Eötvös Loránd Tudományegyetem
Informatikai Kar
Algoritmusok és Alkalmazásaik Tanszék

Szabadformájú felületek görbületeloszlásának optimalizálása

Salvi Péter

Témavezetők: Dr. Várady Tamás és Dr. Vida János

Budapest, 2005. június

Tartalomjegyzék

1. fejezet

Bevezetés

„If she be not so to me,
What care I how fair she be?”

„Ha nem hajlik énfelém,
Szépségét mit bánom én?”

(George Wither) [1]

A számítógéppel segített geometriai tervezés (*Computer Aided Geometric Design, CAGD*) a számítástechnika egy rohamosan fejlődő területe, amely bonyolult testek számítógépes reprezentációjával és a modellek tervezéséhez, módosításához szükséges algoritmusok kifejlesztésével foglalkozik [4].

Különböző testek, objektumok illetve alkatrészek számítógépes ábrázolására különféle módszereket dolgoztak ki. Ezek közül az egyik első a drótvázas (*wireframe*) modellezés volt, majd kialakult a konstruktív testgeometria (*Constructive Solid Geometry, CSG*) elmélete, ahol ún. *primitív testeken* (gömb, téglatest, kúp stb.) végzett halmazműveletek segítségével definiálják a modellt. Végül napjainkban leginkább a határfelület reprezentáció (*Boundary representation, B-rep*) terjedt el, amely a testet határoló felületekkel adja meg.

Ez utóbbiban megjelentek a parametrikus, szabadformájú (*freeform*) felületek, amelyek nagyobb szabadságot adnak a tervezőnek és lehetővé teszik igazán komplex alakzatok (mint pl. egy autókarosszéria) ábrázolását [10, 3]. Ezeket pusztán egyszerű analitikus felületek felhasználásával nem lehet leírni.

1.1. Mérnöki visszafejtés és egyenletesen görbült felületek

Míg az elmúlt két–három évtizedben a számítógépes tervezésben objektumok létrehozása volt az elsődleges cél, az utóbbi időben jelentős hangsúlyt

[1] Szokolay Károly fordítása

kapott a valóságban már meglévő alkatrészek, testek nagyméretű mért ponthalmazok alapján való számítógépes reprezentációjának előállítása, és az így kapott modellek utólagos változtatása, javítása. Ez a mérnöki visszafejtés (*Reverse Engineering, RE*) [4], ahol központi szerepet kap az egyenletes görbületeloszlású (*fair*) felületek készítése, illetve a felületek egyenletessé tétele, simítása (*fairing*)[2].

A mérnöki visszafejtés gyakorlati (főként gépjárműipari) alkalmazásaiban a tökéletesen görbült felületek előállítása igen nagy jelentőséggel bír. A karosszéria „szépsége" ugyanis rendkívül fontos szempont: egyfajta határvonal ez, amely elválasztja az elsőrangú gyártókat vetélytársaiktól – presztízskérdés.

Noha már több évtizede folynak ezen a téren kutatások, ezidáig nem sikerült egy egyértelmű, széles körben elfogadott matematikai definíciót találni a görbületeloszlás minősítésére. A „fair"-ség szubjektív, megítélése emberenként és alkalmazásonként, felhasználástól függően változik. Az egyes konkrét esetekben a vélemények mégis legtöbbször megegyeznek; úgy tűnik, léteznie kell valamiféle többé–kevésbé általános érvényű, objektív meghatározásnak. Mi az tehát, ami megkülönbözteti az elsőosztályú, ún. *Class A* felületeket a többitől?

A fair felületekkel szemben támasztott követelmények egy része szinte magától értetődő. Egy jó minőségű modelltől elvárjuk, hogy *sima* legyen, a felület a tükröződést ne torzítsa, és a visszavert fény egyenletesen változzon, ha mozgatjuk a nézőpontot. Egy életközeli példát véve, ha ezek teljesülnek, akkor például egy, az úttesten haladó gépkocsi motorháztetején az aszfaltra felfestett jelek képe egyenletesen fut végig, míg ha a felület egyenetlen, akkor a jelek meghajlanak, szétfolynak, ezzel csökkentve a vizuális élményt.

1.2. Vizuális eszközök

Felületek készítése, javítása során különböző vizsgálati eszközök (*interrogation tool*) léteznek, amelyek arra hivatottak, hogy segítsék a tervezőt a szabad szemmel nehezen észrevehető egyenetlenségek megtalálásában. Ezek az algoritmusok a modell megjelenítését olyan módon módosítják, hogy annak egy–egy tulajdonsága kiemelődik ill. láthatóvá válik. Ilyenek például a tükröződési vonalak (*reflection lines*) és a fényvonalak (*isophote lines*), amelyek az imént vázolt követelményeken alapulnak.

[2] Nehéz magyar megfelelőt találni a *fair* (szép) szóhoz. A jelentés széles spektrumának érzékeltetése céljából a továbbiakban az eredeti szó mellett az *egyenletes görbületeloszlású*, *tökéletesen görbült*, *sima*, ill. *esztétikus* kifejezéseket használjuk.

A tükröződési vonalakat a valóságban is látni lehet, ha a tárgyat egy párhuzamos neoncsövekkel megvilágított terembe helyezzük; rossz minőségű felületre utal, ha a fényforrások képe megtörik vagy erősen hullámzik. Mivel a visszavert fény iránya a normálvektoroktól függ, ez az eszköz a normálvektormező hibáit hozza ki. Fényvonalaknak azokat a halmazokat nevezzük, amelyeken a felületi normálisok megközelítőleg egyazon irányba mutatnak. Mind a tükröződési, mind a fényvonalak főleg az első deriváltaktól függnek, ezért hasonló eredményeket adnak, de a fényvonalakat, illetve annak még egyszerűbb változatát, a fénysávokat (*highlight band*) általában jobban kedvelik, mert könnyebben számolhatóak [1].

Széles körben használt vizsgálati eszköz még a görbülettérkép (*curvature map*) és a görbületi fésű (*curvature comb*). Ezek a második deriváltakon alapulnak és a görbület mértékét jelenítik meg: az előbbi az értékeket egy egyenletesen változó színskálával, az utóbbi egyenes vonalakkal, „fésűfogakkal" reprezentálja. A fésű ugyan csak görbékre – a felület (általában hozzávetőlegesen merőleges) síkmetszeteire – használható, de intuitívabb a görbülettérképnél.

Jogosan merül fel a kérdés, hogy miért tulajdonítanak olyan nagy fontosságot a görbületnek, amely meg sem jelenik a valóságban? A tapasztalat azt mutatja, hogy az esztétikusnak talált modellek nagy részének görbülete monoton, s még ha ez olyan látványosan nem is fogható meg, mint pl. a tükröződési vonalak esetében, kétségtelen, hogy részét képezi szépségérzetünknek.

A fenti eszközök nagy segítséget nyújtanak a tervezési folyamatban, de a felület hibáinak kijavítása így is egy hosszú, manuális folyamat, amely nagy gyakorlatot igényel. Korántsem egyértelmű ugyanis, hogy milyen változtatásokkal lehet elérni egy–egy egyenetlen rész simítását, és hogy ezek a módosítások milyen mellékhatásokkal járnak. Szükség van ezért olyan algoritmusokra, amelyek kevés felhasználói beavatkozással, automatikusan szépítik a modellt.

1.3. Simasági mértékek

A fairing algoritmusok két nagy csoportra oszthatók: a jelen diplomamunka tárgyát képező utólagos, javító módszerekre és az ún. variációs eljárásokra, amelyek a felület létrehozásába integrálják a szépítést. Ezek különféle kényszereket határoznak meg, majd a fennmaradó szabadsági fokokat egy simasági mérték (*fairness measure* vagy *smoothness measure*) minimalizálására használják [2, 9].

Megjegyzendő, hogy „simaság" alatt itt nem a hétköznapi értelemben vett simaságot, hanem annál magasabbrendű, matematikai simaságot értünk. A simítandó modell az esetek túlnyomó részében több nagy, elsődleges funkcionális felületből, és kisebb, ezeket összekötő (pl. lekerekítő) felületből áll. A „simaság" ilyenkor nem csak a nagy felületek önmagukban vett simaságát, hanem a felületek közti kapcsolódások egyenletességét is jelenti, gyakori elvárás pl. a másodrendű folytonosság.

A simasági mérték lényeges eleme nem csak a variációs, de minden fairing módszernek. Esztétikánkat formulába kell öntenünk, hogy megértethessük a számítógéppel, és az bírálatot tudjon adni – a szépséget számokkal kifejezve – egy felület kinézetéről.

Azt szeretnénk, hogy a formula a reprezentációtól függetlenül ugyanazt az értéket adja az azonosan kinéző esetekben, azaz megköveteljük, hogy legyen paraméterfüggetlen. Azt mondjuk, hogy egy S felület fair, ha

$$\mathcal{F}(S) < \tau$$

teljesül, ahol τ egy, a felhasználó által meghatározott tolerancia. Egy abszolút mérték segítségével a folyamat még automatikusabbá tehető lenne, de ez legfeljebb lokálisan érhető el, mivel globálisan a számértéket befolyásolják a felület alaksajátosságai (*feature*) is, azok a jellegzetességek, amelyek miatt felismerhető a modell. Egy emberi arc modellezésénél például az orr egy fontos ismertetőjel, de éles kiemelkedése miatt megemeli a legtöbb simasági mérték értékét.

A szakirodalom nagy része Bézier és (egyszerű ill. racionális) B-spline görbékkel és felületekkel foglalkozik [4]. A B-spline-ok, konstrukciójuknál fogva, nem igazán alkalmasak manuális javításokra. A csomók (*knot*) mozgatása a felületeknél globális változást okoz, a kontrollpontok elmozdításának hatása nehezen követhető, különösen kis módosítások esetén. Numerikusan azonban a B-spline-ok könnyebben kezelhetőek és a gyakorlatban igen elterjedtek, ezért főleg ezekkel fogunk foglalkozni.

1.4. A dolgozat felépítése

A következő fejezetben a szakirodalomban fellelhető legfontosabb mértékek és simító algoritmusok kerülnek bemutatásra mind az egyszerűbb, kétdimenziós görbék, mind a háromdimenziós felületek esetében; az itt felmerülő problémákra keres megoldást a harmadik fejezet. A negyedik fejezet ismerteti a NURBS felületek szépítése céljából kifejlesztett tesztkörnyezetet és szemlélteti elvégzett kísérleteimet, majd végül az ötödik fejezet összefoglalja az eredményeket.

2. fejezet

Szakirodalmi áttekintés

A felületek görbületének vizsgálata előtt érdemes először a görbék simaságával kapcsolatos ismereteket áttekinteni. Egyrészt strukturális szempontból fontos ez, hiszen a felületekre vonatkozó eredmények szinte mind általánosításai vagy átfogalmazásai görbékre kimondott tételeknek ill. módszereknek, másrészt kronológiai, történeti szempontból: az esztétikus görbék kutatása már az 1960-as években elindult, elsősorban a gépjárműiparnak köszönhetően, bár ezek eredményeit sokáig titokban tartották [3].

Korábban voltak már próbálkozások térbeli modelleket simító olyan eljárásokra, amelyek nem görbéken alapultak (pl. egy viszonylag új és érdekes ötlet a jelfeldolgozás alkalmazása poligonhálókra [17]), de a szakirodalom döntő többsége a hagyományos módszert követi, így itt is ezt a felépítést használjuk.

Roulier és Rando rámutatott arra, hogy noha jelenleg nem reméljük, hogy lesz egy általános érvényű mérték vagy módszer a görbület optimalizálására, fontos, hogy egyre újabbakat definiáljunk és azok hatását minél pontosabban meghatározzuk; ezáltal a tervezők könnyen ki tudják választani a feladathoz megfelelőt [16].

A következőkben először a görbékre, majd a felületekre vonatkozó simasági mértékeket nézzük meg, amit az algoritmusok bemutatása és végül egy rövid összefoglaló követ majd.

2.1. Mértékek görbékre

Egy természetesen adódó mérték a feszítési energia (*strain energy*), amely egy hajók tervezésénél alkalmazott XVIII. századi görberajzolási technikán alapszik. Adott pontokon átmenő sima görbe meghatározásához a pontokra fémből készült súlyokat helyeztek és egy vékony, rugalmas farudat (*spline*) csúsztattak közéjük. Az így kapott görbe általában esztétikus; a módszert ma is használják.

A farúd olyan helyzetet vesz fel, amely minimalizálja a feszítési energiát, tehát ha a rudat egy $\mathbf{c}$ görbével reprezentáljuk, akkor a mérték, amit kapunk:

$$E = \int (\kappa(s))^2 \mathrm{d}s, \tag{2.1}$$

ahol $\kappa(s)$ a görbe ívhossz szerinti görbületét jelöli. A képletet megvizsgálva látszik, hogy ezzel a görbületnek a görbe teljes hosszán vett átlagát csökkentjük, miközben a kiugrásait a négyzetreemeléssel „büntetjük". Következésképp E minimalizálása kiegyenlíti a görbületet, miközben próbálja az átlagot minél lejjebb szorítani [16].

Mivel a görbület kiszámítása nehézkes, a (2.1) energiát sokszor egy egyszerűbb kifejezéssel közelítik [3]:

$$\hat{E} = \int (\mathbf{c}''(t))^2 \mathrm{d}t. \tag{2.2}$$

A harmadfokú interpoláló spline, míg a kapcsolódási pontokban megtartja a C^2 folytonosságot, minimalizálja $\hat{E}$-t, de ebből ívhossz szerinti paraméterezéstől jelentősen eltérő esetben nem következik „ideális" görbületeloszlás.

Sem (2.1), sem (2.2) nem bünteti a görbület előjelének változását, ezért a görbékben gyakran nem kívánt inflexiók jelennek meg. Ennek elkerülésére a spline görbének több változata is született, mint pl. a feszített spline (*spline in tension*) vagy a ν-spline [3]. A feszített spline a $\mathbf{c}'' - \alpha \mathbf{c}'$ mennyiség lineáris változását követeli meg; az újonnan bevezetett α paraméter gyakorlatilag azt a hatást kelti, mintha a farúd két végét valamennyire meghúznánk. Ha α nulla, visszakapjuk a harmadfokú spline-t, ha tart a végtelenhez, akkor a görbe tart a pontok lineáris interpoláltjához.

Ennél könnyebben kezelhető a szakaszonként polinomiális ν-spline, amely hasonló hatást ér el a C^2 folytonosság G^2-re gyengítéséért cserébe. Minden szegmensre bevezetve egy ν_i feszítési paramétert a mérték így változik:

$$E_\nu = \int (\mathbf{c}''(t))^2 \mathrm{d}t + \sum_i \nu_i (\mathbf{c}'(u_i))^2, \tag{2.3}$$

ahol u_i a kapcsolódási pontok paraméterértékét jelöli.

A (2.1) mérték másik változatát használja a Moreton és Séquin [13] által javasolt minimális variációjú görbe (*Minimum Variation Curve, MVC*), melyben a görbület helyett annak deriváltja szerepel:

$$E_{MVC} = \int (\kappa'(s))^2 \mathrm{d}s. \tag{2.4}$$

$$\boxed{\begin{aligned} \mathbf{t} &= \mathbf{c}' \\ \mathbf{t}' &= \kappa\mathbf{n} \\ \mathbf{n}' &= -\kappa\mathbf{t} + \tau\mathbf{b} \\ \mathbf{b}' &= -\tau\mathbf{n} \end{aligned}}$$

2.1. táblázat. A Frenet–Serret formulák

A MVC egyik előnye, hogy konvexitástartó, tehát biztosan nem tartalmaz felesleges inflexiós pontokat. Másrészt előnyben részesíti a köríveket, amelyek tökéletesen simának számítanak.

Más módon definiálja az esztétikus görbéket Farin és Sapidis [3]: egy görbe akkor sima, ha a(z ívhossz szerinti) görbületének grafikonja aránylag kevés monoton darabból áll. A gondolat e mögött az, hogy egy sima görbe görbületének csak ott legyen lokális szélsőértéke, ahol a tervező ezt szeretné. Az előzőekkel összehasonlítva az egyik fontos különbség az, hogy ez egy lokális mérték; a folytonossági hibákat összegezve kaphatunk belőle globálisat.

Mitől lesz jó egy mérték? Hogyan lehet megállapítani a hatását? Ezekre a kérdésekre a választ [16] adja meg, amely áttekinti a legfontosabb görbületi mértékeket és egy módszert ad új mértékek létrehozására. Vezessük be a görbe előjeltől eltekintve paraméterfüggetlen, invariáns jellemzőire a következő jelöléseket: κ – görbület, ρ – görbületi sugár, τ – torzió, $\mathbf{n}$ – normális egységvektor, $\mathbf{t}$ – érintő egységvektor, $\mathbf{b}$ – binormális egységvektor és $S_\mathbf{c} = \int_a^b |\mathbf{c}'(t)|\mathrm{d}t$ jelölje az ívhosszt. Ezekre a mennyiségekre ívhossz szerinti paraméterezés esetén teljesülnek a Frenet–Serret formulák (2.1. táblázat), melyek kényelmessé teszik a számolást.

