

Óbudai Egyetem

Doktori (PhD) értekezés



Felületmodellek lokális hibáinak javítása fényvonalakkal

Gyurecz György

Témavezető: Dr. Renner Gábor

**Alkalmazott Informatikai és Alkalmazott Matematikai
Doktori Iskola**

Budapest, 2015

Köszönetnyilvánítás

Egy TED[®] előadást hallgatva a következő gondolatra lettem figyelmes: a tudományos média elsősorban a kutatási eredmények közzétételére ad teret, az útnak az ismertetése, amelyen az eredményekig a kutató eljutott a baráti beszélgetések terében vagy még ott sem kap helyet. Pedig, jegyzi meg az előadó, milyen hasznos lenne a mások kutatása során szerzett tapasztalatokból is tanulnunk.

Ha ezt elfogadjuk, és a doktori értekezésre, mint tudományos médiumra tekintünk, akkor ennek a beszámolóknak a hangsúlyai, itt a köszönetnyilvánításban, azok köré kell hogy rendeződjenek, akik vezetése, befolyása, tanácsai nélkül soha nem jártam volna be azt az utat, amit bejártam. Talán sohasem fejlődnek ki, vagy épp erősödnek meg bennem bizonyos képességek, amelyek úgy gondolom, a belátható legmagasabb célt tekintve, az elvégzett munka értelmét alkotják.

A kutatói pályán való elindulásomban elsődleges szerepe volt **Bercsey Tibor** első témavezetőmnek, a BME egyetemi tanárának, aki karizmatikus személyiségével, racionalitást és szervezett hangsúlyozó irányításával, kezdeti lendületet nyújtott a kutatói munka megalapozásához és elkezdéséhez.

Az értekezés, szerteágazó, alapos, több éven át tartó kutatói munka eredménye. Ahhoz hogy elkészült, munkámhoz elengedhetetlen volt témavezetőm, **Renner Gábor**, a SZTAKI tudományos főmunkatársának, kitartást, alaposítást, rendszerességet és igényességet hangsúlyozó irányítása és felügyelete.

Köszönöm többéves lelkiismeretes munkáját, hogy *mindig* kész volt az adott problémán való közös gondolkodásra, és hogy *bármikor* fordulhattam hozzá segítségért és tanácsért!

Az értekezés elkészültéhez jelentősen hozzájárult **Horváth Sándor**, az OE-BGK dékánjának erkölcsi és anyagi támogatása is, amellyel a számos konferencián való részvételt, és szakmai folyóiratokban történő megjelenést biztosította.

A munkám során több szakember is sokat segített a kutatási terület és a problémák alaposabb megértésében. Ezek közül kiemelném **Kalyanmoy Deb** professzort, aki baráti tanácsaival és rendelkezésemre bocsájtott tapasztalataival, sokat segített abban, hogy a genetikai algoritmusokkal és a mérnöki gyakorlatban alkalmazott optimalizációs módszerekkel behatóbban megismerkedjek, valamint **Yifan Chen**-t a Ford Motor Company formatervezői részlegének vezetőjét, akinek rendelkezésemre bocsájtott tapasztalatai értékes részletekkel egészítette ki, és szilárdította meg, az autóiipari felülettervezésről alkotott képem. Itt szeretném megköszönni **Steve Teodosiadis** segítségét is, és a **Solid Modeling Solutions**[®] rendelkezésemre bocsájtott NURBS rutinjait, amivel a görbemodellezési elképzeléseim kivitelezését segítették.

Az értekezés végső formába való öntéséhez többen megjegyzéseikkel, tanácsaikkal értékes segítséget adtak. Köszönöm **Bércesné Novák Ágnes**, **Mikó Balázs** és **Sauvern Ouyang** értékes munkáját, amivel ilyen módon a dolgozat elkészültéhez hozzájárultak.

Végül pedig köszönöm **Galántai Aurél** professzornak a doktori iskola vezetőjének, az értekezés színvonalának emeléséhez nyújtott segítséget, valamint az értekezés befejezéshez adott folytonos biztatást és lendületet.

