső esetként kezeljük az s gömböt, azaz a hozzá tartozó érintési kört a már említett módon, s és p közös érintőkúpjának segítségével állítjuk elő. Azonban ez nyilvánvalóan nem érzékeny s szomszédjaira, így bizonyos esetekben nem várhatunk el ettől a módszertől kellően szép kimenetet, a csatlakoztatott felület kifejezetten csúnyán belevág az alapfolyamba, az s gömb környezetébe.

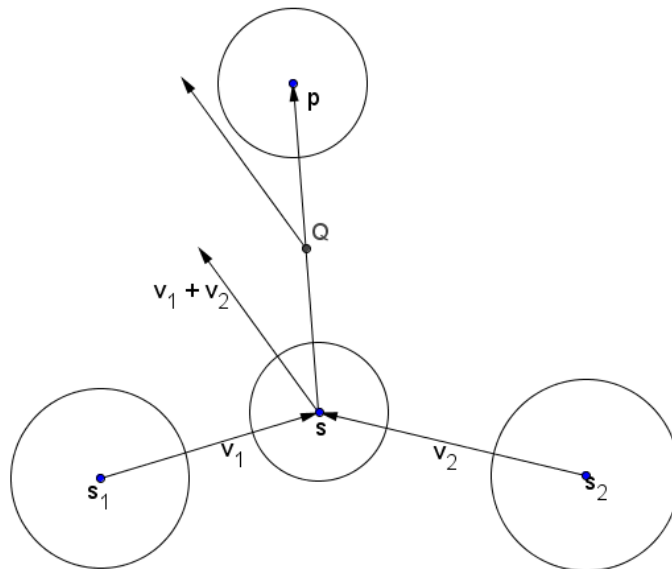


15. ábra. A környezetére érzéketlen módszer belevág az alapfolyamba.

4.3. A szomszédos gömbökre érzékeny módszer

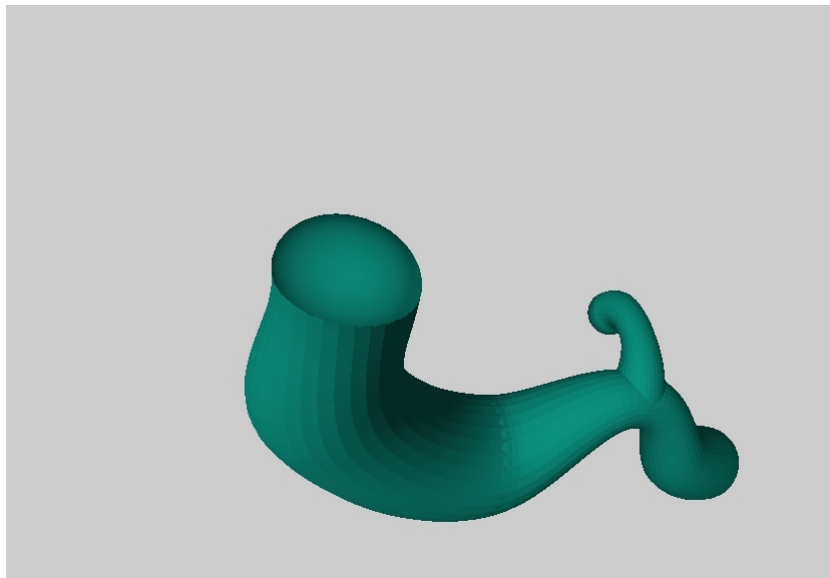
Az első kidolgozott ötletünk alapvetően új módon közelíti meg a problémát. A csatlakoztatott gömbök elsőként számolt érintési pontjának meghatározása nem az Apollóniusz-probléma megoldása alapján történik, hanem a gömbök ún. „erővonalainak” segítségével távolságot határozunk meg, és egy frappáns arány által kiszámolásra kerül a görbék kezdőpontja.

Az elgondolás alapja az, hogy olyan görbét kapjunk, amely függ a csatlakoztatott, és a csatlakoztatandó gömbök környezetétől. Vesszük először az s_1 középpontjából s középpontjába mutató irányvektort, ez legyen v_1 , majd s_2 középpontjából az s középpontjába mutatót, mely legyen v_2 . Ezek után kiszámoljuk az s középpontjából p középpontjába mutatót, aminek hosszát megfelelően egy Q pontot határozunk meg. Az említett arány az eljárás ezen részében jelenik meg, amikor is Q -t a v_1 és v_2 vektoroktól függően eltoljuk. Kifejezetten jó eredményt szolgáltatott a v_1 és v_2 vektorok összege, mellyel tehát a Q -t eltolva meghatároztuk azt a pontot, amelyhez azután kiszámolhatjuk az érintési kört p -n, mint adott pólushoz tartozó polársíkkal vett metszetet. Így ez az a pont, amely meghatá-



16. ábra. Érintési pontok meghatározása az „erővonalakkal”

rozza, merre is görbül a csatoló felület, konstrukciója az eddigiekben alkalmazott módon, a pólus és az érintési körök figyelembevételével történik.



17. ábra. Az „erővonalas” megoldás eredménye

Ezzel a módszerrel már sikerült a térbeli csatlakoztatást megoldani, s ahogy a képen is látható (17. ábra), itt is kifejezetten szép kimenetet kaptunk. Azonban hosszútávon

ez sem bizonyult megfelelő megoldásnak, ugyanis a gömbök elhelyezkedésétől függően a csatlakozás eredményeképp létrejött felület jelentős mértékben eltorzult.

4.4. A megoldást szolgáltató eljárás

A következőkben a jelenlegi módszerünket fogjuk bemutatni, amellyel a csatlakoztatás megoldását pontosan leírhatjuk, s kellően szép, pontos kimenetet biztosít.