A módszer lényege, hogy az eredeti görbe alapján a fenti mennyiségek segítségével készítünk egy másik, ún. származtatott görbét (*derived curve*, $\mathbf{d}$), és ennek az ívhosszát tekintjük simasági mértéknek. Például, ha a görbéhez a görbületi középpontokból álló görbét (síkgörbe esetén ez az evolúta) rendeljük, azaz

$$\mathbf{d}(t) = \mathbf{c}(t) + \rho(t)\mathbf{n}(t),$$

akkor azt várjuk, hogy a származtatott görbe ívhosszának csökkentése közelebb húzza a görbületi középpontokat és így a görbe körívhez fog hasonlítani. Kis számolással megkapjuk, hogy

$$\mathcal{F}(\mathbf{c}) = S_\mathbf{d} = \int_0^{S_\mathbf{c}} (\rho^2\tau^2 + (\rho')^2)^{1/2}\mathrm{d}s, \tag{2.5}$$

amelyből már jobban látszik, mi is a pontos hatás: τ^2 minimalizálása a görbét próbálja egy síkba hozni, a ρ^2 tag juttatja előnyhöz a körívet, míg a $(\rho')^2$

Származtatott görbe	Mérték ($\mathcal{F}(\mathbf{c}) = S_{\mathbf{d}}$)	$\mathbb{R}^2$	$\mathbb{R}^3$
$\mathbf{c} + \rho\mathbf{n}$	$\int_0^{S_{\mathbf{c}}}(\rho^2\tau^2 + (\rho')^2)^{1/2}\mathrm{d}s$	simít	simít lekerekít
$\kappa\mathbf{b}$	$\int_0^{S_{\mathbf{c}}}(\kappa^2\tau^2 + (\kappa')^2)^{1/2}\mathrm{d}s$	simít	simít lapít
$\kappa\mathbf{n}$	$\int_0^{S_{\mathbf{c}}}(\kappa^2\tau^2 + (\kappa')^2 + \kappa^4)^{1/2}\mathrm{d}s$	simít egyenesít	simít lapít
$\kappa\mathbf{t}$	$\int_0^{S_{\mathbf{c}}}(\kappa^4 + (\kappa')^2)^{1/2}\mathrm{d}s$	simít egyenesít	simít lapít
$\rho\mathbf{b}$	$\int_0^{S_{\mathbf{c}}}(\rho^2\tau^2 + (\rho')^2)^{1/2}\mathrm{d}s$	simít	simít lekerekít
$\mathbf{n}$	$\int_0^{S_{\mathbf{c}}}(\tau^2 + \kappa^2)^{1/2}\mathrm{d}s$	egyenesít	lapít
$\rho^2\mathbf{n}$	$\int_0^{S_{\mathbf{c}}}(4\rho^2(\rho')^2 + \rho^4\tau^2 + \rho^2)^{1/2}\mathrm{d}s$	simít lekerekít	simít lekerekít
$\rho^2\mathbf{t}$	$\int_0^{S_{\mathbf{c}}}(4\rho^2(\rho')^2 + \rho^2)^{1/2}\mathrm{d}s$	simít lekerekít	simít lekerekít

2.2. táblázat. A Roulier–Rando-féle mértékek görbékre

kiküszöböli az éles kanyarokat. Viszont mivel $\rho = 1/\kappa$ és inflexiós pontban $\kappa = 0$, bizonyos esetekben ez a mérték nem használható. A 2.2. táblázat összefoglalja a cikkben tárgyalt összetett mértékeket és azok hatásait. Ezek természetesen tetszőlegesen kombinálhatóak is.

2.2. Mértékek felületekre

Hasonlóan a feszítési energiához, felületek esetén egy vékony, rugalmas lemez energiájáról (*thin plate energy*) beszélhetünk. Ha κ_1 és κ_2 egy S felület főgörbületei, akkor az energia [7]:

$$\Pi_P = \iint_S a(\kappa_1^2 + \kappa_2^2) + 2(1-b)\ \kappa_1\kappa_2\ \mathrm{d}S, \tag{2.6}$$

ahol a és b a lemez anyagától függő konstansok, melyek szokásos értéke $a = 1$ és $b = 0$ vagy $b = 1$. A fenti képletet gyakran közelítik a következő, egyszerűbb kifejezéssel:

$$\Pi = \iint_A S_{uu}^2 + 2S_{uv}^2 + S_{vv}^2\ \mathrm{d}u\ \mathrm{d}v, \tag{2.7}$$

$E = S_u S_u$	$L = \mathbf{n} \cdot S_{uu}$
$F = S_u S_v$	$M = \mathbf{n} \cdot S_{uv}$
$G = S_v S_v$	$N = \mathbf{n} \cdot S_{vv}$
($\mathbf{n}$ a felületi normális)	

2.3. táblázat. A Gauss-féle első- és másodrendű főmennyiségek együtthatói

amely izometrikus paraméterezés esetén ad pontos értéket, tehát ha a paraméterirányok merőlegesek és a felület mindkét irányban ívhossz szerint paraméterezett.

Szintén elterjedt, de főként háromszöghálók simításánál használt mérték a membrán energia (*membrane energy*):

$$\Pi_m = \iint_A S_u^2 + S_v^2 \,\mathrm{d}u\,\mathrm{d}v. \tag{2.8}$$

Moreton és Séquin [13] itt is a variációs technikát javasolja: a főgörbületeket a főirányokban deriválva megkapjuk a minimális variációjú felületek funkcionálját:

$$\Pi_{MVS} = \iint_S \left(\frac{\partial \kappa_1}{\partial e_1}\right)^2 + \left(\frac{\partial \kappa_2}{\partial e_2}\right)^2 \mathrm{d}S, \tag{2.9}$$

amely gömbökre, kúpokra és tóruszokra eltűnik. Ezt a paramétertérben felírva [12] szerint:

$$\iint_A \left[\left(\frac{\partial \kappa_1}{\partial e_1}\right)^2 + \left(\frac{\partial \kappa_2}{\partial e_2}\right)^2\right] \|S_u \times S_v\| \,\mathrm{d}u\,\mathrm{d}v,$$

ahol $\|S_u \times S_v\| = \sqrt{EG - F^2}$ és E, F, G az elsőrendű Gauss-féle főmennyiség együtthatói (ld. a 2.3. táblázatot). A nem lineáris főgörbületek miatt sajnos ez még így is nehezen számolható.

Westgaard és Nowacki [18] összefoglalja a felületek invariáns tulajdonságait, jelöljük most ezeket az alábbi módon:

$K_2 = \kappa_1 \cdot \kappa_2$	Gauss-görbület	(*Gaussian curvature*)
$H_2 = (\kappa_1 + \kappa_2)/2$	átlagos görbület	(*mean curvature*)
$A_2 = \lvert\kappa_1\rvert + \lvert\kappa_2\rvert$	abszolút görbület	(*absolute curvature*)
$T_2 = \kappa_1^2 + \kappa_2^2$	főgörbület-norma	(*principal curvature norm*)

Bevezeti ezeken kívül 3-as indexszel ezen értékek variációit, 4-es indexszel pedig a variációk variációit, majd ezeket közelítő (paraméterfüggő) kvadratikus alakra hozza és a felület második, harmadik és negyedik paraméterirányú deriváltjaiból készít (2.7) jellegű mértékeket.

Származtatott felület		**Hatás**	
i)	$K_2\mathbf{n}$	lapítás	(*flattening*)
ii)	$S+(H_2/K_2)\mathbf{n}$	lekerekítés	(*rounding*)
iii)	$(K_2+H_2^2)\mathbf{n}$	görgetés	(*rolling*)
iv)	$H_2\mathbf{n}$	görgetés	(*rolling*)
v)	$\mathbf{n}/T_2$	görgetés lapítás nélkül	(*rolling w/o flat.*)

2.4. táblázat. A Roulier–Rando-féle származtatott felületek

Hahmann [8] a két paraméterirányra külön–külön mértéket vezet be:

$$z^u(u_0,v_0)=\frac{|\frac{\partial g}{\partial u}(u_0,v_0)|}{\|\frac{\partial S(u,v)}{\partial u}\|},\ z^v(u_0,v_0)=\frac{|\frac{\partial g}{\partial v}(u_0,v_0)|}{\|\frac{\partial S(u,v)}{\partial v}\|}, \tag{2.10}$$

ahol g a főgörbületek valamilyen függvénye, pl. K_2 vagy T_2.

Végül [16] kiterjeszti az új mértékek származtatásának módszerét felületekre, a mérték ekkor az S felülethez rendelt D származtatott felület felszíne lesz:

$$\mathcal{F}(S)=\Gamma_D=\iint_A\left|\frac{\partial D}{\partial u}\times\frac{\partial D}{\partial v}\right|\ \mathrm{d}u\ \mathrm{d}v. \tag{2.11}$$

A mértékek vizsgálatához érdemes áttérni egy olyan (s,t) paraméterezésre, amelyre teljesül, hogy

$$\left|\frac{\partial S}{\partial s}\times\frac{\partial S}{\partial t}\right|=1,$$

mivel így a felületi normális megadható $\mathbf{n}=\frac{\partial S}{\partial s}\times\frac{\partial S}{\partial t}$ alakban és teljesülnek a Rodrigues-formulák:

$$\frac{\partial\mathbf{n}}{\partial s}=-\kappa_1\frac{\partial S}{\partial s},\qquad\frac{\partial\mathbf{n}}{\partial t}=-\kappa_2\frac{\partial S}{\partial t}.$$

A 2.4. táblázat foglalja össze a cikkben szereplő származtatott felületeket. Nézzük meg az ezek által létrejövő mértékeket!

i) A felszínt kiszámolva a következő mértéket kapjuk:

$$\iint_A|K_2|\left[\left(\kappa_1\frac{\partial K_2}{\partial t}\right)^2+\left(\kappa_2\frac{\partial K_2}{\partial s}\right)^2+K_2^4\right]^{1/2}\mathrm{d}s\ \mathrm{d}t \tag{2.12}$$

Mivel ez a Gauss-görbületet és a görbületi vonalak menti változást bünteti, minimalizálása lapossá teszi a felületet.

ii) A lekerekítő mérték képlete:

$$\iint_A \left[(1-Q\kappa_1)^2(1-Q\kappa_2)^2 + (1-Q\kappa_1)^2\left(\frac{\partial Q}{\partial t}\right)^2 \right. \\ \left. + (1-Q\kappa_2)^2\left(\frac{\partial Q}{\partial s}\right)^2\right]^{1/2} \mathrm{d}s\ \mathrm{d}t, \tag{2.13}$$

ahol $Q = H_2/K_2$. Értéke főleg $(1-Q\kappa_1)$-től és $(1-Q\kappa_2)$-től függ, amelyek $\kappa_1 = \kappa_2$ esetén tűnnek el, így a mérték a felületet egy gömbfelület felé húzza.

iii) Hasonlóan látható, hogy

$$\iint_A |W| \left[\left(\kappa_1 \frac{\partial W}{\partial t}\right)^2 + \left(\kappa_2 \frac{\partial W}{\partial s}\right)^2 + W^2 K_2^2\right]^{1/2} \mathrm{d}s\ \mathrm{d}t, \tag{2.14}$$

ahol $W = K_2 + H_2^2$, henger- ill. kúpfelszínt próbál létrehozni.

iv) Kicsit összetettebb az alábbi:

$$\iint_A |H_2| \left[\left(\kappa_1 \frac{\partial H_2}{\partial t}\right)^2 + \left(\kappa_2 \frac{\partial H_2}{\partial s}\right)^2 + H_2^2 K_2^2\right]^{1/2} \mathrm{d}s\ \mathrm{d}t. \tag{2.15}$$

Itt attól függően, hogy H_2-t, K_2-t vagy (pl. $\kappa_1 \neq 0$ esetén) $\frac{\partial H_2}{\partial t}$-t csökkentjük, sík-, henger- vagy kúpfelszínhez fog tartani.

v) Végül az utolsó:

$$\iint_A |R| \left[\left(\kappa_1 \frac{\partial R}{\partial t}\right)^2 + \left(\kappa_2 \frac{\partial R}{\partial s}\right)^2 + R^2 K_2^2\right]^{1/2} \mathrm{d}s\ \mathrm{d}t, \tag{2.16}$$

ahol $R = 1/T_2$. Megmutatható, hogy különböző feltételek mellett görgető ill. lekerekítő hatása van, de nem lapítja le a felületet. Akkor is használható, ha κ_1 vagy κ_2 (de nem mindkettő) 0.

2.3. Algoritmusok görbékre

B-spline-ok simítására az egyik legegyszerűbb, jól ismert módszer a csomók kivétele és visszaillesztése (*knot removal and reinsertion, KRR*). Az algoritmus eredetileg Kjellandertől származik, Sapidis és Farin ennek egy lokális változatát dolgozták ki [2]. Legyen a k-adrendű ($k-1$ fokú) görbe

$$\mathbf{x}(t) = \sum_{i=0}^{n} \mathbf{d}_i \cdot N_{i,k}(t), \quad t \in [t_{k-1}, t_{n+1}] \tag{2.17}$$

alakú, ahol d_i-k a kontrollpontok, $N_{i,k}$ a B-spline bázis és $T = (t_j)_{j=0}^{n+k}$ a csomók sorozata, amelyből $t_k, \ldots, t_n$ a belső csomók (*inner knots*). Ekkor $\mathbf{x}$ a csomópontokban legfeljebb C^{k-2}-folytonos.

Csomók hozzáadása a sorozathoz megtehető egyértelműen úgy, hogy a görbe alakja ne változzon, ehhez elegendő egy egyszerű mátrixszorzás a kontrollpontok vektorán. Ugyanakkor egy csomó elvétele csak akkor hagyja változatlanul a görbét, ha az a csomópontban C^{k-1}-folytonos volt, mivel ekkor az általa elválasztott két szegmens valójában eggyel is leírható.

A feladat tehát az

$$\tilde{\mathbf{x}} = \sum_{i=0}^{n-1} \tilde{\mathbf{d}}_i N_{i,k,\tilde{T}} \qquad (\tilde{T} \subset T)$$

görbe $\tilde{\mathbf{d}}_i$ kontrollpontjait úgy meghatározni, hogy $\mathbf{d} = A\tilde{\mathbf{d}}$ teljesüljön, ahol A a csomóbeszúrás mátrixa [8]. Erre több közelítő megoldás született, mint pl. $\|\tilde{\mathbf{x}} - \mathbf{x}\|$ minimalizálása, mi azonban lokális algoritmust szeretnénk, tehát olyat, amely minél kevesebb kontrollpontot változtat.

Farin [3] harmadfokú esetben ezt mindössze egy pont elmozgatásával oldja meg, úgy, hogy a csomópontban C^3 folytonosságot alakít ki. A t_j csomó elvételekor $\mathbf{d}_j$-n a következő affin transzformációt alkalmazza:

$$\tilde{\mathbf{d}}_j = \frac{(t_{j+2} - t_j)\mathbf{l}_j + (t_j - t_{j-2})\mathbf{r}_j}{t_{j+2} - t_{j-2}},$$

ahol $\mathbf{l}_j$ és $\mathbf{r}_j$ az alábbi módon adott:

$$\mathbf{l}_j = \frac{(t_{j+1} - t_{j-3})\mathbf{d}_{j-1} - (t_{j+1} - t_j)\mathbf{d}_{j-2}}{t_j - t_{j-3}},$$
$$\mathbf{r}_j = \frac{(t_{j+3} - t_{j-1})\mathbf{d}_{j+1} - (t_j - t_{j-1})\mathbf{d}_{j+2}}{t_{j+3} - t_j}.$$

Ez csak a $[t_j, t_{j+4}]$ intervallumban változtatja a görbét és így a lehető leginkább lokális algoritmus. Hátránya, hogy a régi és az új spline közti különbség általában nagyobb, mint pl. a legkisebb négyzetes különbség (*least squares knot removal*) módszernél, amely a $\mathbf{d} = A\mathbf{d}$ egyenlet egy 3×2-es alrendszerének legkisebb négyzetes megoldását adja meg. Ez utóbbi legfeljebb 3 kontrollpontot mozgat, tehát csak a $[t_{j-2}, t_{j+4}]$ intervallumban történik változás [8].