Tartalomjegyzék

1. Bevezetés	5
2. Előzmények.....	7
2.1. A NURBS görbe	7
2.2. NURBS tenzor szorzat felület.....	8
2.3. A magas minőségű felületmodellek lokális hibáinak okai	10
2.4. Vizuális felületértékelő módszerek	11
2.5. A genetikus algoritmus alapsajátosságai.....	13
2.6. Genetikus algoritmus a görbe- és felületjavításban.....	15
3. Felületek felületértékelő vonalakkal történő javítása	16
4. Célkitűzések.....	20
5. Módszer fényvonalak számítására.....	21
5.1. Fényforrás-egyenesek számítása.....	22
5.2. Fényvonal pontok számítása	23
5.3. A fényvonal számító módszer alkalmazhatósága	24
5.4. Összefoglalás	25
6. Módszer fényvonalak javítására	25
6.1. Hibás szakaszok végpontjainak kijelölése	26
6.2. Fényvonal struktúra hibáinak értékelése	26
6.2.1. Pontsorozat számító módszerek	27
6.2.2. Távolság és szögfüggvények számítása	29
6.2.3. Pontsorozat számító módszerek nyomvonalaik szerinti értékelése.....	31
6.2.4. Pontsorozat számító módszerek hibafüggvényeik szerinti értékelése.....	32
6.3. Fényvonal struktúra javítása	32
6.3.1. A távolság és szögfüggvények simítása	33
6.3.2. Pontsorozat számító eljárások értékelése simítófüggvényeikkel.....	33
6.3.3. Pontok számítása a javított fényvonalakhoz	34
6.3.4. Javított fényvonalak létrehozása	35
6.4. Alkalmazási példák	35
6.5. Összefoglalás	36
7. Módszer a felület javított fényvonalakhoz történő igazítására	36
7.1. A felületek genetikus reprezentációja.....	36
7.2. A genetikai kódolás kiválasztása	37
7.2.1. Bináris kódolás	37
7.2.2. Valós kódolás.....	37
7.3. Keresési tér	38
7.4. A kezdeti populáció létrehozása.....	39
7.5. Fitnessfüggvény	39
7.6. Megállási feltételek	40
7.6.1. Megállási feltételekhez alkalmazott mutatók számítása	40
7.6.2. Biztonsági megállási feltétel	41
7.7. Szűrők	41
7.8. Reprodukció	42
7.9. Skálázás	42
7.10. Kiválasztás (szelekció).....	43
7.11. Keresztezés.....	43
7.12. Mutáció	44
8. A GA megtervezésének munkamódszere.....	45
8.1. A GA alapváltozat létrehozása.....	45
8.2. A GA genetikus keresési folyamat szerinti értékelése	46

8.3. A GA sebesség és megbízhatóság szerinti értékelése	48
8.4. A GA megtervezésében alkalmazott tesztfelületek	49
8.5. A genetikus algoritmus vizsgálatainak megvalósítása számítógépes környezetben	49
9. A GA megtervezése	50
9.1. A keresési tér nagyságának meghatározása.....	51
9.2. A populáció méretének meghatározása	51
9.3. Fitnessfüggvény vizsgálata	52
9.3.1. Szűrő alkalmazása a fitnessfüggvényben	54
9.3.2. A szűrők paramétereinek értelmezési tartománya	55
9.4. Keresztezés vizsgálata	55
9.5. Skálázás vizsgálata	56
9.6. Mutáció vizsgálata	57
9.7. Mutációs arány vizsgálata	58
10. A hibajellemzők hatása a GA működésére	58
10.1. A vizsgálatok munkamódszere	58
10.2. Az algoritmus robusztusságának vizsgálata	60
10.3. Hibanagyság hatása	61
10.4. Hibakiterjedés hatása	63
10.5. Fényvonal-komplexitás hatása	64
10.6. Összefoglalás	66
11. Alkalmazási példák	66
12. Összegzés (tézisek)	67
13. Az eredmények hasznosítási lehetősége	71
14. A dolgozat tartalmának angol nyelvű ismertetése	72
15. Mellékletek	73
15.1. Példák fényvonalak számítására.....	73
15.2. Fényvonal struktúra javítása	74
15.3. Példák felületmodellek lokális hibáinak javítására.....	77
16. Irodalomjegyzék.....	80