A megvalósításához a gömbökből egy gráfot építünk, melynek gyökéreleme a gömb-sorozat első tagja, s egy szülő csomópontnak a gyermeke az őt reprezentáló gömbnek a sorozatbeli rákövetkezője. A csatlakoztatandó elemek pedig a szülőnek további gyermekeiként jelennek meg.

A felület kirajzolásánál ezt a gráfot hibrid módon járjuk be: csatlakoztatás esetén szélesség, egyébként pedig mélység alapján.

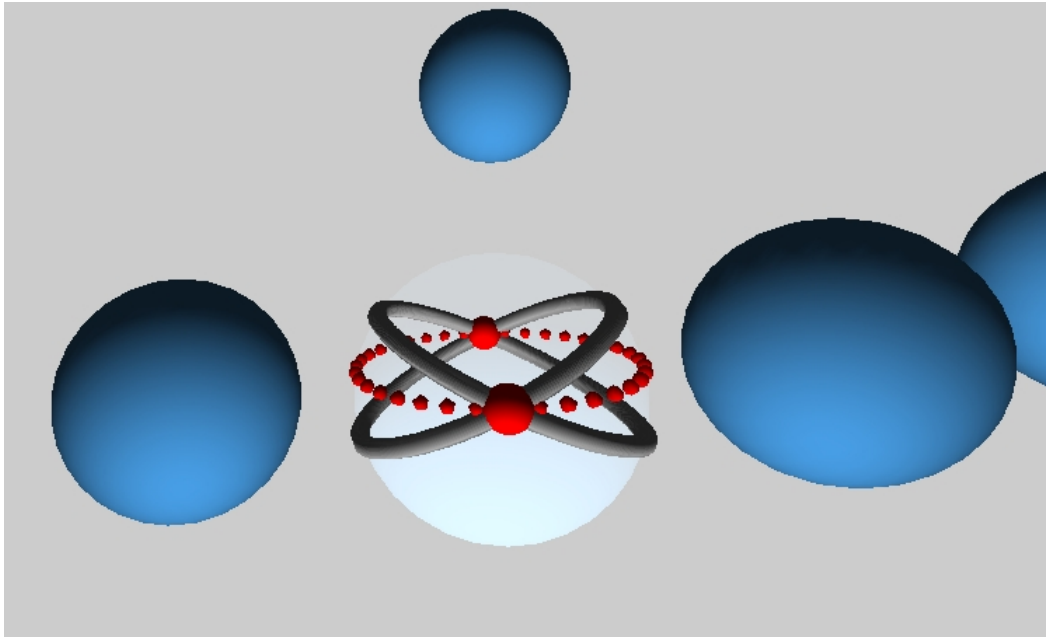
A csatlakoztatott burkolófelülethez szükséges érintési kör előállítása e módszerünk szerint az Apollóniusz-probléma megoldásán alapszik. Ennek a körnek a definiálásakor kiemelt figyelmet fordítottunk arra, hogy a kapott kimenet azonos mértékben függjön s_1 , s és s_2 elhelyezkedésétől és nagyságától, s kellően érzékeny legyen a folyam mindkét irányára.

Tekintsük először az s_1 , s és p , majd a p , s , s_2 gömbhármassokat, s ezekre oldjuk meg az Apollóniusz-feladatot. Majd a kapott két érintési körből (síkjai legyenek P_1 és P_2 módon jelölve) egyfajta átlagot határozunk meg: metsszük az s gömböt a körlapok síkjainak metszéspontjával, így két pontot kapunk a gömbön.

Bizonyos speciális esetekben előfordulhat, hogy P_1 és P_2 metszéspontja nem metszi az s gömböt. Ha ez az egyenes érinti a gömböt, akkor ezzel az érintési ponttal tovább tudunk dolgozni, hiszen egy normálvektor definiálásával fel tudjuk írni a kívánt síkot a körlap meghatározására.

Azonban ha a gömbnek és a metszéspontnak nincs közös pontja, akkor ez a módszer nem alkalmazható. Viszont ez csak az s_1 , s , s_2 és p nagyon speciális elhelyezkedésekor áll fenn, ami a gyakorlatban nem jelenik meg. Így ennek kezelése, bár csak elméleti probléma, a jövőbeni kutatási terv része lehet.

Ez jó kiindulási alap lett a csatlakoztatáshoz szükséges érintési kör meghatározására. Az eljárás által adott kimenet pontosítására és tökéletesítésére megpróbáltuk ezt a két pontot előre meghatározott vektorokkal eltolni a gömbön a környező gömbök elhelyezke-