Farin és Sapidis a KRR-t a $z_j = |\kappa'(t_j^+) - \kappa'(t_j^-)|$ mértékkel párosítja, amely t_j kivétele és visszaillesztése után nulla lesz, hiszen $\tilde{\mathbf{x}}(t)$ C^3-folytonos t-ben. A legnagyobb javulás érdekében a KRR-t a z_j mérték szerinti „legbántóbb csomóra" (*most offending knot*) alkalmazzuk. Az algoritmus tehát a következő:

`ciklus`

1. a $\xi = \sum z_i$ globális mérték kiszámolása
2. t_j meghatározása, ahol $z_j = \max(z_i)$
3. t_j kivétele és visszaillesztése

`amíg` *feltétel*

A feltétel meghatározása nem triviális: egy olyan csomópont-sorozatot szeretnénk, amelynek elemeire a fenti módszert sorban alkalmazva a ξ globális minimumához jutunk el.

Eck és Hadenfeld [2] a (2.2) mértéket lokálisan minimalizálja úgy, hogy egy kivételével minden kontrollpontot lerögzít. Második derivált helyett harmadikat is ajánl, tehát általánosan az

$$E_l = \int_{t_{k-1}}^{t_{n+1}} \left(\tilde{\mathbf{x}}^{(l)}(t)\right)^2 \quad l = 2,3 \tag{2.18}$$

mértéket minimalizálja, miközben az eredeti görbétől való eltérést egy δ tolerancia alatt tartja:

$$\max\{|\mathbf{x}(t) - \tilde{\mathbf{x}}(t)| \mid t \in [t_{k-1}, t_{n+1}]\} \leq \delta. \tag{2.19}$$

A B-spline konvex héja miatt a fenti megszorításhoz elég a $|\mathbf{d}_r - \tilde{\mathbf{d}}_r| \leq \delta$ egyenlőtlenségnek teljesülnie. Ha $\tilde{\mathbf{d}}_r$ kívül esik $\mathbf{d}_r$ δ-sugarú környezetén, akkor a két pontot összekötő szakasz és a δ sugarú gömb metszéspontját választjuk az új $\tilde{\mathbf{d}}_r$ pontnak, azaz

$$\tilde{\mathbf{d}}_r^* = \mathbf{d}_r + \delta \cdot \frac{\tilde{\mathbf{d}}_r - \mathbf{d}_r}{|\tilde{\mathbf{d}}_r - \mathbf{d}_r|}.$$

A minimalizáció egy $\tilde{\mathbf{d}}_r = \sum_{i \neq r} \gamma_i \cdot \mathbf{d}_i$ alakú megoldáshoz vezet, ahol a γ_i súlyok kvadratúrák segítségével számolhatóak. Bevezetve a

$$z_r = E_l(\mathbf{d}_r) - E_l(\tilde{\mathbf{d}}_r) = (\mathbf{d}_r - \tilde{\mathbf{d}}_r)^2 \cdot \int \left(N_{r,k}^{(l)}(t)\right)^2 \mathrm{d}t \tag{2.20}$$

mértéket, az algoritmus:

`ciklus`

1. z_i-k kiszámolása $0 \leq i \leq n$-re.
2. z_r meghatározása, ahol $z_r = \max\{z_i \mid 0 \leq i \leq n\}$
3. $\tilde{\mathbf{d}}_r$, és ebből $\tilde{\mathbf{d}}_r^*$ kiszámolása

`amíg` *feltétel*

Végül megemlítendő még Renka [15] Sobolev gradiens alapú módszere, amely tetszőleges simasági mértékre alkalmazható. Ez a legmeredekebb leszállást (*steepest descent*) alkalmazza Sobolev terek normájával, amelyek sokkal érzékenyebben reagálnak a függvény hirtelen változásaira. A nem kívánt hurkok elkerülése érdekében a mértékhez súlyozva hozzáadja a görbe hosszának négyzetét.

2.4. Algoritmusok felületekre

Lássuk először a KRR kiterjesztését felületekre. Legyen X egy (k, l) rangú B-spline felület az

$$X(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{d}_{ij} N_{i,k,U}(u) N_{j,l,V}(v),\ (u,v) \in [u_{k-1}, u_{n+1}] \times [v_{l-1}, v_{m+1}] \tag{2.21}$$

alakban, ahol $U = (u_i)_{i=0}^{n+k}$ és $V = (v_j)_{j=0}^{m+l}$ jelölik a csomósorozatokat. Itt egy KRR lépés egy teljes csomóvonal (*knot line*) elvételét és visszaillesztését jelenti, tehát ha pl. kivesszük a v_s csomópontot, akkor a görbéknél bemutatott algoritmust alkalmazni kell külön–külön minden $\mathbf{x}_i = \sum_{j=0}^{m} \mathbf{d}_{ij} N_{j,l,V}(t)$ B-spline-ra, ahol $i = k, \ldots, n$. Fontos megjegyezni, hogy a kiterjesztett KRR nem biztosít C^3 folytonosságot, csak az egyik irányban, és ez egy u és v irányú KRR egymás utáni alkalmazásával sem érhető el. A módszer ebben a formájában nem mondható lokálisnak sem, de bizonyítható, hogy a visszahelyezett csomópontban való, adott irányú C^3 folytonossághoz elég három görbére elvégezni a KRR-t [8]. Az algoritmus menete ettől eltekintve megegyezik az előző pontban leírttal, mértékként a csomópontokban vett (2.10) mértéket használva.

Ideje, hogy pár szót szóljunk a leállási feltételről. Legegyszerűbb addig folytatni az algoritmust, amíg a felület javul (a mérték csökken), vagy amíg a javulás egy előre meghatározott határ felett van. Ekkor viszont könnyen lehet, hogy lokális minimumot fogunk kapni. Hahmann két megoldást javasol: a legjobb első keresését (*best-first-search*) és a szimulált hűtést (*simulated annealing*) [14].

A legjobb első keresése megkeresi azt a belső csomópontpárt, amelyikre a legtöbbet javul a mérték és az erre alkalmazott KRR-ből keletkező felület csomópontjait megint végignézi – így folytatja a keresést egyészen egy megadott k szintig.

A szimulált hűtés egy bizonyos valószínűséggel elfogad olyan lépéseket is, amelyekben növekszik (romlik) a simaság mértéke, de ez a valószínűség egy előre definiált ütemben csökken. Az algoritmus akkor áll le, amikor már k

rossz lépést nem fogadott el.

B-spline-ok kevert harmadik deriváltjai ($X_{u^\mu v^\nu}$, ahol $\mu, \nu \geq 1$ és $\mu+\nu = 3$) mindig folytonosak. Ezt használja ki az a módszer, amely X_{uuu} u-irányú, és X_{vvv} v-irányú differenciáját veszi lokális mértéknek egy (u_k, v_l) csomópontban [9]:

$$E_{kl} = \|\Delta_{uuu}(u_k, v_l)\|^2 + \|\Delta_{vvv}(u_k, v_l)\|^2, \text{ ahol}$$

$$\Delta_{uuu}(u_k, v_l) = X_{uuu}(u_k^-, v_l) - X_{uuu}(u_k^+, v_l) = \sum_{i=k-4}^{k} \sum_{j=l-3}^{l-1} \alpha_{ij} \mathbf{d}_{ij},$$

$$\Delta_{vvv}(u_k, v_l) = X_{vvv}(u_k, v_l^-) - X_{vvv}(u_k, v_l^+) = \sum_{i=k-3}^{k-1} \sum_{j=l-4}^{l} \beta_{ij} \mathbf{d}_{ij}.$$

Itt α_{ij} és β_{ij} csak a csomósorozatoktól függnek és előre kiszámolhatóak. A képletben részt vevő 21 kontrollpontra mindössze két egyenletünk van: $\Delta_{uuu} = 0$ és $\Delta_{vvv} = 0$. A külső 12 kontrollpont lerögzítésével elérhetjük, hogy a felület ne nagyon torzuljon, a maradék 9 pontra pedig a Lagrange szorzók módszerét alkalmazva kapunk egy megoldást. Egyenletes csomósorozatú felület esetén az ehhez szükséges mátrixot csak egyszer kell kiszámolni, tehát a lokális simító lépésre egy explicit megoldást kapunk.

Hadenfeld [7] a (2.7) mérték segítségével ismét egy olyan módszert ad, amely csak egy kontrollpontot mozgat. A megoldás – hasonló módon – $\tilde{\mathbf{d}}_{r_1 r_2} = \sum_{(i,j)\neq(r_1,r_2)} \gamma_{ij} \mathbf{d}_{ij}$ alakú lesz, és a

$$z_{r_1,r_2} = \Pi(\mathbf{d}_{r_1 r_2}) - \Pi(\tilde{\mathbf{d}}_{r_1 r_2}) \tag{2.22}$$

mértéket bevezetve az algoritmus megegyezik a görbéknél bemutatottal, a távolsági megkötés itt is ugyanúgy kezelhető.

2.5. Motiváció

Ahogyan Moreton és Séquin [13] is megmutatta, a (2.1) ill. (2.6) típusú, tehát pusztán az összgörbületen alapuló mértékek nem igazán jó mutatói a simaságnak. Használatuk nem feltétlenül eredményez szebb görbét ill. felületet, azonban könnyen vezethet az alaksajátosságok deformációjához, eltűnéséhez. Ezzel szemben a [13]-ban javasolt minimális variációjú görbék és felületek ((2.4), (2.9)) a görbület változását, eloszlását javítják, s így igen jól simítanak. E megközelítés hátránya viszont, hogy a görbület deriváltjának kiszámítása bonyolult, és mivel ez nem lineáris, illesztési feladatokban triviálisan nem alkalmazható. Megoldást jelenthetnek a harmadik deriválttal közelítő értékek, ám ezek nem paraméterfüggetlenek és ezért nem várt eredményeket szülhetnek [16].

A globális mértéket minimalizáló simító algoritmusok hibája, hogy elmossák a felületet és így az alaksajátosságokat is. A gyakorlati felhasználás szempontjából ezért egy gyors, lokális fairing algoritmusra van szükség. Ezeknél azonban a globális minimum megtalálása nem garantált, elképzelhető, hogy lokális változtatások sorozatánál jobb eredményt ad, ha több helyen egyszerre módosítunk.

Az Eck–Hadenfeld eljárás paraméterfüggő, s a görbék esetén széles körben alkalmazott, alapvető módszerként számon tartott KRR felületekre való lokális kiterjesztésének hatásossága is erősen függ a paraméterezéstől. Másrészt követendő példának tűnik a két algoritmusnak az a közös gondolata, hogy egy kivételével minden kontrollpontot lerögzít és így minimalizál iteratív módon egy funkcionált.

Ilyen típusú megoldási javaslatot mutatunk be a következő fejezetben.

3. fejezet

Megoldási javaslatok

A Hahmann-féle KRR algoritmus egy aránylag új simító eljárás, amely sok lehetőséget nyújt a kísérletezéshez. Viszont van több hátránya is, amelyeket nem szabad figyelmen kívül hagyni.

Egy u irányú KRR lépés nem eredményez lokális C^3 folytonosságot v irányban, sőt, ronthat is a v irányú simaságon. Ezen kicsit segít, ha a C^3 folytonosság biztosítása helyett mind u, mind v irányban elvégzünk egy–egy KRR lépést. Így az elmozgatott kontrollponthoz két új pozíciót kapunk és természetesen adódik az ötlet, hogy a kontrollpontot az ezeket összekötő szakasz felezőpontjába mozgassuk.

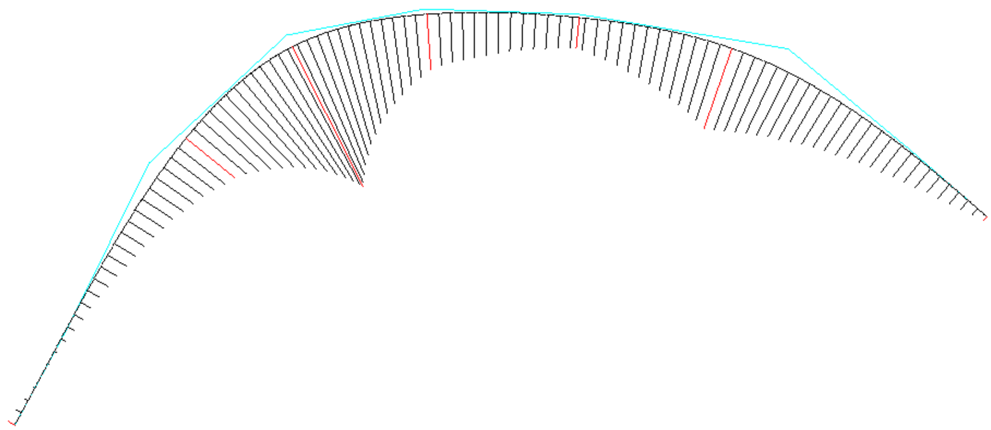
Probléma lehet továbbá, hogy a görbe változására nincsen korlát. Be lehet ugyan vezetni egy toleranciát a kontrollpontok maximális megváltozására az Eck–Hadenfeld eljáráshoz hasonlóan, de ennek hatása nem jósolható meg pontosan.

Sajnos a KRR mindezekkel a változtatásokkal együtt is csak speciális felületekre lesz hatásos. Ennek az az oka, hogy a csomók elhelyezkedése nem mindig áll összhangban a felület geometriai struktúrájával – nem is állhat, hiszen csomóvektorból az egész felületre mindössze kettő jut, s így ezek nem tükrözik teljesen a lokális alaksajátosságokat.

Ilyen megfontolások vezettek arra a következtetésre, hogy egy új megközelítésre van szükség; a következőkben bemutatásra kerülő algoritmushoz hasonló eljárás – tudomásom szerint – eddig csak diszkrét adatok simításával kapcsolatban született [3].

3.1. B-spline fairing

Elsőként vizsgáljuk a síkbeli B-spline görbéket. Egy tökéletes görbületeloszlású görbétől elvárjuk, hogy a görbületi fésűje (3.1. ábra) sima legyen, ne



3.1. ábra. B-spline görbületi fésűvel.

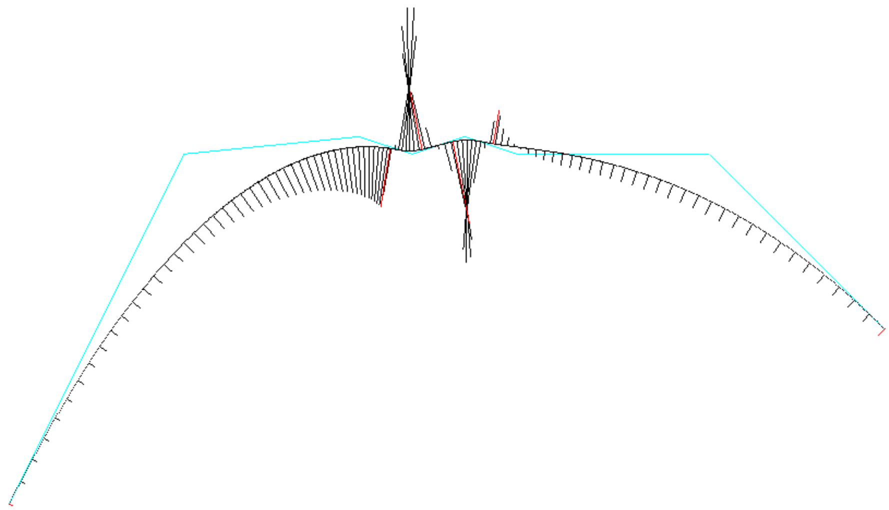
legyenek benne ugrások, törések. A feladat tehát átvihető a fésű csúcspontjai által meghatározott görbére, amely a $\kappa = 1/\rho$ egyenlőség miatt ekvivalens a görbületi középpontok görbéjével, az evolútával. Ha ezt valamilyen simító módszerrel egyenletesebbé tesszük, egy olyan célgörbét kapunk ($\mathbf{e}$), amely megfelel egy ideálisan simított B-spline evolútájának.

Következő lépésként meg kell keresnünk azt a görbét, amely az eredetitől a lehető legkevésbé különbözik, de a görbületi középpontjai az imént kiszámolt célgörbére esnek. Ez a feladat egyúttal meghatároz egy mértéket is: a görbe evolútájának a célgörbétől való távolsága jellemzi a simaságot, azaz

$$E = \int \|(\mathbf{c}(t) + \rho\mathbf{n}(t)) - \mathbf{e}(t)\|^2 \mathrm{d}t. \tag{3.1}$$

Fontos megjegyezni, hogy ez a mérték feltételezi, hogy a görbe és a célgörbe azonosan paraméterezett. Magának a feladatnak a megoldására számos alternatíva létezik, ezeket részletesen a 3.4–3.5. pontokban tárgyaljuk, itt csak a két legfontosabb követelményt említjük meg. Az egyik a lokalitás: ez legegyszerűbben úgy érhető el, hogy egyszerre csak egy kontrollpontot mozgatunk; a másik pedig a változás korlátozása, amit itt is a kontrollpontok elmozdulására vonatkozó tolerancia bevezetésével valósítunk meg.