1. Bevezetés

Az ipari termékek számos műszaki és esztétikai funkciója csak magas formai minőségű, szabadformájú felületekkel valósítható meg (pl. járműkarosszéria, használati- illetve háztartási eszközök, stb. külső felületei). A számítógéppel segített tervezés (CAD) és a gyártástechnológia fejlődésével párhuzamosan, ezen funkciókkal szemben támasztott felhasználói elvárások egyre magasabbak lesznek, aminek a következménye, hogy a funkciók felértékelődnek, újabb funkciók jönnek létre, és nő a magas minőségű felületek iránti igény.

Ennek eredményeképp a termékek piaci árában ezek a felületek egyre nagyobb részarányt tesznek ki [Luchs et al., 2012], [Talke et al., 2009], [Creusen & Schoormans, 2005], megtervezésük a vállalatok közötti versengés egyik fontos tere. A felületek egyre összetettebbé [Creusen et al., 2010], a minőségi követelmények egyre magasabbá válnak [Homburg et al., 2015]. A termékek piaci versenyében az érvényesüléshez ma már nem elég a magas gyártási minőség, és a minőségi anyagok használata, szükség van újabbnál újabb, innovatív formai kialakítású, magas minőségű felületekre is.

A magas minőségű felületekkel szemben támasztott egyik fontos követelmény, hogy helyi hibáktól (gyűrődések, horpadások, kitüremkedések, ráncok, stb.) mentesek legyenek. A teljes felület méretéhez képest a viszonylag kisméretű, és kiterjedésű helyi hibák, erősen befolyásolhatják a felület minőségét, és alkalmazhatóságát is. Gondoljunk csak az ipari termékek külső megjelenésének esztétikai korrektségére (pl. járművek, használati vagy háztartási eszközök, stb.) az energiaipari berendezések bizonyos elemeinek (pl. turbinalapátok) magas hatásfokára, az optikai eszközök képalkotási minőségére (pl. tükrök) vagy az orvosi protézisek életminőség javító funkciójára (pl. térdprotézis).

A szabadformájú felületek megtervezése, és a felületek számítógépes modelljének (felületmodell) előállítása, CAD rendszerekben történik. A felületek megtervezésének első lépése az elsődleges felületmodell, CAD rendszer eszközeivel vagy felületrekonstrukcióval [Várady & Martin, 2002] történő, előállítása. A felületmodellt ezután méretre és minőségre ellenőrzik, majd a hibákat javítják [Dankwort et al., 2001]. A felületmodellek lokális hibái a felületmodellek interaktív módosításából, a tervezés hibáiból illetve felületrekonstrukciói egyes műveleteinek pontatlanságából erednek.

A felületekkel szemben támasztott növekvő minőségi igények a CAD rendszerek felülettervező eszközeinek folyamatos fejlesztését igénylik; szükség van a felülettervezési feladatok minél gyorsabb, hatékonyabb, összetettebb és lehetőleg automatizált megoldására, az eszközök tervezői igényekhez történő igazítására. Ezen igényeknek megfelelő eszközök kifejlesztése ugyanakkor, még a mai fejlett számítógéppel segített geometriai tervezés (CAGD) mellett is, sokszor nagy kihívást jelent.

A kutatásomban a magas minőségű, szabadformájú felületmodellek lokális hibáit feltáró és javító módszerekkel, ezek problémáival és a problémák megoldásával foglalkoztam.