18. ábra. A két érintési kör átlagának meghatározása

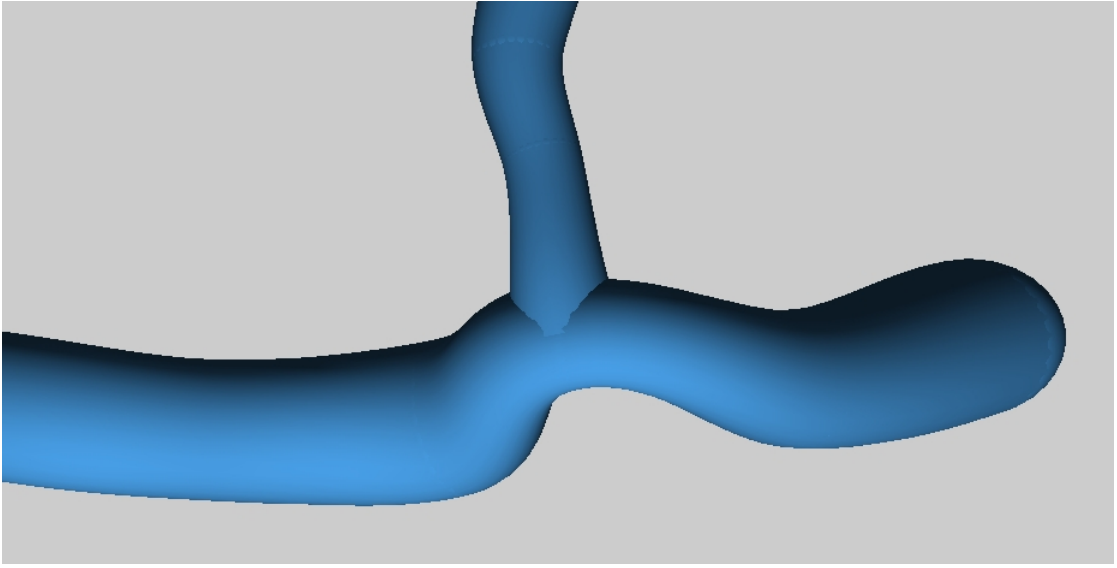
désétől és nagyságától függően, akár egy speciális konstans felhasználásával.

A kívánt érintési körlap normálvektorának meghatározása több lehetőséget rejt magában, hiszen itt is a környezetre kellően érzékeny vektort szeretnénk használni. Használható kimenetet kaptunk, mikor P_1 és P_2 normálvektorai mellett az s középpontjából p középpontjába mutató vektort is felhasználtuk a normálvektor definiálásakor.

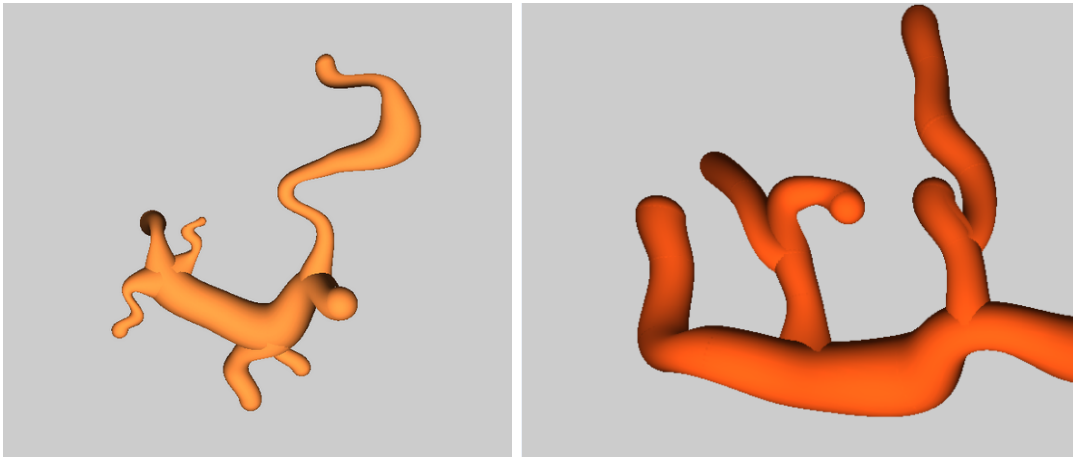
Az algoritmusunk a létrejövő két pont alapján P_1 és P_2 normálvektorainak átlagából készít egy új körlapot s -en. Ezután a burkolófelületet a fentebb tárgyalt módszerrel állítjuk elő, úgy hogy s és p már egy új, saját folyamat alkot. Ezzel a megoldással így tehát nem szükséges egy másik, teljesen új algoritmus készítése a csatlakoztatandó gömbök folyamának egészére, csupán az s -t és p -t burkoló felületdarabhoz.

Már ezzel a csatlakoztatással is szép kimenetet tudtunk generálni, figyelembe véve, hogy a többi, már fent említett ötlet kevésbé használható eredményt adott. A felületdarabok egymáshoz illesztésének tökéletesítésére egy újabb algoritmust dolgoztunk ki. Annak érdekében, hogy az új felületdarab szélei ne durván érkezzenek bele a meglévő alapfolyamba, egy lyukat vágunk az s gömbön, s a meglévő Hermite-ívek felhasználásával új paraméteres görbéket írunk fel.

Az előzőleg használt jelöléseket megtartva vezessük be a következőket: legyen S_1 az s