A gyakorlatban ajánlatos néhány egyszerűsítést bevezetni. A célgörbétől nem várjuk el, hogy nagyon pontosan illeszkedjen az eredeti evolútához – sokkal fontosabb, hogy sima legyen. Érdemes ezért nem túl sűrű mintavételezésű diszkrét adatokkal számolni. Minél lazább a simítás toleranciája, annál simább célgörbét kapunk, de ha túl nagy játékteret engedünk, fontos in-



3.2. ábra. B-spline önmetsző görbületi fésűvel.

formációkat is elveszíthetünk. Előnyös ezenkívül a görbületi középpontokról áttérni a velük gyakorlatilag ekvivalens görbületre, mivel az evolúta (és a görbületi fésű) lehet önmetsző (3.2. ábra), és ez megnehezíti a számolásokat. Nevezzük az így meghatározott görbét *célgörbületnek* és jelöljük g-vel. Ekkor a (3.1) mérték a következőképp módosul:

$$\hat{E} = \sum_i |\kappa(t_i) - g(t_i)|^2. \tag{3.2}$$

3.2. Kiterjesztés felületekre

A B-spline-okat simító eljárás kiterjesztésének egyik lehetséges módja, hogy a NURBS felület u és v paraméterirányú görbéit az előző pontban bemutatott módon kisimítjuk és a kapott fair görbék kontrollpontjai alapján egy új NURBS felületet készítünk.

Természetesebbnek tűnik azonban magát az algoritmust ültetni át felületekre. A görbület jellemzésére a κ_1, κ_2 felületi főgörbületeket és ezek kombinációit használjuk. Jelölje g_1 és g_2 a főgörbületek alapján meghatározott célgörbületeket, ekkor

$$\hat{\Pi} = \sum_i \sum_j (|\kappa_1(u_i, v_j) - g_1(u_i, v_j)|^2 + |\kappa_2(u_i, v_j) - g_2(u_i, v_j)|^2) \tag{3.3}$$

egy lehetséges mérték.

3.3. A célgörbület meghatározása

Diszkrét pontokkal megadott görbe simítására nagyon sok módszer található, ezek közül itt csak kettőt vizsgálunk meg. Az első talán a legegyszerűbb algoritmus, az átlagolás. Egy iterációban az eredeti görbe/felület minden pontját a szomszédai átlagával helyettesítjük és ezt ismételjük addig, amíg az eredmény megfelelő nem lesz (3.3. ábra). Már közepesen sűrű mintavételezésnél is elég lassú a simulás – ezt felgyorsíthatjuk azzal, ha nem a közvetlen, hanem a k lépésre levő szomszédok átlagát vesszük. A módszer hátránya, hogy a nemkívánatos egyenetlenségekkel együtt a görbület értékes részei is kisimulnak.

Másik lehetőség a megadott pontok approximációja egy azonos csomóvektorú B-spline-nal ($\mathbf{g}$) ill. NURBS-zel ($\mathbf{g}_1, \mathbf{g}_2$). Ez jóval bonyolultabb feladat, de ennek megfelelően szebb eredményt is ad (3.4. ábra), és még csak nem is költségesebb. Ehhez egy olyan lineáris egyenletrendszert kell megoldani, amely az

$$F(\mathbf{g}) = \sum_i \|\mathbf{g}(t_i) - \kappa^0(t_i)\|^2,$$

ill. felületeknél $\mathbf{g}_q$ ($q = 1, 2$) illesztése esetén az

$$F(\mathbf{g}_q) = \sum_i \sum_j \|\mathbf{g}_q(u_i, v_j) - \kappa_q^0(u_i, v_j)\|^2$$

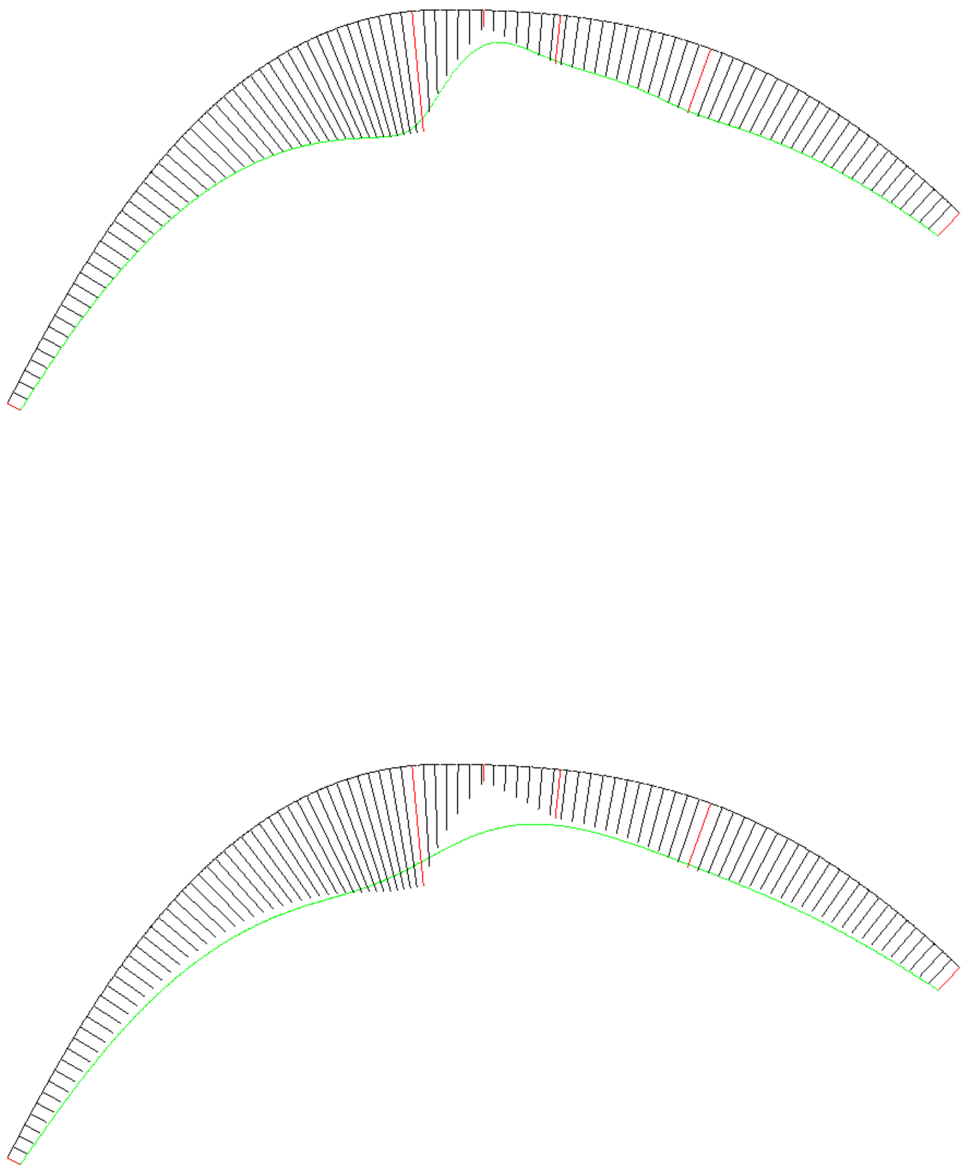
funkcionált minimalizálja, ahol κ^0 és κ_q^0 az eredeti görbe/felület görbületei[1]. A célgörbület simítása érdekében érdemes még hozzátenni egy, a függvény görbületének simítására vonatkozó tagot. Az első funkcionál ekkor így alakul:

$$\hat{F}(\mathbf{g}) = \sum_i \|\mathbf{g}(t_i) - \kappa^0(t_i)\|^2 + \int_t \hat{\kappa}(t)\mathrm{d}t,$$

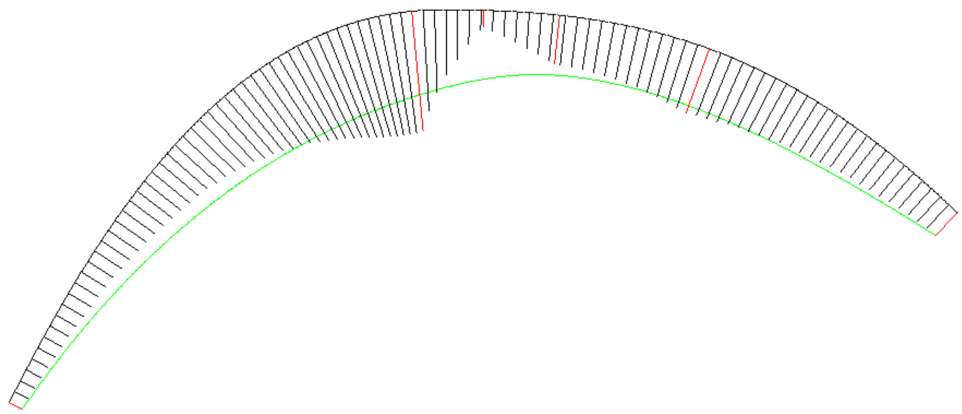
ahol $\hat{\kappa}$ a $\mathbf{g}$ görbülete. Görbéknél érdemes lerögzíteni a végpontokat is, de felületeknél a szélek rögzítése elég körülményes és az algoritmus működését lényegesen nem befolyásolja. Az illesztési probléma megoldására [14] ad algoritmusokat, de ezek kifejtése túlmutat e diplomamunka keretein.

Az approximáció toleranciája egy fontos paraméter, melynek beállítása egyelőre nem automatizált. Bármennyire jól határozzuk is meg ezt a paramétert, bizonyos esetekben elkerülhetetlen, hogy a görbületben „felesleges" inflexiók maradnak (3.5. ábra). Ilyenkor segít, ha néhány fairing lépés után újraillesztjük a görbületet. A célgörbület függvény elfogadását a felhasználóra bízzuk; ugyanakkor eszközöket biztosítunk különböző függvények előállítására.

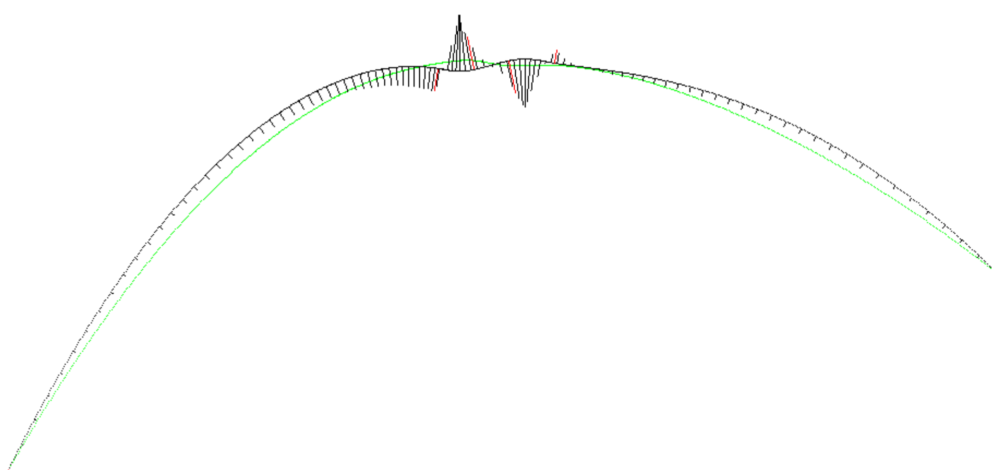
[1] Itt a görbületeket a paramétertéren értelmezett görbeként ill. felületként értelmezzük.



3.3. ábra. B-spline átlagolt görbülete 10 ill. 100 iteráció után.



3.4. ábra. B-spline a görbületi fésűre illesztett B-spline-nal.



3.5. ábra. A görbület inflexiói az illesztés után is megmaradhatnak.

3.4. A legjobb közelítés keresése

Az $\hat{E}$ és $\hat{\Pi}$ mértékek minimumát különböző módokon találhatjuk meg, a legegyszerűbb egy iteratív, a kontrollpontok perturbációján alapuló módszer, amely nagy vonalakban a következő lépésekből áll:

`ciklus`

1. egy kontrollpont kiválasztása
2. a kontrollpont elmozgatása
3. a mérték újraszámolása
4. `ha` a mérték nőtt (romlott), `akkor` a kontrollpont visszaállítása

`amíg` *feltétel*

A kontrollpont kiválasztása történhet a mérték alapján, vagyis azt a kontrollpontot változtatjuk, amelyikhez közel veszi fel a legnagyobb értékét, valamint választhatunk véletlenszerűen is. Az első esetben valahogyan ki kell védeni a program „megakadását”, lehet pl. listát készíteni az utolsó néhány kiválasztott pontról és ezeket letiltani stb., de a tapasztalat azt mutatja, hogy ha elég sok iterációt futtatunk, akkor a véletlenszerű választás is jó eredményt ad.

A mozgatás iránya szintén lehet véletlenszerű, pl. a 6 tengelyirányban (26, ha összetett irányokat is megengedünk), de egy irány kiválasztása helyett érdemes mindet végignézni, és ha valamelyiknél volt javulás, akkor a legjobbat kiválasztani.

Fontosabb kérdés, hogy milyen távolságra mozgassuk a kontrollpontot. Támpontot adhat a felület határolódobozának testátlója, ennek az egy százaléka vagy néhány ezreléke reális érték lehet. Jobb becslést kaphatunk, ha a kontrollpontok szomszédjaiktól való átlagos távolságát vesszük alapul; ennek 10-50%-a valószínűleg használható érték. Függővé lehet tenni a távolságot az adott kontrollponthoz tartozó görbepont/felületi pont simasági mértékétől is: logikusnak tűnik, hogy az elmozgatás mértéke legyen arányos a mértékkel vagy pl. annak logaritmusával, hiszen ez a problémás helyeken nagyobb változást idéz elő, míg a nagyjából jó részeket nem engedi elrontani.

Természetesen ez az algoritmus nem biztosítja, hogy megtaláljuk a globális minimumot (valójában ilyen algoritmus nem is igazán létezik [14]), sőt, még azt sem, hogy a megoldás, amit találtunk, az azonos simaságúak közül a lehető legközelebb van az eredeti görbéhez ill. felülethez. Ennek egyik oka az, hogy az egymás után következő lépések folyamatosan növelhetik a távolságot. Javíthat egy kicsit, ha egy–egy iterációban egyszerre több pontot mozgatunk, egymástól függetlenül.

Megtilthatjuk ezenkívül a szélső kontrollpontok elmozdítását, valós helyzetekben ez mind görbék, mind felületek esetén gyakran kívánatos. A leállási feltételre itt is ugyanazok érvényesek, mint a Hahmann-algoritmus esetén. Nincsen tökéletes választás, alkalmazásonként érdemes többfélét kipróbálni.

3.5. Hatékonysági megfontolások

Ebben a fejezetben célunk az eljárás helyességének bemutatása volt, azonban nem árt néhány szót szólni a számítások hatékonyságáról. Ideális esetben a fairing eljárásnak olyan gyorsnak kellene lennie, hogy 1–2 perc alatt elfogadható eredményt tudjon adni (az autóipari gyakorlatban egy felületmodell simításával gyakran több napot töltenek el a szakemberek).

Az algoritmusban egyértelműen a görbe ill. felület pontjainak és görbületének kiszámítása a szűk keresztmetszet. Ez sokszorosára gyorsítható a B-spline bázisfüggvények és deriváltjaik előzetes kiszámításával és eltárolásával. Mivel a program végig egy adott paraméterezéssel dolgozik, ezeket az értékeket felesleges újra és újra kiszámítani.

Szintén sokat lendíthet a hatékonyságon egy jobb közelítő módszer használata. Több dimenziós minimalizálási feladatra különböző módszereket ismerünk, pl. a Powell-eljárást vagy a szimplex módszert [14]. Numerikus algoritmusok helyett genetikus algoritmusokkal is lehet próbálkozni – ezeknek a módszereknek a megvalósítása, összehasonlítása a további kutatómunka egy fejezetét képezik.

4. fejezet

Tesztkörnyezet és futtatási eredmények

A szakirodalomban található mértékek és módszerek, valamint az új ötletek kipróbálására létrehozott szoftverkörnyezet görbék és felületek vizsgálatára egyaránt alkalmas, segítségével az előző fejezetekben bemutatott algoritmusok könnyen megvalósíthatóak, tesztelhetőek.

4.1. A tesztkörnyezet felépítése

A program *Microsoft Windows* operációs rendszerben készült, *Scheme* és *C++* nyelven. A fejlesztéshez a *Microsoft Visual Studio 6.0* környezetet, az *Intel C++ Compiler 8.0* fordítóprogramot, az *STLport* sablonkönyvtárat és az ingyenes *SCM 5d9* interpretert használtam.

Ezeken kívül a *Raindrop Geomagic Hungary* cég az alábbi két könyvtárat bocsátotta rendelkezésemre:

- Egy matematikai könyvtárat, amely tartalmazza a két- és háromdimenziós görbékkel és felületekkel kapcsolatos legfontosabb osztályokat (pl. pont, vektor, mátrix, bázisfüggvény, B-spline, NURBS stb.) és függvényeket (pl. affin transzformációk, görbe/felület görbülete, approximációja megkötésekkel stb.).