A felületmodellek hibáinak feltárására több módszer áll rendelkezésre [Hahmann et al., 2008]. Ezek közé tartoznak a felületértékelő vonalak, amelyek különösen a finom, lokális felületi hibák feltárására [Dankwort et al., 2001] alkalmasak. A felületértékelő vonalak a felület felett elhelyezett lineáris vagy pontszerű fényforrások lenyomatai a felületen. Ide tartoznak az árnyékvonalak [Andersson, 2005], fénykontúrok [Poeschl, 1984], [Andersson, 1996], reflexiós vonalak [Klass, 1980], [Kaufman & Klass, 1988] és a fényvonalak [Chen, 1993], [Beier & Chen, 1994].

A tervezői gyakorlatban a fényvonalak alkalmazása az elterjedtebb, aminek két fő indoka van. Egyrészt, a fényvonalak alakja nagyban hasonlít az iparban alkalmazott fizikai laborok reflexiós vonalainak alakjára, másrészt a fényvonalak kezelése a számítógépes környezetben hatékonyabb, mint a reflexiós vonalaké. Gyorsabban számíthatók, alakjuk nem függ a nézőponttól, ami következtében a tervezés során szétválaszthatók a nézeti és a fényvonal kezelési műveletek.

A felületek javításának utólagos és variációs módszerei léteznek. Az utólagos (direkt) módszerek [Kaufmann & Klass, 1988], [Farin & Sapidis, 1989], [Hagen & Santarelli, 1992], [Hermann et al., 1997] jellemzően lokálisan módosítanak, megtartják a felület globális alakját. A variációs (indirekt) módszerek [Hagen & Schulze, 1990], [Bonneau & Hagen, 1994], [Greiner, 1994] többnyire globálisan javítanak, a felület alakjában nagyobb változások is megengedettek.

A kutatásomban a felületek lokális hibáit felületértékelő vonalakkal feltáró és direkt módon javító módszereket vizsgáltam meg részletesen. Ezekben a módszerekben először előállítják a felületértékelő vonalakat, majd a tervező, a saját igényeinek megfelelően a hibás vonalakat kijavítja. Ezt követi a felület javítása, amiben a felületet a javított felületértékelő vonalakhoz igazítják [Dankwort et al., 2001].

2. Előzmények

A felületmodellek lokális hibáinak javítására számos módszert dolgoztak ki. A módszerek kritikai értékelését az 3. fejezetben tárgyalom. Ahhoz hogy az értékelés és a későbbi, általam kidolgozott módszerek ismertetése szakszerű és érthető legyen, előzetes ismeretek szükségesek, amelyeket ebben a fejezetben ismertetek.

A kutatás során a görbék és felületek számítógépes leírásához a CAD rendszerekben széleskörűen alkalmazott NURBS (Non-uniform Rational B-Spline) reprezentációt alkalmaztam [Piegl & Tiller, 1997]. A NURBS a B-Spline reprezentáció legáltalánosabb formája, a CAD rendszerekben jelenleg alkalmazott geometriai reprezentációs módszerek közül ez adja a legtöbb alakmódosítási lehetőséget. Különösen alkalmas a szabadformájú geometria leírására, viszonylag gyorsan, stabil matematikai algoritmusokkal számolható és értékelhető, intuitív, viselkedése kiszámítható.

2.1. A NURBS görbe

A NURBS görbe $\mathbf{C}(u)$ paraméteres, vektorértékű, szakaszos, racionális függvény, amit a bázisfüggvényei, súlyozott kontrollpontjai és csomóvektora határoznak meg:

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n-1} w_i \mathbf{P}_i N_{i,T}^d(u)}{\sum_{i=0}^{n-1} w_i N_{i,T}^d(u)} \quad (1)$$

ahol $N_{i,T}^d(u)$ bázisfüggvény
 u a függvény paramétere
 $\mathbf{T} = (u_i)_{i=0}^{n+d}$ csomóvektor,
 $\mathbf{P}_i, i = 0, 1, \dots, n-1$ kontrollpontok
 $w_i, i = 0, 1, \dots, n-1$ kontrollpontokhoz rendelt súlyok