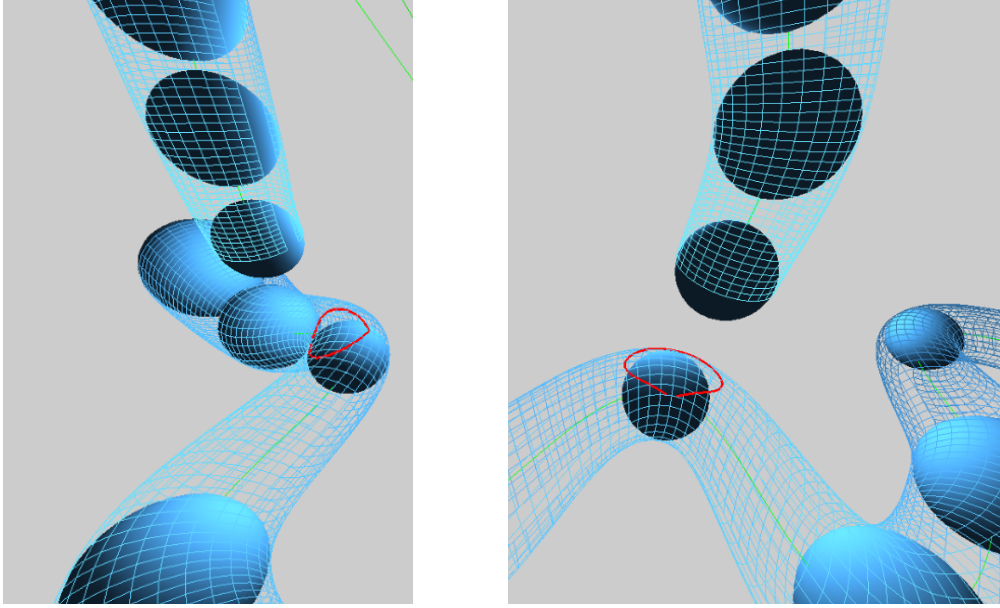


19. ábra. Az érintőkörök átlagolásával nyert csatlakoztatás



20. ábra. A módszer alkalmas bonyolultabb alakzatok, felületek előállítására

gömbön azon érintési kör, amelyet meghatároztunk még s -nek a saját, eredeti folyamában, S_2 pedig az új, csatlakoztatással nyert érintési kör. Tekintsük az S_1 -nek azon pontját, amely legközelebb esik a p gömbhöz, ezt jelöljük W -vel. Ennek a pontnak a meghatározásához egy ortogonális projekciót használunk, melynek lényege, hogy az s középpontjából a p középpontjába mutató vektort az eredeti koordináta-rendszerből az S_1 érintési kör síkja és annak normálvektora által meghatározott koordináta-rendszerbe transzformáljuk. Az így kapott vektornak a harmadik koordinátáját tekintjük 0-nak, így ez a vektor az érintőkör síkjára kerül. Ezt visszatranszformálva az eredeti koordináta-rendszerbe, a megfelelő



21. ábra. A felületen kivágott lyuk meghatározása.

hossz beállítása után megkapjuk az érintőkörön azt a pontot, amit a lyuk középpontjának tekintünk.

A lyuk felírásához ebből a W pontból egy $L(\phi, t)$ kétváltozós függvény segítségével meghatározunk pontokat a folyamon, ezek fogják szolgáltatni a lyuk „körvonalát”. (A függvény felírásához a [3]-ban közölt jelölésrendszert használtuk.)

$$\begin{aligned} \mathbf{L}(\phi, t) = & H_0^3(t) \mathbf{z}_i(\phi) + H_1^3(t) \mathbf{z}_{i+1}(\phi) + \\ & + H_2^3(t) \cdot p(s_i) \cdot 2 \cdot d(M_i, \mathbf{z}_i(\phi)) \cdot \frac{\mathbf{w}_i - \mathbf{z}_i(\phi)}{\|\mathbf{w}_i - \mathbf{z}_i(\phi)\|} + \\ & H_3^3(t) \cdot p(s_{i+1}) \cdot 2 \cdot d(M_i, \mathbf{z}_{i+1}(\phi)) \cdot \frac{\mathbf{w}_{i+1} - \mathbf{z}_{i+1}(\phi)}{\|\mathbf{w}_{i+1} - \mathbf{z}_{i+1}(\phi)\|} \\ & \phi \in \left[\frac{-\pi}{4}, \frac{\pi}{4} \right], \quad t = \cos \frac{2\phi}{5} \end{aligned}$$

Majd ezekben a pontokban meghatározzuk a felület pontbeli érintősíkjait, és ezek segítségével állítjuk elő azoknak az íveknek a kezdőpontját, illetve kezdőpontbeli érintővektorát, amelyek csatlakozni fognak a felülethez.

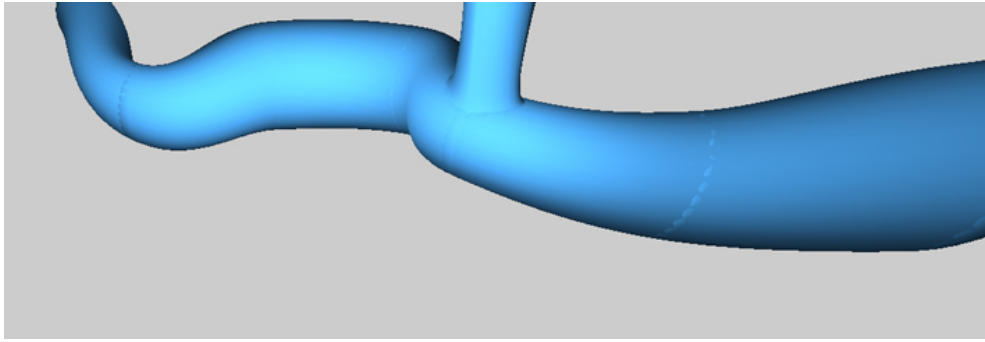
A kezdőpontok meghatározása úgy történik, hogy a csatlakoztatott gömb középpontjá-

ból a gömbön kapott megfelelő z_i -be mutató vektort konstruálunk, ezt levetítjük az előbbieken ismert koordinátatranszformációs technika segítségével az érintősíkra. Ezután a lyuk középpontjából indítva a vektor reprezentánsát megkaphatjuk a lyukon a kívánt pont.

A felület paramétervonalainak a felírásához a lyuk esetében is meg kell állapítani a görbe érintővektorait, amit hasonlóan síkra történő vetítés segítségével számolunk koordinátatranszformációk segítségével. A lyuk egy pontjából a lyuk középpontjába mutató vektort vetítjük az érintősíkra, így megkapjuk azt a vektort, ami az érintősíkon van és a lyuk középpontja felé mutat.