- Egy Scheme könyvtárat, amely a kezelőfelülethez szükséges elemek (pl. ablakok, gombok stb.) létrehozását teszi lehetővé és a háromdimenziós színtér megjelenítéséhez *OpenGL* alapú utasításokat biztosít.

A Scheme könyvtárban kifejlesztettem egy–egy új osztályt görbék ill. felületek megjelenítésére és változtatására (`bsc`, `bss`), és implementáltam a

make-bsc			make-bss		
bsc:get-knot	<bsc>	<index>	bss:get-u-knot	<bss>	<index>
bsc:load-file	<bsc>	<file>	bss:load-file	<bss>	<file>
bsc:regenerate	<bsc>		bss:regenerate	<bss>	
bsc:resolution	<bsc>	[res]	bss:resolution	<bss>	[res]
bsc:fit-points	<bsc>	<points>	bss:display-map	<bss>	[map]
...			...		

4.1. táblázat. Scheme utasítások B-spline görbék és felületek kezelésére

legfontosabb grafikus vizsgálati eljárásokat és néhány simító algoritmust. A 4.1. táblázat felsorol néhányat az ezeknek megfelelő Scheme parancsok közül.

4.2. A program használata

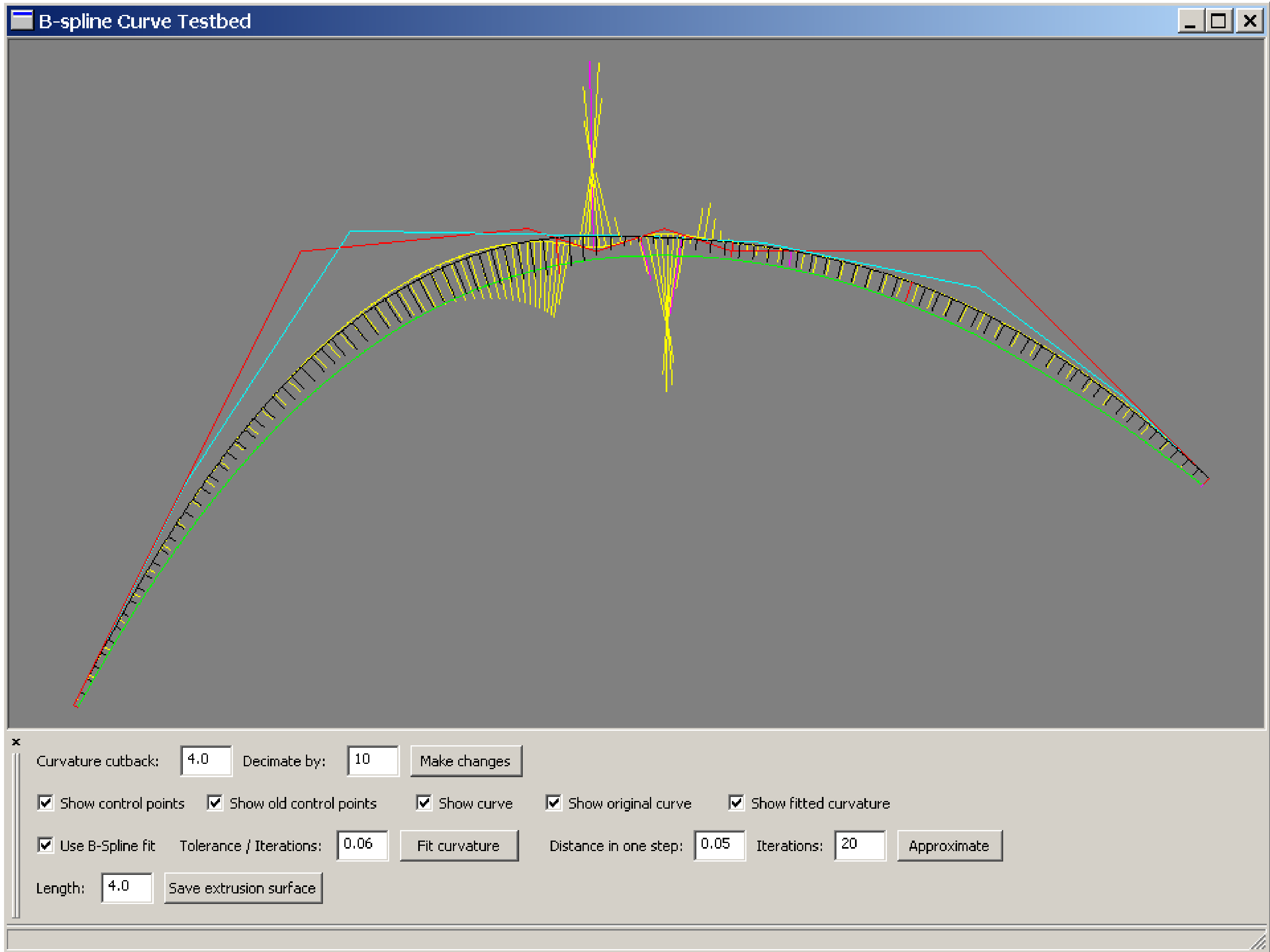
A tesztkörnyezet két különálló programból áll: a *B-Spline Curve Testbed (BSCTest)* görbéket, míg a *B-Spline Surface Testbed (BSSTest)* felületeket kezel. A programokat az SCM interpreterrel lehet futtatni; az interpreteres nyelvek – így a Scheme is – átlátható, egyszerű szintaktikájuk és magasszintű utasításaik miatt különösen alkalmasak prototípusok készítésére. A sebesség elvben problémát okozhat, de mivel itt az összes számítást a már lefordított könyvtár hajtja végre, a simítási lépések aránylag gyorsan lefutnak és a fényvonalak valós időben, akadozás nélkül nézhetőek.

4.2.1. B-Spline Curve Testbed

A program képes megjeleníteni B-spline görbéket és ezek legfontosabb jellemzőit. A kirajzolást számos paraméterrel hangolhatjuk a feladathoz, pl. a túllógó görbületi fésűt visszavághatjuk, a kontrollpoligont eltüntethetjük, vagy éppen megváltoztathatjuk a felbontást.

A 3. fejezetben bemutatott módszer különböző variánsai (pl. átlagolásos és illesztéses módszer) egyaránt tesztelhetőek, a paraméterek a kezelőfelületen (4.1. ábra) könnyedén beállíthatóak.

Az egyenetlenségek megmutatása céljából a görbéket egy egyenes szakaszon mozgatva transzlációs felületet (*extruded surface*) is tud generálni, ezeket a BSSTest kezeli.



4.1. ábra. A B-Spline Curve Testbed kezelőfelülete.

4.2.2. B-Spline Surface Testbed

Ez a program négy feladatot lát el: a NURBS felület megjelenítését, változtatását, simasági mértékeinek kiszámítását és simítását. Ezek a funkciók a kezelőfelületen (4.2. ábra) pár kattintással elérhetőek.

Megjelenítés

A megjelenítésnek számtalan módja van, ezek legtöbbjéhez paraméterek is tartoznak. Alapértelmezésben a felület árnyalva jelenik meg, a kontrollhálóval együtt (4.3. ábra). Gyakran hasznos eszköz a szelettérkép (*slice map*), amely a felületet párhuzamos síkokkal metszi el (4.4. ábra). A síkok normálvektorát és sűrűségét a felhasználó állíthatja be. Noha a csomóvektorok arányait mindig látni a képernyőn, fontos lehet tudni, hogy a csomóvonalak pontosan hol helyezkednek el (4.5. ábra). Az egyébként simának tűnő felület hibáit jól kihozzák a görbülettérképek (4.6–4.7. ábra) és a fénysáv (4.8. ábra). Végül pedig a leglátványosabb vizsgálati eszköz a fényvonal (4.9. ábra), a teszteredményeknél is ezt fogjuk használni.

Változtatás

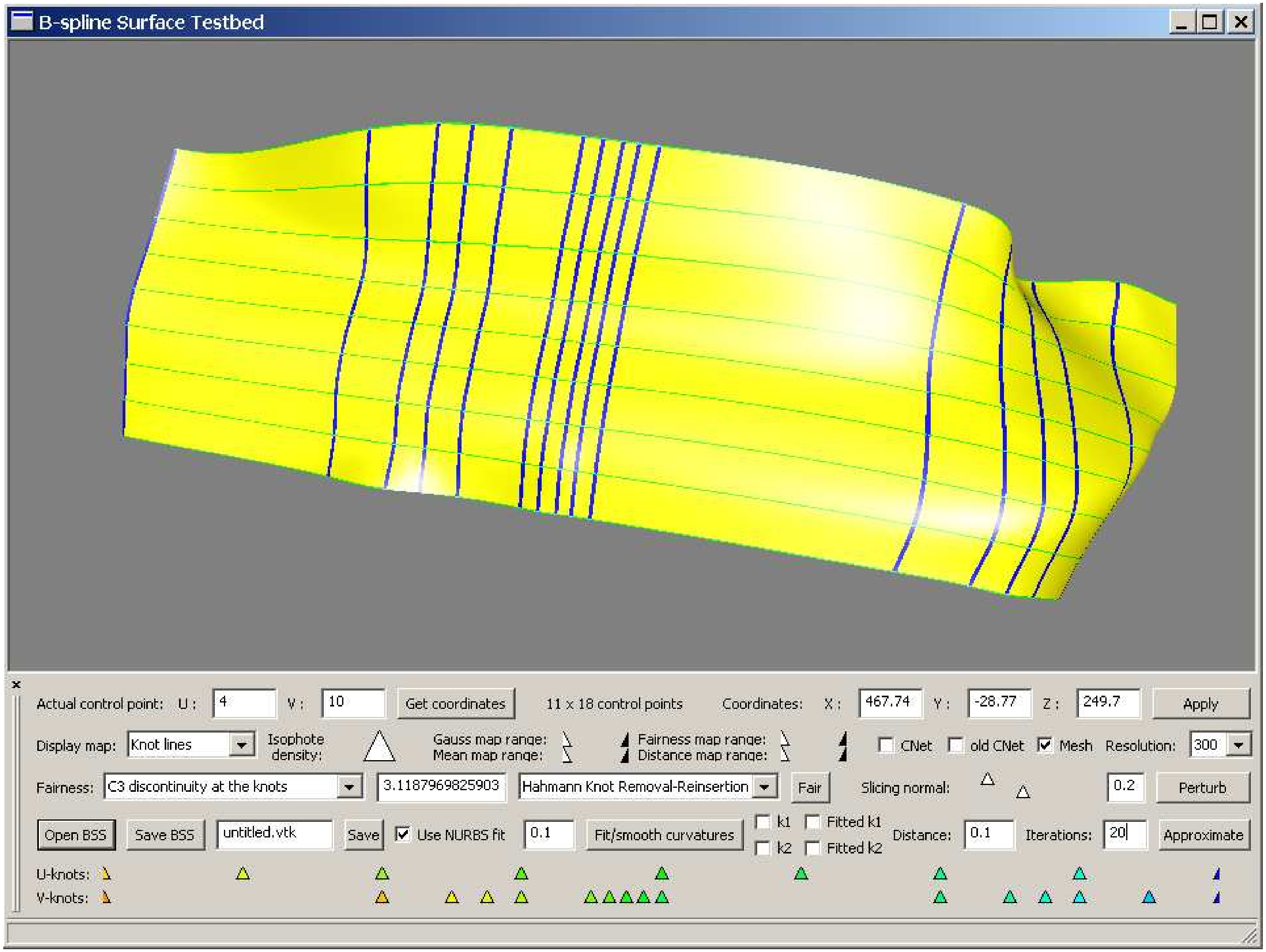
A felület változtatása három módon lehetséges: a kontrollpontok mozgatásával, a csomópontok mozgatásával és perturbációval. Ez utóbbi a kontrollpontokat egy adott tolerancián belül véletlenszerűen elmozdítja, így egy fair modellből az algoritmusok tesztelésére alkalmas felületet csinál.

Mértékek

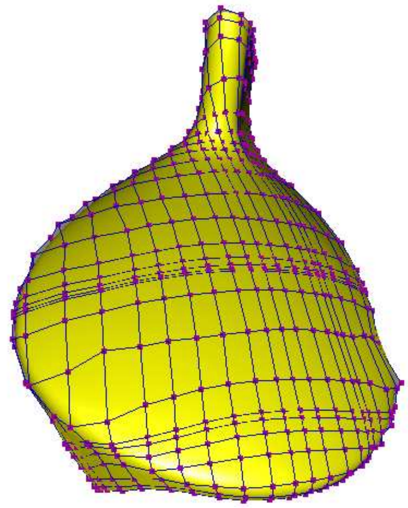
A 2. fejezetben szereplő mértékek többsége, és természetesen a 3. fejezetben bevezetett mérték is bármikor lekérdezhető.

Simítás

A Hahmann-algoritmus különféle változatai és a saját fejlesztésű algoritmusvariánsok mind használhatóak, a fontosabb paramétereket a kezelőfelületen is lehet módosítani. Lehetőség van a főgörbületek és a célgörbületek megjelenítésére és a simított felületek elmentésére.

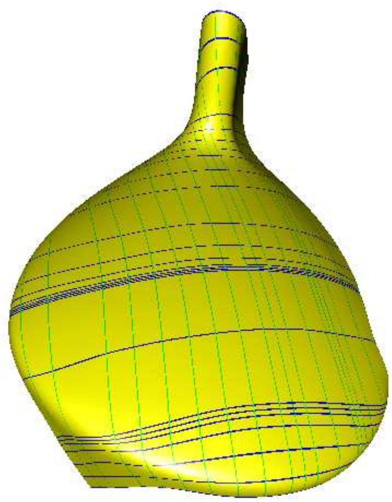


4.2. ábra. A B-Spline Surface Testbed kezelőfelülete.

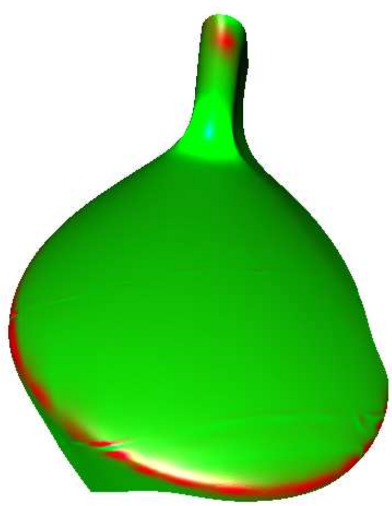


4.3. ábra. A „golfütő” árnyékolt megjelenítése, kontrollhálóval.

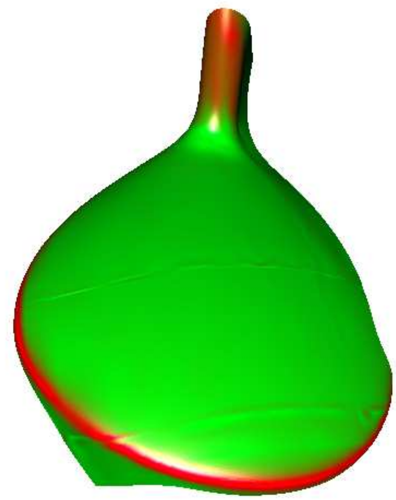
4.4. ábra. A „golfütő” szelettérképe.



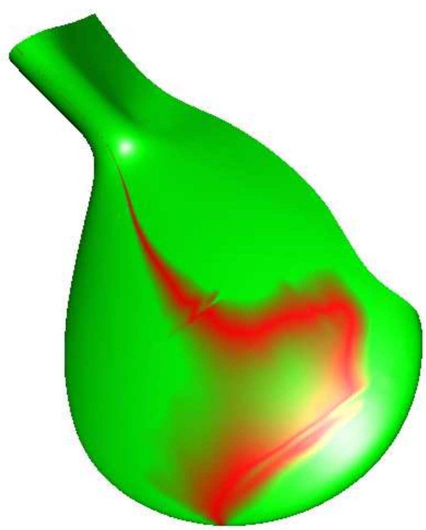
4.5. ábra. A „golfütő” csomóvonalai.



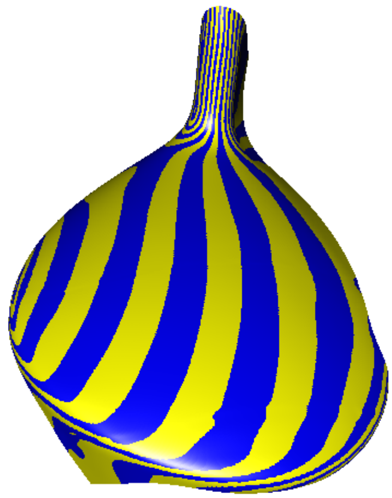
4.6. ábra. A „golfütő” Gauss-görbülettérképe.