Bázisfüggvények

A bázisfüggvények $N_{i,T}^d(u)$ $i \in [0, n-1]$ az u paraméter, szakaszos polinomiális függvényei, amelyeket a d fokszámuk és a \mathbf{T} csomóvektoruk teljes mértékben meghatároz. A csomóvektor $\mathbf{T} = \{u_i\}_{i=0}^{n+d}$ monoton növekvő u paraméterértékek (valós számok) u_i sorozata. A függvények, mint ismert a következő rekurzív összefüggéssel állnak elő:

$$N_{i,T}^0(u) = \begin{cases} 1 & \text{ha } u \in [u_i, u_{i+1}) \\ 0 & \text{egyébként} \end{cases} \quad (2)$$

$$N_{i,T}^d(u) = \frac{u - u_i}{u_{i+d} - u_i} N_{i,T}^{d-1}(u) + \frac{u_{i+d+1} - u}{u_{i+d+1} - u_{i+1}} N_{i+1,T}^{d-1}(u) \quad (3)$$

Az egyes bázisfüggvények csak a csomók meghatározta tartományban pozitívak:

$N_{i,T}^d(u) > 0$ $u \in (u_i, u_{i+d+1})$ a tartományon kívül értékük zérus. A bázisfüggvények e tulajdonsága a helyi módosíthatóság egyik lehetőségét teremti meg.

A NURBS görbe érintője

A görbék alakjának értékelésében illetve a görbék egymáshoz illesztéséhez szükségem volt a görbe érintővektoraira. Számításukhoz először alakítsuk át az (1) összefüggést a következőképp:

$$\mathbf{C}(u) = \frac{\mathbf{A}(u)}{W(u)} \quad (4)$$

ahol

$$\mathbf{A}(u) = \sum_{i=0}^{n-1} w_i \mathbf{P}_i N_{i,T}^d(u) \quad (5)$$

$$W(u) = \sum_{i=0}^{n-1} w_i N_{i,T}^d(u) \quad (6)$$

Az u $\dot{\mathbf{C}}_u(u)$ érintővektort, mint ismert az (4) összefüggés differenciálásával kapjuk:

$$\dot{\mathbf{C}}(u) = \frac{\dot{\mathbf{A}}(u)W(u) - \mathbf{A}(u)\dot{W}(u)}{W(u)^2} = \frac{W(u)\dot{\mathbf{A}}(u) - \dot{W}(u)W(u)\mathbf{C}(u)}{W(u)^2} = \frac{\dot{\mathbf{A}}(u) - \dot{W}(u)\mathbf{C}(u)}{W(u)} \quad (7)$$

ahol

$$\dot{\mathbf{A}}(u) = \sum_{i=0}^{n-1} \dot{N}_{i,T}^d(u) \mathbf{P}_i w_i \quad (8)$$

$$\dot{W}(u) = \sum_{i=0}^{n-1} \dot{N}_{i,T}^d \quad (9)$$

A bázisfüggvények deriváltjai

$$\dot{N}_{i,T}^d(u): \dot{N}_{i,T}^d(u) = \frac{d}{u_{i+d}-u_i} N_{i,T}^{d-1}(u) - \frac{d}{u_{i+d+1}-u_{i+1}} N_{i+1,T}^{d-1}(u) \quad (10)$$

Paraméter intervallum transzformáció

Ugyanaz a görbe más paraméterintervallumban is előállítható. Az intervallum transzformációt a folytonos paraméterezés megtartásához, görbék egymáshoz illesztésében alkalmaztam (6.2.1. fejezet). Az $u \in [u_a, u_b]$ paraméter intervallumról a $t \in [t_c, t_d]$ intervallumra az alábbi összefüggéssel térek át [Horváth & Juhász, 1996]:

$$t = \frac{u_b - u_a}{t_d - t_c} (u - t_c) + u_a \quad (11)$$

A görbe ívhosszának számítása

Ívhosszat a fényvonal görbék javításánál, a 6.2. fejezetben számolok. Legyen \mathbf{L} a $\mathbf{C}(u)$, $u \in [a, b]$ görbe, a és b paramétere közé eső ívének hossza:

$$\mathbf{L} = \int_a^b |\dot{\mathbf{C}}(u)| du \quad (12)$$

Az ívhosszat a pontos értéke helyett, a végpontok közé beírt L_n törött vonal hosszával közelítőleg számolom. Legyen $\mathbf{C}(u)$, $u \in [a, b]$ görbe, n egyenlő részre felosztva, és $u_i = a + i\Delta u$, $\Delta u = \frac{b-a}{n}$ és $\Delta \mathbf{C}_i = \mathbf{C}(u_{i+1}) - \mathbf{C}(u_i)$. Az u_i paraméterértékekhez tartozó \mathbf{C}_i görbepontokat összekötő egyenes vonalszakaszok teljes hossza [Horváth & Juhász, 1996]:

$$\mathbf{L}_n = \sum_{i=0}^{n-1} |\Delta \mathbf{C}_i| = \sum_{i=0}^{n-1} \left| \frac{\Delta \mathbf{C}_i}{\Delta u} \right| \Delta u \quad (13)$$

Az n értékét növelve \mathbf{L}_n egyre pontosabban közelíti meg a görbe ívének hosszát.

2.2. NURBS tenzor szorzat felület

A NURBS felület az u és v paraméterek véges tartományának a leképezése a térbe az $\mathbf{S}(u, v) \in \mathbb{R}^3$ vektor-skalár függvénnyel. A felületet az u és v paraméterirányonként értelmezett nyek $N_{i,T_u}^{d_u}$, $N_{j,T_v}^{d_v}$, a fokszámok d_u , d_v , a csomóvektorok $\mathbf{T}_u = (\phi_i)_0^{n_u+d_u}$, $\mathbf{T}_v = (\psi_j)_0^{n_v+d_v}$, a $n_u \times n_v$ kontrollpontok $\mathbf{P}_{i,j} \in \mathbb{R}^3$ és a kontrollpontokhoz rendelt $w_{i,j}$ súlyok határozzák meg:

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^{n_u-1} \sum_{j=0}^{n_v-1} N_{i,T_u}^{d_u}(u) N_{j,T_v}^{d_v}(v) \mathbf{P}_{i,j} w_{i,j}}{\sum_{i=0}^{n_u-1} \sum_{j=0}^{n_v-1} N_{i,T_u}^{d_u}(u) N_{j,T_v}^{d_v}(v)} \quad (14)$$

ahol $(u, v) \in [\phi_{d_u}, \phi_{n_u}] \times [\psi_{d_v}, \psi_{n_v}]$

A $\mathbf{P}_{i,j}$ kontrollpont mozgatása a felület alakját csak az $[u_i, u_{i+d_u+1}] \times [v_j, v_{j+d_v+1}]$ tartományhoz tartozó felületfolt (felületrész) alakját változtatja meg. A felületjavításban érintett kontrollpontok meghatározására a felületleírás ezen tulajdonságát használtam fel.

A NURBS felület normálvektorai

A fényvonal-pontok számításánál (5.2. fejezet) szükség van a felület pontjaira $\mathbf{S}(u, v)$ és normálvektoraira $\mathbf{N}(u, v)$. A normálvektorok a felület u és v irányban vett érintővektorainak vektoriális szorzataként állnak elő.

Az érintővektorok számításához először alakítsuk át a (14) összefüggést a következőképp:

$$\mathbf{S}(u, v) = \frac{\mathbf{A}(u, v)}{W(u, v)} \quad (15)$$

ahol

$$\mathbf{A}(u, v) = \sum_{i=0}^{n_u-1} \sum_{j=0}^{n_v-1} N_{i,T_u}^{d_u}(u) N_{j,T_v}^{d_v}(v) \mathbf{P}_{i,j} w_{i,j} \quad (16)$$

$$W(u, v) = \sum_{i=0}^{n_u-1} \sum_{j=0}^{n_v-1} N_{i,T_u}^{d_u}(u) N_{j,T_v}^{d_v}(v) \quad (17)$$