Az ezen görbéket meghatározó érintővektorok hosszának meghatározására viszont nem az előzőekben használt módszert alkalmaztuk (a két gömb hatványvonalától vett távolság kétszerese), hanem egy olyan speciális értéket definiáltunk, amelynek használatával egyenlő mértékben függ az érintőhossz a két gömbtől és magától a kialakított lyuktól is.

Az érintősík meghatározására felírjuk a felület szögelfordulás szerinti deriváltjának (érintővektorának) kiszámítását, ami a hosszanti (t szerinti) deriválttal együtt az adott pontban megadja az érintősíkot. A számolás könnyítésének érdekében, mivel az érintőkúp csúcsától a kör minden pontja egyforma távol van, így a $\|\mathbf{w}_i - \mathbf{z}_i(\phi)\|$ típusú két nevező helyett a fix (azaz ϕ -től független) $\|\mathbf{w}_i - \mathbf{z}_i\|$ értékeket írjuk.



22. ábra. Kép a simított csatlakozásról

$$\begin{aligned} \mathbf{s}_i(\phi, t) = & H_0^3(t) \mathbf{z}_i(\phi) + H_1^3(t) \mathbf{z}_{i+1}(\phi) + \\ & + H_2^3(t) \cdot \frac{p(s_i)}{\|\mathbf{w}_i - \mathbf{z}_i\|} \cdot 2 \cdot d(M_i, \mathbf{z}_i(\phi)) \cdot (\mathbf{w}_i - \mathbf{z}_i(\phi)) + \\ & + H_3^3(t) \cdot \frac{p(s_{i+1})}{\|\mathbf{w}_{i+1} - \mathbf{z}_{i+1}\|} \cdot 2 \cdot d(M_i, \mathbf{z}_{i+1}(\phi)) \cdot (\mathbf{w}_{i+1} - \mathbf{z}_{i+1}(\phi)) \end{aligned}$$

4. Az új csatlakoztatási módszerek

A felületet ϕ szerint tagonként deriváljuk, ehhez először a $\mathbf{z}_i(\phi)$ -t kell felírunk és deriválnunk. Az érintési körben egy derékszögű ortonormált rendszert írunk föl, középpontja $\tilde{\mathbf{o}}_i$, egyik vektora $\mathbf{e}_i^1 = \frac{\mathbf{z}_i - \tilde{\mathbf{o}}_i}{\|\mathbf{z}_i - \tilde{\mathbf{o}}_i\|}$, a másik vektorát pedig ennek a vektornak és az elforgatás tengelyeként szolgáló $\mathbf{p}(s_i) \cdot (\mathbf{w}_i - \tilde{\mathbf{o}}_i)$ vektornak a vektoriális szorzataként kapjuk: $\mathbf{e}_i^2 = \mathbf{e}_i^1 \times \frac{\mathbf{p}(s_i) \cdot (\mathbf{w}_i - \tilde{\mathbf{o}}_i)}{\|\mathbf{p}(s_i) \cdot (\mathbf{w}_i - \tilde{\mathbf{o}}_i)\|}$. Itt feltesszük, hogy a forgatás a $\mathbf{p}(s_i) \cdot (\mathbf{w}_i - \tilde{\mathbf{o}}_i)$ irányba nézve óramutató járásával megegyező.

Ebben a rendszerben

$$\mathbf{z}_i(\phi) = R_i \mathbf{e}_i^1 \cos \phi + R_i \mathbf{e}_i^2 \sin \phi,$$

ϕ szerinti deriváltja pedig (az egyszerűség kedvéért a $\frac{d}{d\phi} \mathbf{z}_i(\phi)$ deriváltat $\dot{\mathbf{z}}_i(\phi)$ -vel jelöljük):

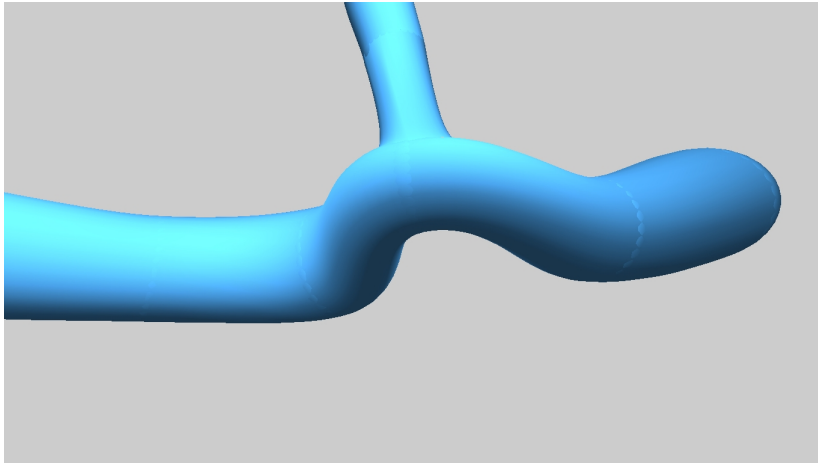
$$\dot{\mathbf{z}}_i(\phi) = -R_i \mathbf{e}_i^1 \sin \phi + R_i \mathbf{e}_i^2 \cos \phi.$$

Ezzel analóg módon számolható $\dot{\mathbf{z}}_{i+1}(\phi)$ is.