4.7. ábra. A „golfütő” átlag-görbülettérképe.



4.8. ábra. Fénysáv a „golfütő” felületen.



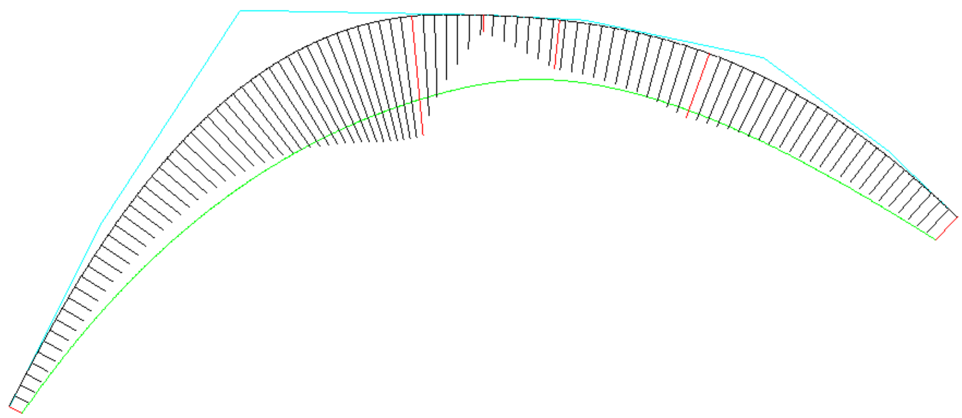
4.9. ábra. Fényvonalak a „golfütő” felületen.

4.3. Teszteredmények

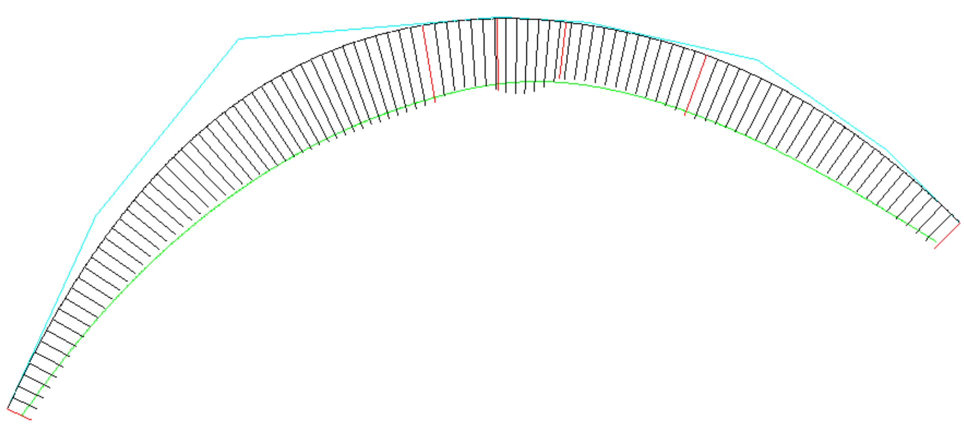
A következő képek négy fairing folyamatot illusztrálnak. Az első kettő görbéket, a másik kettő felületeket simít.

A 4.10–4.13. képsorozaton jól látszik, hogy a görbület milyen sokat javul. Magán a görbén ez a változás alig érzékelhető, de a 4.14–4.15. ábrák már jelentős különbséget mutatnak. Szintén szembeötlő a javulás a 4.16–4.21. szekvencián: a 4.22. és 4.23. ábrákat összevetve látszik, hogy egy viszonylag durva görbéből még átlagolt célgörbülettel is igényes eredményt kapunk.

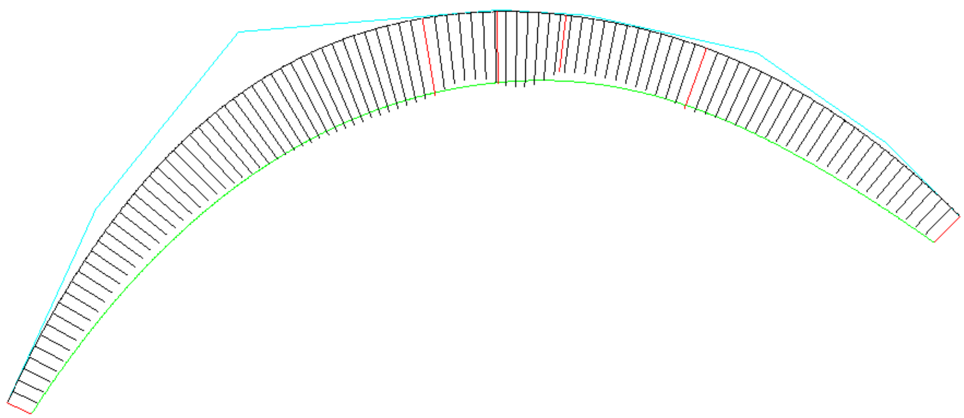
A felületekre vonatkozó teszteket egy mért adatok alapján rekonstruált Fiat karosszériaelem (4.24. ábra) két nagy funkcionális felületéből generáltam, ezek elnevezése *top* (sárga) és *bottom* (lila). A *top* felület simítására (4.25–4.31. ábra) vonatkozó adatokat a 4.2. táblázat foglalja össze. Hasonlóan, a *bottom* felület simításának (4.32–4.37. ábra) eredményei a 4.3. táblázatban olvashatóak.



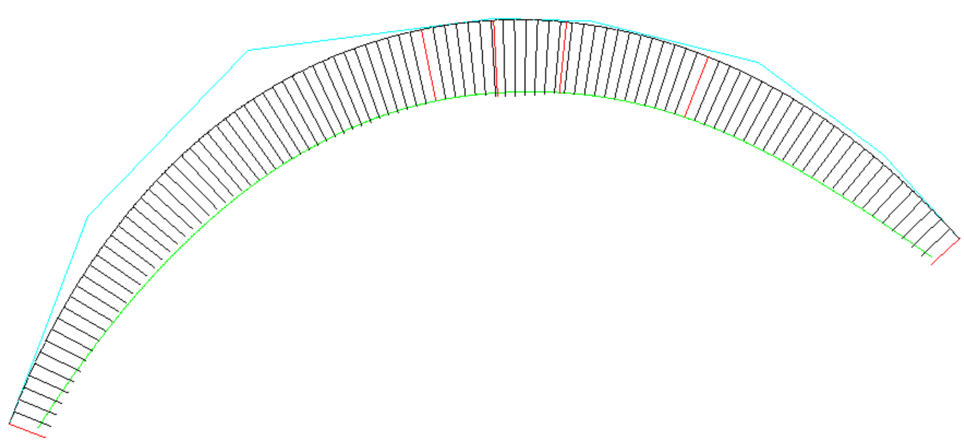
4.10. ábra. A kiinduló görbe az illesztett célgörbülettel.



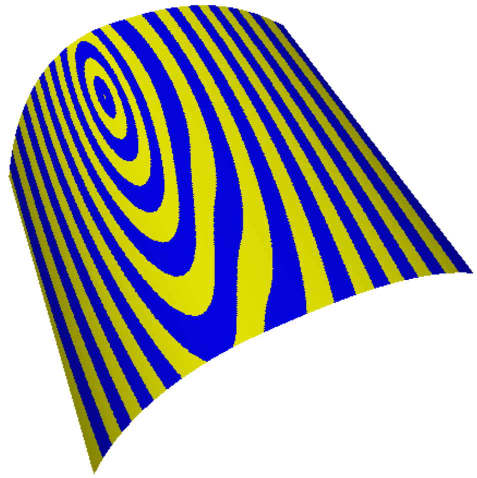
4.11. ábra. A görbe 20 iteráció után.



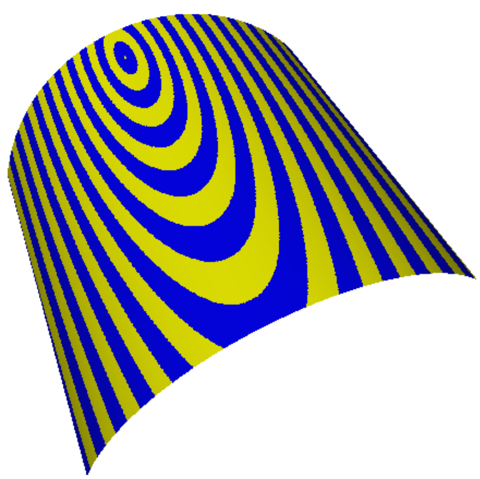
4.12. ábra. Az újraillesztett célgörbület.



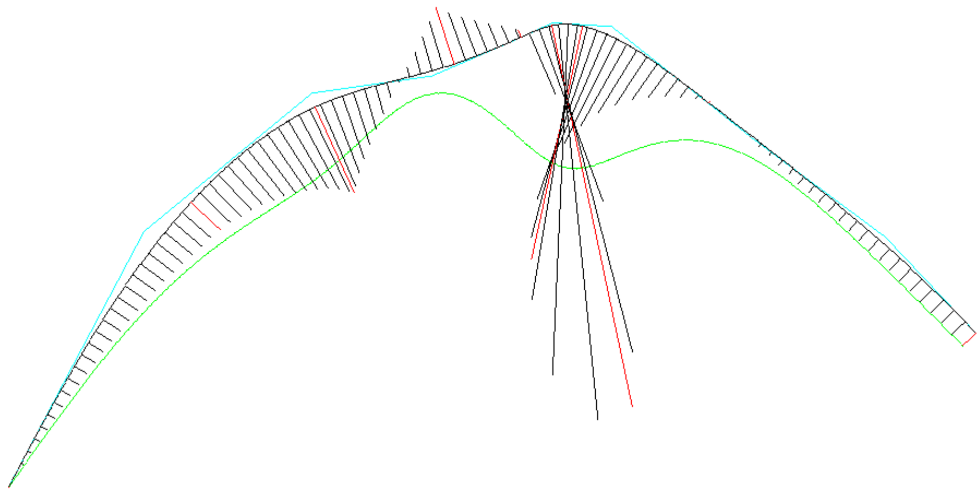
4.13. ábra. A görbe újabb 20 iteráció után.



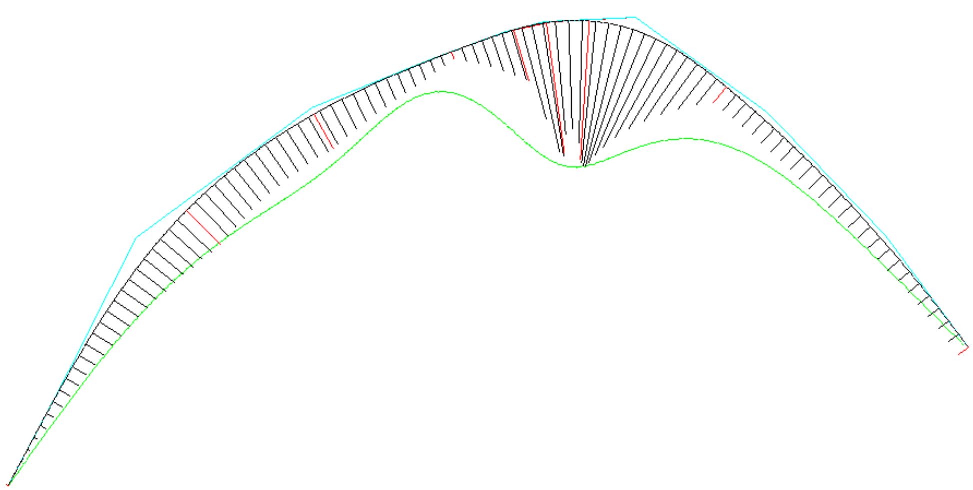
4.14. ábra. A kezdeti görbéből készített transzlációs felület.



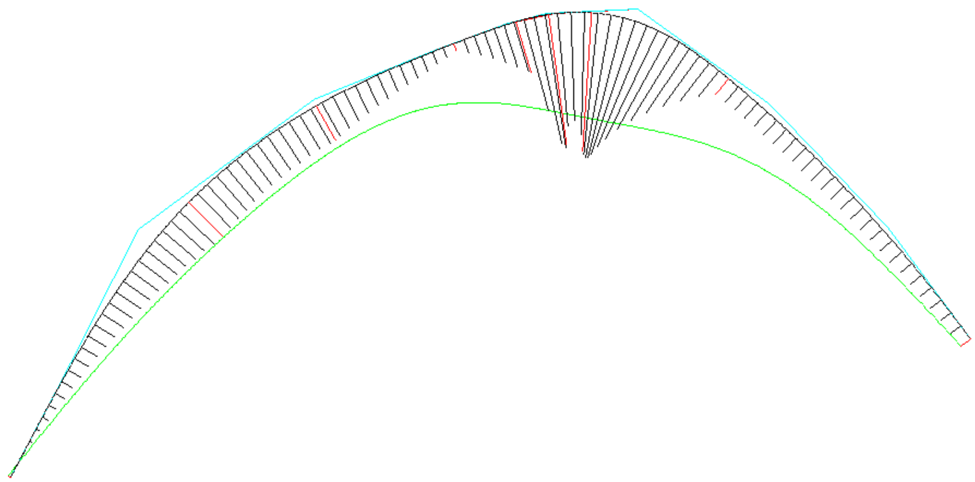
4.15. ábra. A fair görbéből készített transzlációs felület.



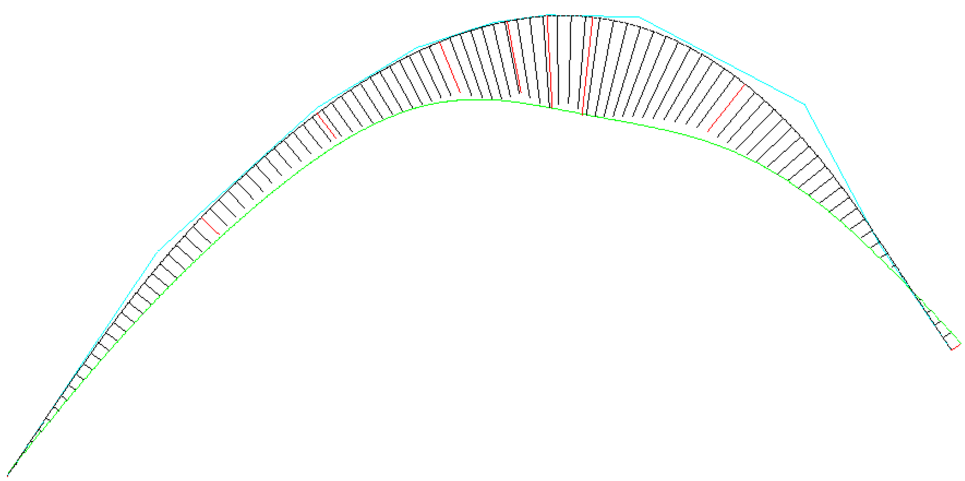
4.16. ábra. A kiinduló görbe az átlagolt célgörbülettel.



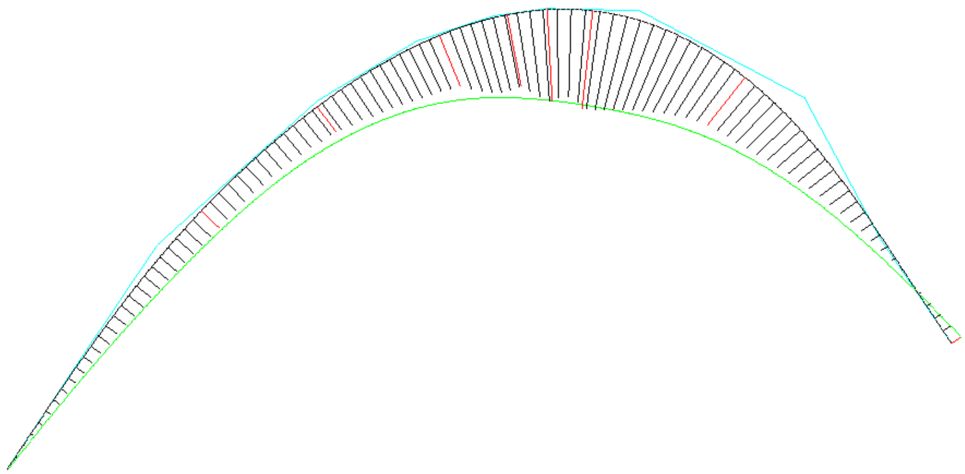
4.17. ábra. A görbe 20 iteráció után.



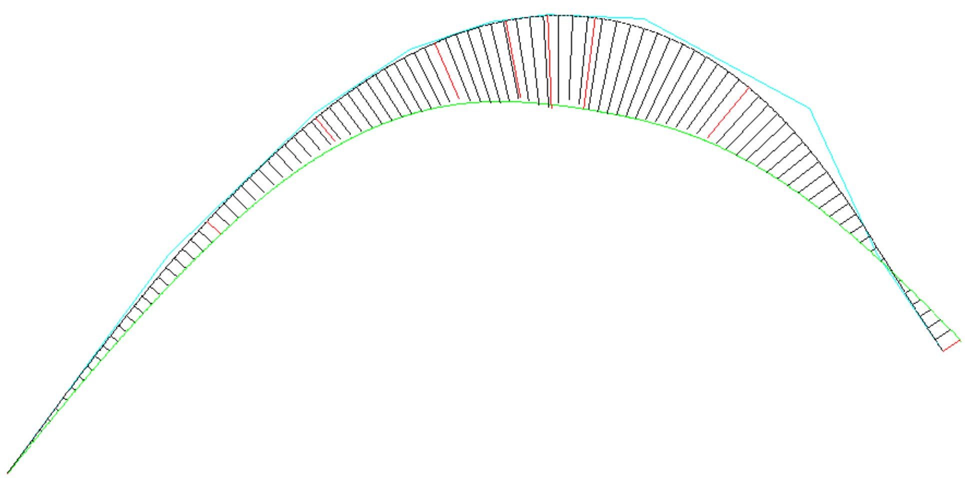
4.18. ábra. Az újraátlagolt célgörbület.



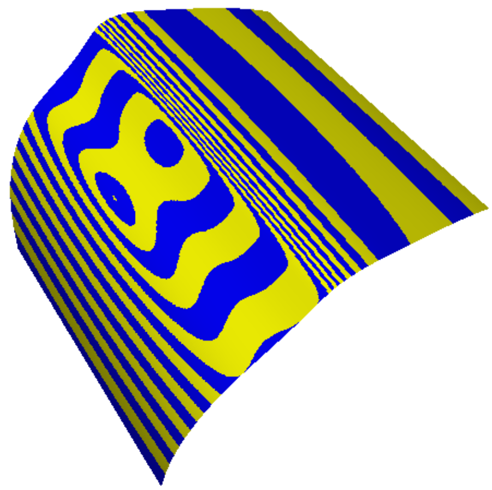
4.19. ábra. A görbe újabb 20 iteráció után.



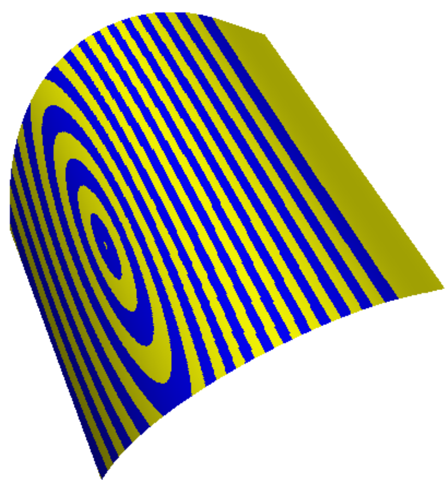
4.20. ábra. Az ismét kiszámolt célgörbület.



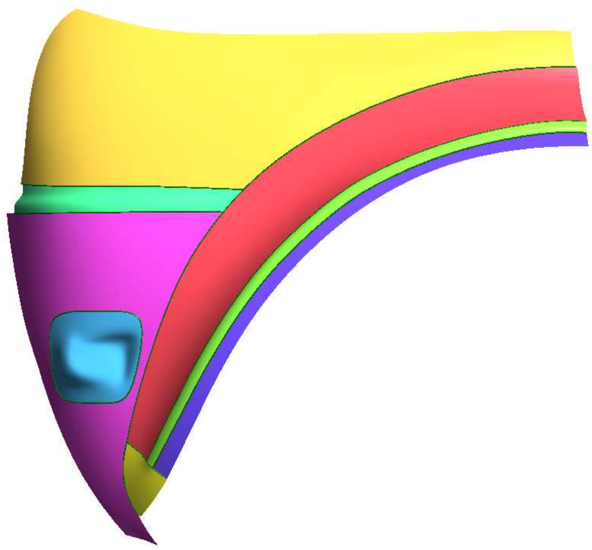
4.21. ábra. A görbe újabb 20 iteráció után.



4.22. ábra. A kezdeti görbéből készített transzlációs felület.



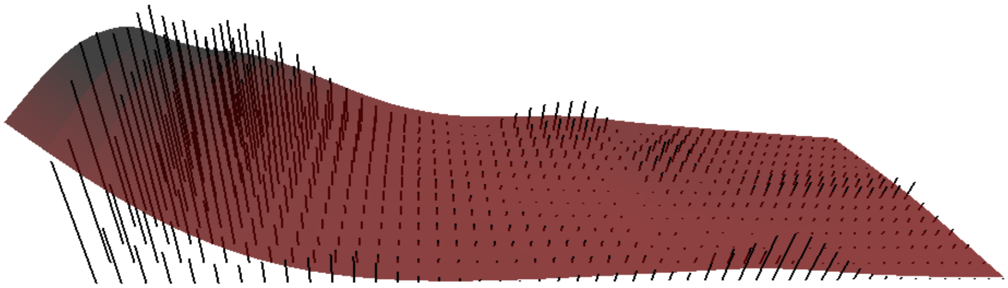
4.23. ábra. A fair görbéből készített transzlációs felület.



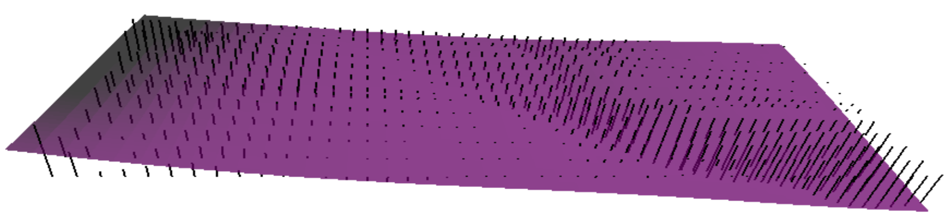
4.24. ábra. Valós mért adatok alapján NURBS felületekkel közelített Fiat karosszériaelem.

Iteráció száma	**Simasági mérték**	**Max. távolság (mm)**
0	0.901531	0.00000
200	0.244919	2.91302
400	0.223582	4.40624
600	0.215052	5.02049
800	0.201543	5.14342
1000	0.198280	5.85641

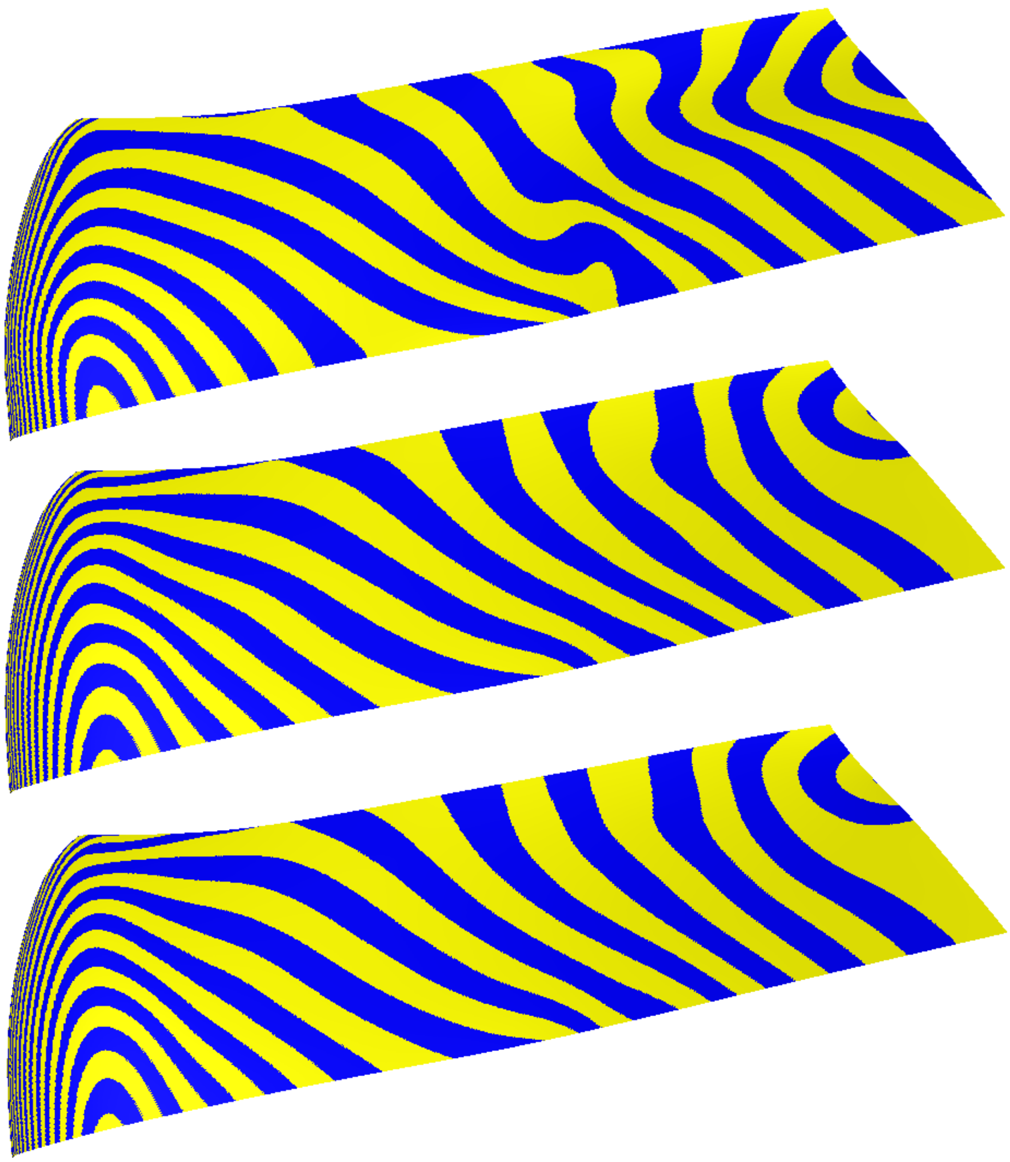
4.2. táblázat. A *top* felület simításának fázisai.



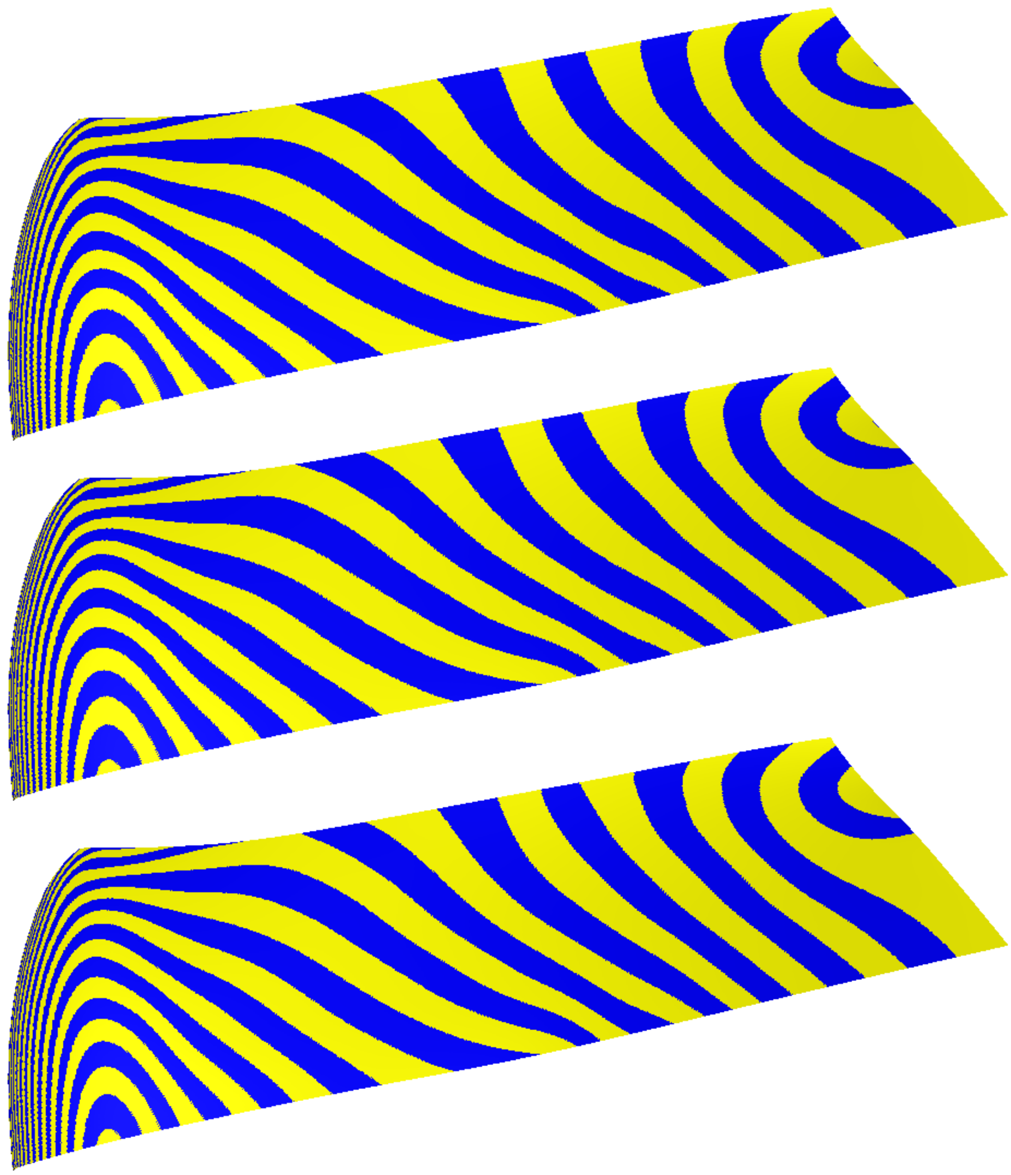
4.25. ábra. A *top* felület κ_1 főgörbülete és az erre illesztett célfelület.



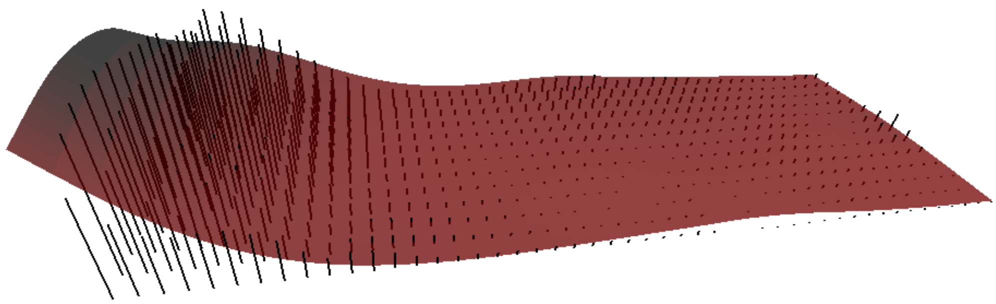
4.26. ábra. A *top* felület κ_2 főgörbülete és az erre illesztett célfelület.



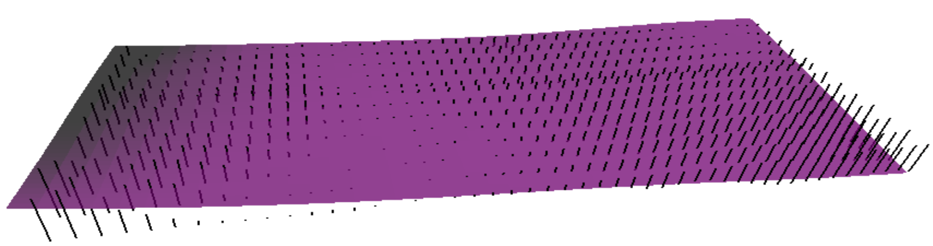
4.27. ábra. A *top* felület fényvonaltérképe kezdőállapotban, 200 és 400 iteráció után.



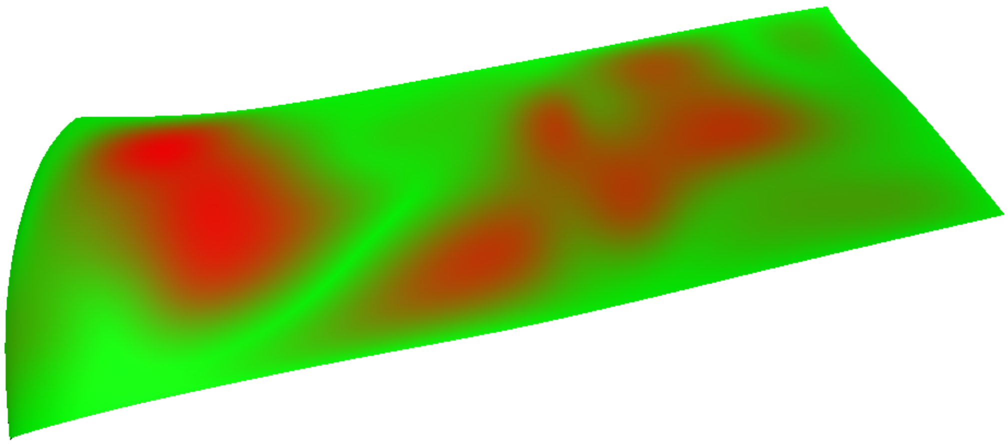
4.28. ábra. A *top* felület fényvonaltérképe 600, 800 és 1000 iteráció után.