A $d(M_i, \mathbf{z}_i(\phi))$ kifejezést szintén deriválnunk kell. Az említett $\mathbf{m}$ és $\mathbf{n}_m$ vektorok segítségével felírhatjuk a távolságot mint skaláris szorzatot:

$$d(M_i, \mathbf{z}_i(\phi)) = \langle \mathbf{n}_m, \mathbf{z}_i(\phi) - \mathbf{m} \rangle.$$

Ha az $\mathbf{a}$ vektor koordinátáit $(\mathbf{a}.x, \mathbf{a}.y, \mathbf{a}.z)$ -vel jelöljük, akkor a távolság



23. ábra. A csatlakoztatás kisimítása az újabb algoritmus segítségével

$$d(M_i, \mathbf{z}_i(\phi)) = \mathbf{n}_m.x \cdot (\mathbf{z}_i(\phi).x - \mathbf{m}.x) + \mathbf{n}_m.y \cdot (\mathbf{z}_i(\phi).y - \mathbf{m}.y) + \mathbf{n}_m.z \cdot (\mathbf{z}_i(\phi).z - \mathbf{m}.z)$$

ennek deriváltja pedig

$$\frac{d}{d\phi}d(M_i, \mathbf{z}_i(\phi)) = \mathbf{n}_m.x \cdot \dot{\mathbf{z}}_i(\phi).x + \mathbf{n}_m.y \cdot \dot{\mathbf{z}}_i(\phi).y + \mathbf{n}_m.z \cdot \dot{\mathbf{z}}_i(\phi).z,$$

ami az előző, $\dot{\mathbf{z}}_i(\phi)$ -re vonatkozó formulát behelyettesítve:

$$\begin{aligned} \frac{d}{d\phi}d(M_i, \mathbf{z}_i(\phi)) &= \mathbf{n}_m.x \cdot (-R_i \mathbf{e}_i^1.x \sin \phi + R_i \mathbf{e}_i^2.x \cos \phi) + \\ &\quad + \mathbf{n}_m.y \cdot (-R_i \mathbf{e}_i^1.y \sin \phi + R_i \mathbf{e}_i^2.y \cos \phi) + \\ &\quad + \mathbf{n}_m.z \cdot (-R_i \mathbf{e}_i^1.z \sin \phi + R_i \mathbf{e}_i^2.z \cos \phi). \end{aligned}$$

Ezzel analóg módon számolható $\frac{d}{d\phi}d(M_i, \mathbf{z}_{i+1}(\phi))$ is, amiben a $\dot{\mathbf{z}}_{i+1}(\phi)$ koordinátái fognak megjelenni.

Ezek alapján a felület ϕ szerinti deriváltja:

$$\begin{aligned} \frac{d}{d\phi} \mathbf{s}_i(\phi, t) &= H_0^3(t) \dot{\mathbf{z}}_i(\phi) + H_1^3(t) \dot{\mathbf{z}}_{i+1}(\phi) + \\ &\quad + \left(H_2^3(t) \cdot \frac{p(s_i)}{\|\mathbf{w}_i - \mathbf{z}_i\|} \cdot 2(\mathbf{w}_i - \mathbf{z}_i(\phi)) \right) \cdot \frac{d}{d\phi}d(M_i, \mathbf{z}_i(\phi)) - \\ &\quad - \left(H_2^3(t) \cdot \frac{p(s_i)}{\|\mathbf{w}_i - \mathbf{z}_i\|} \cdot 2d(M_i, \mathbf{z}_i(\phi)) \right) \dot{\mathbf{z}}_i(\phi) + \\ &\quad + \left(H_3^3(t) \cdot \frac{p(s_{i+1})}{\|\mathbf{w}_{i+1} - \mathbf{z}_{i+1}\|} \cdot 2(\mathbf{w}_{i+1} - \mathbf{z}_{i+1}(\phi)) \right) \frac{d}{d\phi}d(M_i, \mathbf{z}_{i+1}(\phi)) - \\ &\quad - \left(H_3^3(t) \cdot \frac{p(s_{i+1})}{\|\mathbf{w}_{i+1} - \mathbf{z}_{i+1}\|} \cdot 2d(M_i, \mathbf{z}_{i+1}(\phi)) \right) \dot{\mathbf{z}}_{i+1}(\phi). \end{aligned}$$

Az érintősík meghatározásához már csak a t szerinti deriválásra van szükség, ami az

Hermite-polinomok deriválása nyomán egyszerűen adódik:

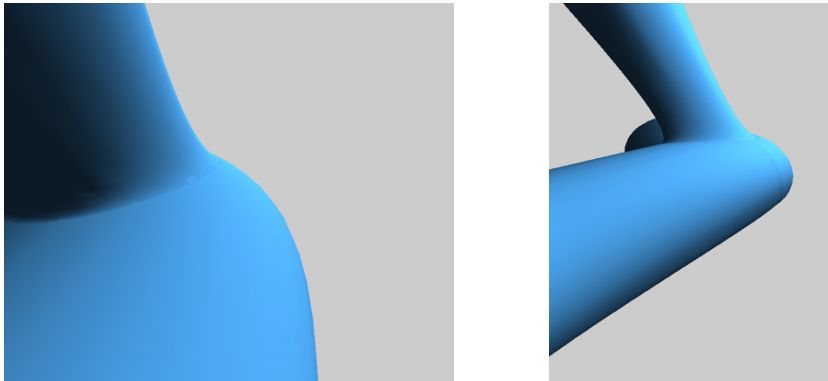
$$\begin{aligned} \frac{d}{dt} \mathbf{s}_i(\phi, t) = & H_0^{3'}(t) \mathbf{z}_i(\phi) + H_1^{3'}(t) \mathbf{z}_{i+1}(\phi) + \\ & + H_2^{3'}(t) \cdot \frac{p(s_i)}{\|\mathbf{w}_i - \mathbf{z}_i\|} \cdot 2 \cdot d(M_i, \mathbf{z}_i(\phi)) \cdot (\mathbf{w}_i - \mathbf{z}_i(\phi)) + \\ & + H_3^{3'}(t) \cdot \frac{p(s_{i+1})}{\|\mathbf{w}_{i+1} - \mathbf{z}_{i+1}\|} \cdot 2 \cdot d(M_i, \mathbf{z}_{i+1}(\phi)) \cdot (\mathbf{w}_{i+1} - \mathbf{z}_{i+1}(\phi)) \end{aligned}$$

A fentiek ismeretében tetszőleges felületpontban felírhatjuk az érintősíkot.