4.29. ábra. A *top* felület κ_1 főgörbülete 1000 iteráció után és a célfelület.



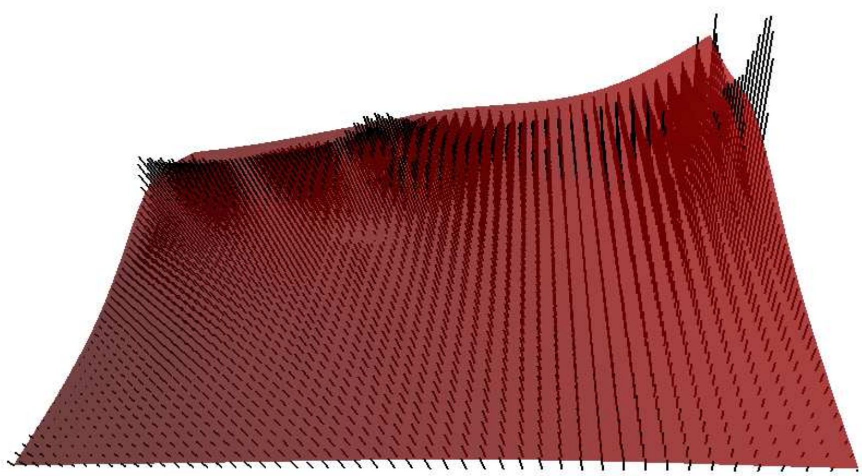
4.30. ábra. A *top* felület κ_2 főgörbülete 1000 iteráció után és a célfelület.



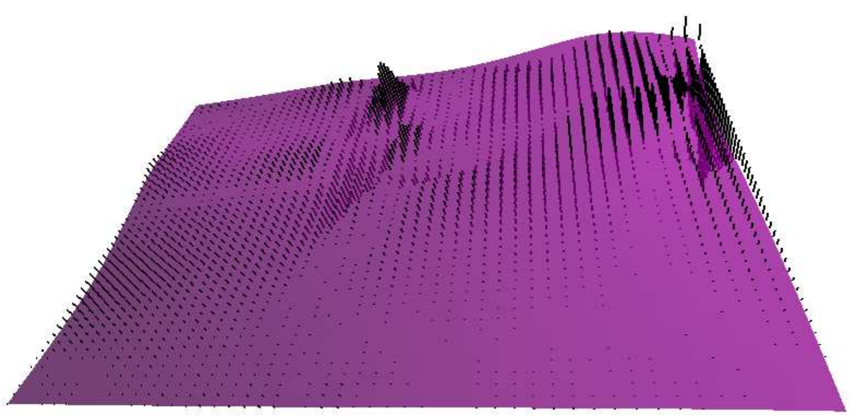
4.31. ábra. A simított *top* felület távolsága az eredetitől.

Iteráció száma	**Simasági mérték**	**Max. távolság (mm)**
0	1.367201	0.00000
200	0.628472	3.57769
400	0.534705	5.28126
600	0.503814	5.94847
800	0.476197	6.84875
1000	0.462829	6.96793

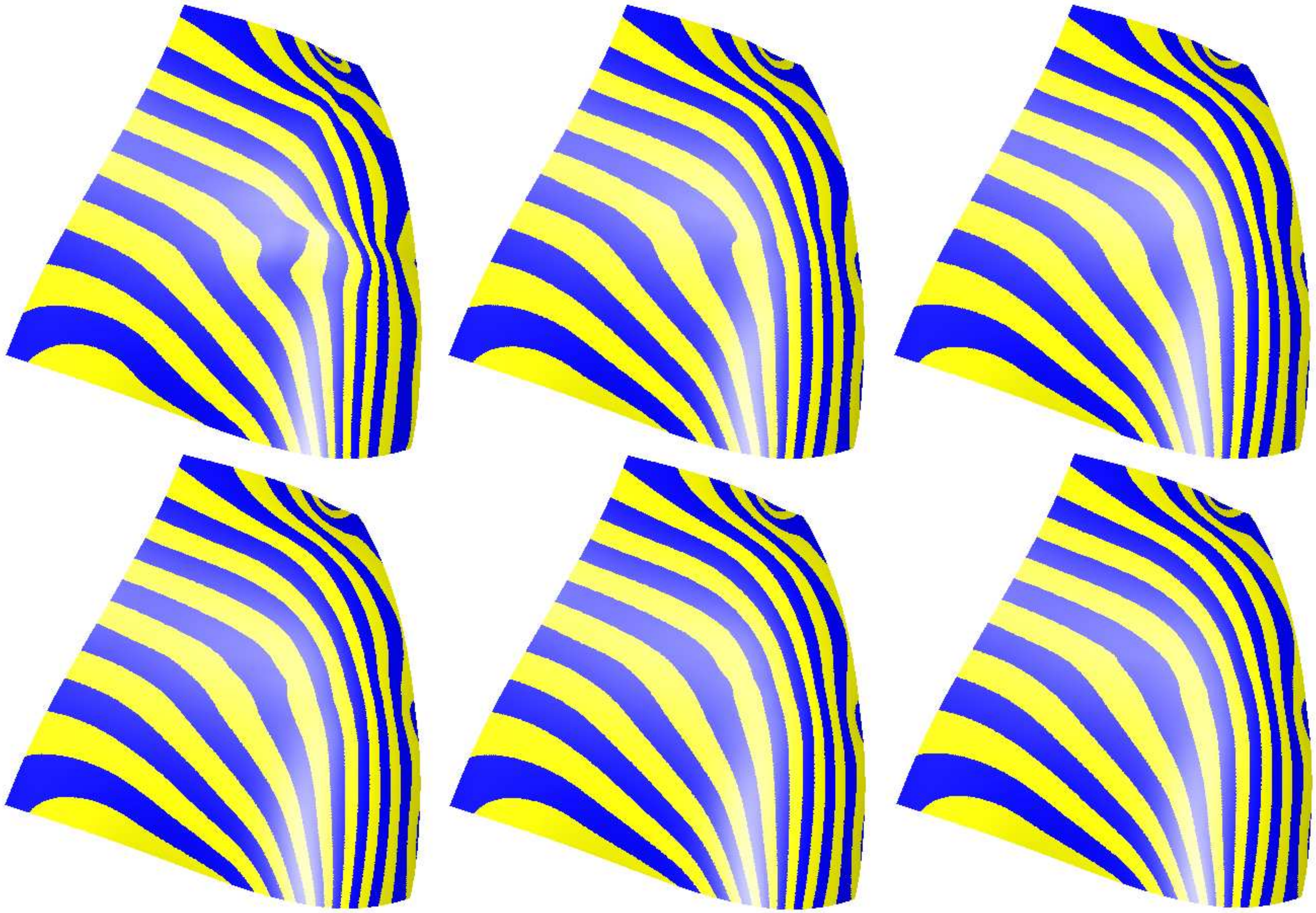
4.3. táblázat. A *bottom* felület simításának fázisai.



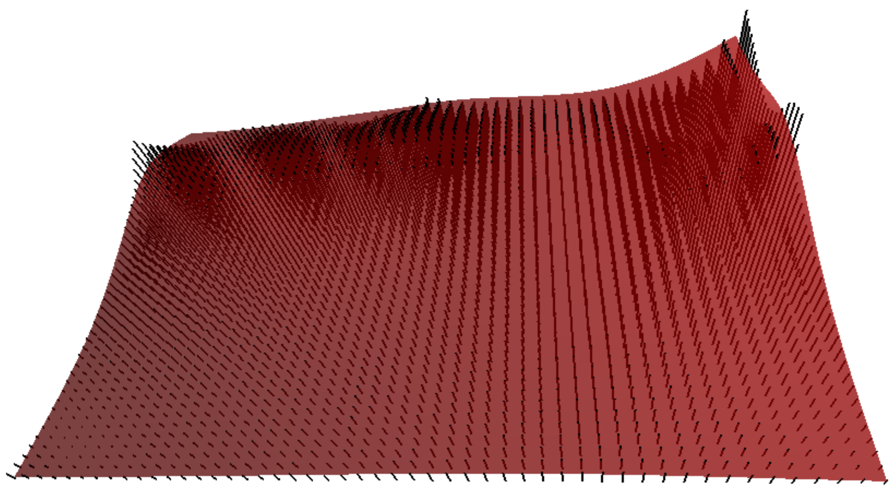
4.32. ábra. A *bottom* felület κ_1 főgörbülete és az erre illesztett célfelület.



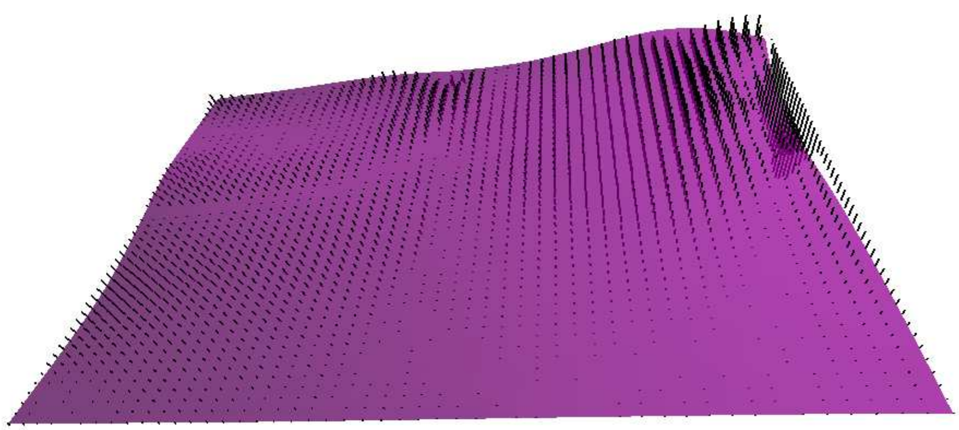
4.33. ábra. A *bottom* felület κ_2 főgörbülete és az erre illesztett célfelület.



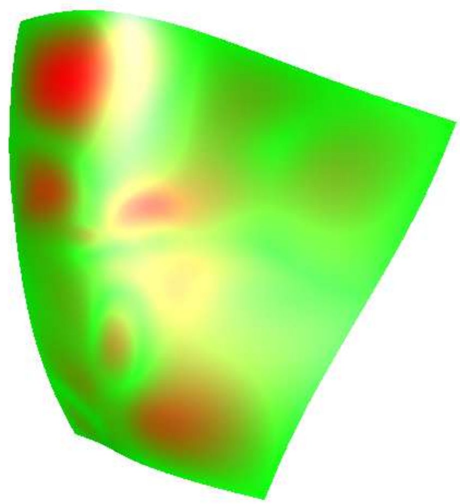
4.34. ábra. A *bottom* felület fényvonaltérképe kezdőállapotban, 200, 400, 600, 800 és 1000 iteráció után.



4.35. ábra. A *bottom* felület κ_1 főgörbülete 1000 iteráció után és a célfelület.



4.36. ábra. A *bottom* felület κ_2 főgörbülete 1000 iteráció után és a célfelület.



4.37. ábra. A simított *bottom* felület távolsága az eredetitől.

5. fejezet

Összefoglalás

Jelen diplomamunka keretén belül a következő feladatok megoldására vállalkoztam:

1. Megismertem és bemutattam a mérnöki visszafejtés fázisait, eszközeit és alapvető kérdéseit. Érzékeltettem a fair felületek, s így a felületsimítás feladatának fontosságát, problémakörének bonyolultságát.

2. Elolvastam a szakirodalom e témával kapcsolatos legfontosabb cikkeit és összefoglaltam az ezekben leginkább lényegesnek tekintett kutatási eredményeket. Rámutattam a fairing módszerek gyengén megoldott részfeladataira és a mértékekkel szemben támasztott alapvető követelményekre.

3. A görbe- és felületsimító algoritmusok közül kiválasztottam kettőt. Megmutattam ezek hiányosságait és készítettem belőlük egy új eljárást, amely megfelel a korábban felállított feltételeknek.

4. Készítettem két tesztprogramot, amelyek segítségével a görbékre ill. felületekre vonatkozó algoritmusok és variánsaik kipróbálhatóak, összehasonlíthatóak.

5. Kísérleteket végeztem a programok segítségével különféle tesztobjektumokon, s ezekkel igazoltam a javasolt módszer helyességét.

6. Nem foglalkoztam az eljárás robusztus, gyors numerikus módszerrel történő megvalósításával. Jelen munka célja elsősorban az új eljárás működőképességének demonstrálása volt; a számítási hatékonyság növelése elkülönült programozási feladatnak tekinthető.

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani mindazoknak, akik nélkül ez a diplomamunka nem készülhetett volna el – elsősorban témavezetőimnek, Várady Tamásnak és Vida Jánosnak, valamint a *Raindrop Geomagic Hungary* cégnek, amely biztosította a fejlesztéshez szükséges körülményeket és eszközöket. Köszönettel tartozom még a cég dolgozóinak, akik tanácsaikkal és észrevételeikkel sokat segítettek, s különösképp Volker Weißnak és Terék Zsoltnak, akikhez mindig bátran fordulhattam kérdéseimmel.

Irodalomjegyzék

[1] K-P. Beier, Y. Chen, *The Highlight Band, a Simplified Reflection Model for Interactive Smoothness Evaluation.* In: N. S. Sapidis (Ed.), Designing Fair Curves and Surfaces, pp. 213-230, SIAM, ISBN 0-898-71332-3, 1994.

[2] M. Eck, J. Hadenfeld, *Local Energy Fairing of B-Spline Curves.* Computing Supplement 10, pp. 129-147, 1995.

[3] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide.* Academic Press, 5th edition, 2002.

[4] G. Farin, J. Hoschek, M.-S. Kim (Eds.), *Handbook of Computer Aided Geometric Design.* North-Holland, 2002.

[5] G. Greiner, *Curvature approximation with application to surface modeling.* In: J. Hoschek, P. Kaklis (Eds.), Advanced Course on FAIRSHAPE, pp. 59-75, Teubner Stuttgart, ISBN 3-519-02634-1, 1996.

[6] J. Hadenfeld, *Fairing of B-Spline Curves and Surfaces.* In: J. Hoschek, P. Kaklis (Eds.), Advanced Course on FAIRSHAPE, pp. 59-75, Teubner Stuttgart, ISBN 3-519-02634-1, 1996.

[7] J. Hadenfeld, *Local Energy Fairing of B-Spline Surfaces.* In: M. Daehlen, T. Lyche, L. L. Schumaker (Eds.), Mathematical Methods for Curves and Surfaces, pp. 203-212, Vanderbilt University Press, ISBN 0-826-51268-2, 1995.

[8] S. Hahmann, S. Konz, *Knot-Removal Surface Fairing using Search Strategies.* Computer Aided Design 30, pp. 131-138, 1998.

[9] S. Hahmann, *Shape improvement of surfaces.* Computing Supplement 13, pp. 135-152, 1998.

[10] M. Hoffmann, *Geometric and Solid Modeling. An Introduction.* Morgan Kaufmann Publishers, 1989.

[11] G. D. Koras, P. D. Kaklis, *On the local shape effect of a moving control point*. Computer Aided Geometric Design, Volume 20, pp. 549-562, 2003.

[12] H. P. Moreton, C. H. Sequin, *Functional Optimization for Fair Surface Design*. ACM SIGGRAPH Computer Graphics, Volume 26, Issue 2, pp. 167-176, 1992.

[13] H. P. Moreton, C. H. Sequin, *Minimum Variation Curves and Surfaces for Computer-Aided Geometric Design*. In: N. S. Sapidis (Ed.), Designing Fair Curves and Surfaces, pp. 123-159, SIAM, ISBN 0-898-71332-3, 1994.

[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, 2nd Edition, ISBN 0-521-43108-5, 1992.

[15] R. J. Renka, *Constructing fair curves and surfaces with a Sobolev gradient method*. Computer Aided Design 21, pp. 137-149, 2004.

[16] J. Roulier, T. Rando, *Measures of Fairness for Curves and Surfaces*. In: N. S. Sapidis (Ed.), Designing Fair Curves and Surfaces, pp. 75-122, SIAM, ISBN 0-898-71332-3, 1994.

[17] G. Taubin, *Geometric Signal Processing on Polygonal Meshes*. State of the Art Reports (STAR), Eurographics '2000, pp. 107-117, 2000.

[18] G. Westgaard, H. Nowacki, *Construction of Fair Surfaces over Irregular Meshes*. Journal of Computing and Information Science in Engineering, Volume 1, Issue 4, pp. 376-384, 2001.