A p gömb z_i -je és a W által meghatározott szöggel elforgatjuk W -t, így az induló görbe a p gömbön lévő z_i irányból fog indulni, tehát nem csavarodik meg.

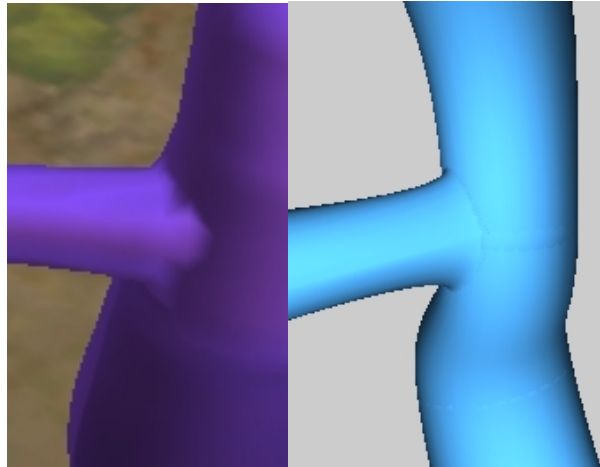
4.5. Meglévő módszerekkel való összehasonlítás

Érdemes megfigyelni, hogy a már említett modellező szoftverek által szolgáltatott kimenetekhez képest módszerünkkel jobb eredményt kaptunk abban az értelemben, hogy az illesztés esetünkben matematikailag pontosan definiált, a felületek folytonosan csatlakoznak, illetve megoldásunk mélyebb differenciálgeometriai vizsgálatra is lehetőséget ad.

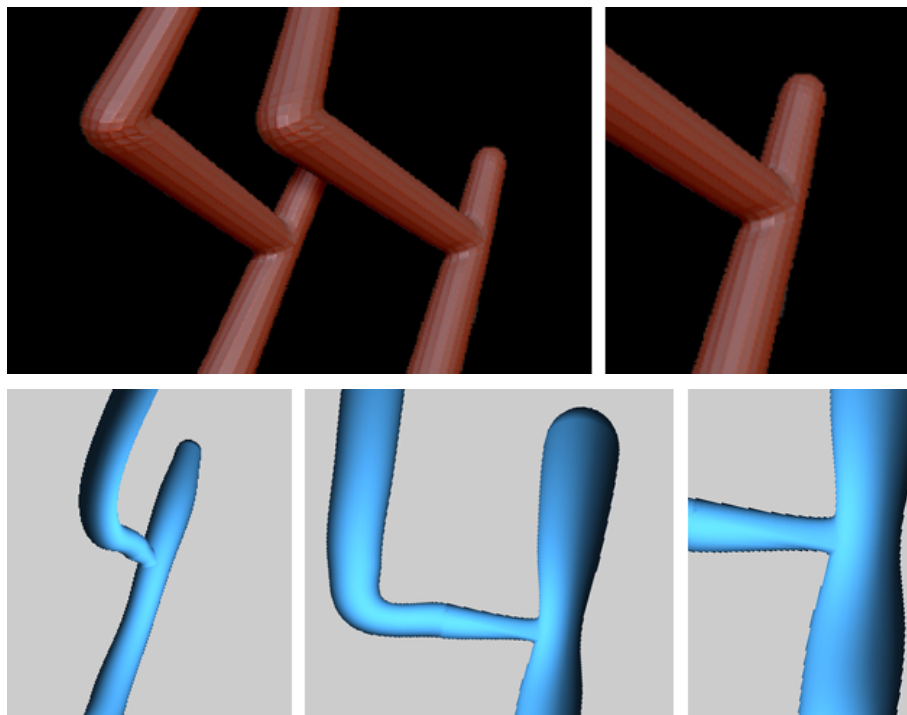


24. ábra. Simított csatlakoztatás a saját módszerrel

A módszerek között látványos a különbség, az általunk adott módszerrel az illesztés pontosabb, simább, folytonosabb. Az alábbi képeken látható, hogy az eljárásunkkal sokkal finomabb csatlakoztatást tudunk megvalósítani.



25. ábra. A SporeTM[7] szoftver kimenete (bal) és a saját módszerünkkel kapott eredmény (jobb) összehasonlítása



26. ábra. Csatlakoztatás a ZSpheres [6], illetve a saját szoftverünkben

5. Összefoglalás

A körök és gömbök interpolációján alapuló felületmodellezés a komputergrafikában bár már megjelent, a használt eljárások sok esetben nem tudnak kielégítő eredménnyel szolgálni, s matematikailag sem tudnak pontos leírást nyújtani az elkészített burkolófelületről. A munkánk alapjául szolgáló cikkben olyan módszer kerül bemutatásra, amelyet érdemes volt áttanulmányozni és továbbgondolni.

A meglévő modellezési technikák már gömbsorozatok burkolásánál sem tudnak igazán jó kimenetet adni, az interpoláció során nyert különböző burkolófelület-darabok egymáshoz történő csatlakoztatására pedig végképp nem tudnak megfelelő eredményt adni. Így mindenképp szükségesnek éreztük egy olyan algoritmus kifejlesztését, ami hatékonyabb, gyorsabb, illetve szebb, vizuálisan kellemes csatlakoztatást biztosít.

A csatlakoztatás problémájának megoldását több irányból is megközelítettük. Foglalkoztunk a síkbeli esetben a körsorozatot interpoláló görbék csatlakoztatásával, s bár sok esetben látszólag jó kimenetet szolgáltatott, ezt a módszert nem lehetséges több dimenzióra általánosítani, így a gömbsorozatok burkolásánál már nem tudtuk alkalmazni.

Térbeli csatlakoztatásra is többféle algoritmust dolgoztunk ki. A kézenfekvőnek tűnő eljárás, mely szerint a meglévő módszerrel egyszerűen új gömbfolyamot burkolunk, sem adott problémamentes megoldást. A módszer ugyanis érzéketlen a csatlakoztatás környezetére, így azokban az esetekben, mikor a környező gömbök kellően közel vannak a csatlakoztatáshoz, az újonnan létrehozott felület belehasít a meglévő burkolófelületbe.

Ezek után mindenképp olyan módszert szerettünk volna kitalálni, aminél figyelembe vesszük a környező gömbsorozatot is. Ennek eredményeképp született meg egy olyan elgondolás, amelyben a szomszédos gömbökből származtatott pontok és vektorok segítségével csatlakoztatunk. Ez az eljárás sikeresnek tűnt, sok esetben szép kimenetet kaptunk, de sajnos itt is problémákba ütköztünk a gömbök elhelyezkedésétől függően.

Azonban egy másik ötlet, mely szerint a csatlakoztatott burkolófelülethez szükséges érintési kör előállítására is az Apolloniusz probléma megoldásával történjen, ígéretesnek bizonyult. Az eljárás érzékeny a gömbök elhelyezkedésére és nagyságára is, s bizonyos arányok hangolásával kellően szép és pontos csatlakoztatást sikerült elérni.

A felületdarabok egymáshoz illesztésének tökéletesítésére egy újabb algoritmust dolgoztunk ki. Új paraméteres görbéket írunk fel annak érdekében, hogy az csatlakoztatott

felületdarab szélei ne durván érkezzenek a meglévő alapfolyamba. Ezzel az eljárással kiegészítve a csatlakoztatásra adott algoritmusunkat, rendkívül sima és pontos csatlakoztatást tudunk biztosítani.

Sikerült tehát olyan algoritmust adni, amelynek segítségével bonyolultabb felületeket, élesebb elágazásokat is kezelni tudunk, illetve a meglévő módszereknél szebb és matematikailag pontosabban definiált felületet, alakzatokat hozhatunk létre.

Munkánk során egy olyan felhasználóbarát interaktív szoftvert fejlesztettünk, amely nemcsak egyszerűen a számításaink és eredményeink reprezentálását biztosítja, hanem további finomhangolással akár egy modellezőrendszer alapjául is szolgálhat.

A jövőben továbbra is szeretnénk ezzel a témával foglalkozni, hiszen rendkívül sok lehetőséget rejt még magában a csatlakoztatás algoritmusának finomítása, esetlegesen egy másik módszer kidolgozása. Példának okáért megfogalmazódott egy olyan gondolatunk is, hogy a csatlakoztatásnál csak foltozunk, kisebb felületdarabokat illesztünk rá a kiindulási felületekre.

6. Köszönetnyilvánítás

Ezúton szeretnénk megköszönni a sok segítséget témavezetőinknek, *Kunkli Rolandnak* és *dr.Hoffmann Miklósnak*, akiknek kitartó munkájuk, türelmük és támogatásuk nélkül ez a dolgozat nem készülhetett volna el.

Köszönjük a rengeteg hasznos tanácsot, ötletet, építő jellegű kritikát, s a rendkívüli segítőkészséget, hogy bármilyen apró kérdéssel, kéréssel fordulhattunk hozzájuk.

7. Irodalomjegyzék

- [1] Bácsó Sándor, Papp Ildikó, Szabó József: *Projektív geometria*, 2006, Kossuth Egyetemi Kiadó.
- [2] Dahl, H.E.I., Krasauskas, R.: *Rational fixed radius rolling ball blends between natural quadrics*, Computer Aided Geometric Design, 29 (2012), 691–706.
- [3] Kunkli, R., Hoffmann, M.: *Skinning of circles and spheres*, Computer Aided Geometric Design, 27 (2010), 611–621.
- [4] Rossignac, J., Whited, B., Slabaugh, G., Fang, T., Unal, G., Pearling: *3D Interactive Extraction of Tubular Structures from Volumetric Images*, MICCAI Workshop: Interaction in Medical Image Analysis and Visualization, 2007.
- [5] Slabaugh, G., Rossignac, J., Whited, B., Fang, T., Unal, G.: *3D Ball Skinning using PDEs for Generation of Smooth Tubular Surfaces*, Computer-Aided Design, 42 (2009), 18–26.
- [6] ZSpheres, <http://www.pixologic.com/docs/index.php/ZSpheres>
- [7] The Official SporeTM, <http://www.spore.com>

8. Saját munka leírása

- Kör- és gömbsorozatok interpolációja: meglévő számítások egyszerűsítése, gyorsítása, implementálása
- Két dimenziós csatlakoztatási algoritmus kifejlesztése és implementálása
- Térbeli csatlakoztatás mint szélső gömbök algoritmusának megtervezése, implementálása
- Térbeli csatlakoztatás a gömbök környezetének figyelembevételével: kifejlesztése és megvalósítása
- Végso, pontos algoritmus fejlesztése és implementálása a térbeli csatlakoztatás eredményes megoldására
- A csatlakoztatást kiegészítő, tökéletesítő eljárás tervezése és kidolgozása
- Az eredményeket bemutató, interaktív modellezőszoftver elkészítése