Az u $\mathbf{S}_u(u, v)$ és a v $\mathbf{S}_v(u, v)$ érintővektorokat a (15) összefüggés u és v szerinti differenciálásával a következőképp kapjuk:

$$\mathbf{S}_u(u, v) = \frac{\mathbf{A}_u(u, v)W(u, v) - \mathbf{A}(u, v)W_u(u, v)}{W(u, v)^2} \quad (18)$$

$$\mathbf{S}_v(u, v) = \frac{\mathbf{A}_v(u, v)W(u, v) - \mathbf{A}(u, v)W_v(u, v)}{W(u, v)^2} \quad (19)$$

ahol

$$\mathbf{A}_u(u, v) = \sum_{i=0}^{n_u-1} \sum_{j=0}^{n_v-1} \dot{N}_{i,T_u}^{d_u}(u) N_{j,T_v}^{d_v}(v) \mathbf{P}_{i,j} w_{i,j} \quad (20)$$

$$\mathbf{A}_v(u, v) = \sum_{i=0}^{n_u-1} \sum_{j=0}^{n_v-1} N_{i,T_u}^{d_u}(u) \dot{N}_{j,T_v}^{d_v}(v) \mathbf{P}_{i,j} w_{i,j} \quad (21)$$

$$W_u(u, v) = \sum_{i=0}^{n_u-1} \sum_{j=0}^{n_v-1} \dot{N}_{i,T_u}^{d_u}(u) N_{j,T_v}^{d_v}(v) \quad (22)$$

$$W_v(u, v) = \sum_{i=0}^{n_u-1} \sum_{j=0}^{n_v-1} N_{i,T_u}^{d_u}(u) \dot{N}_{j,T_v}^{d_v}(v) \quad (23)$$

A bázisfüggvények deriváltjai $\dot{N}_{i,T}^{d_u}(u)$ és $\dot{N}_{i,T}^{d_v}(v)$ [Piegl & Tiller, 1997]:

$$\dot{N}_{i,T}^d(t) = \frac{T - t_i}{t_{i+d} - t_i} N_{i,t}^{d-1}(t) - \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+1,t}^{d-1}(t) \quad (24)$$

ahol $\dot{N}_{i,T}^{d_u}(u)$ esetén: $t = u$, $T = T_u$ és $t_i = t_i^u$

$\dot{N}_{i,T}^{d_v}(v)$ esetén: $t = v$, $T = T_v$ és $t_i = t_i^v$

A felület normálvektorai $\mathbf{N}(u, v)$ a következőképp adódnak:

$$\mathbf{N}(u, v) = \mathbf{S}_u(u, v) \times \mathbf{S}_v(u, v) \quad (25)$$

A felületjavító módszer kidolgozásánál a javítandó felületekről azt feltételeztem, hogy nem önmetsződők, és a paraméterezésük nem szinguláris.

A felületet alakja a csomó vektorral, a bázisfüggvények fokszámával, és a kontrollpontok módosításával befolyásolható, tervezés szempontjából a kontrollpontok módosítása a leginkább hatékony. A fényvonal javítása, illetve a javított fényvonalaknak megfelelő felület előállításához csak a kontrollpontokat változtattam, a fokszám és a csomóvektor változatlan marad. Ebből kifolyólag a görbe és a felület belső folytonossága (amit a csomók esetleges többszörözése befolyásolhatna) nem változik.

2.3. A magas minőségű felületmodellek lokális hibáinak okai

A magas minőségű felületek műszaki és esztétikai szempontból is sima felületek, görbületük eloszlása egyenletes, vagy harmonikusan változó. Nincsenek benne hirtelen változások, amelyek lokális hibákat (egyenletlenségek, ráncok, horpadások, kitüremkedések, törések) okoznak. A felületek modelljeit a CAD rendszerekben rendszeren szabadkézi vázlatokból létrehozott sík / térgörbékől kiindulva vagy felületrekonstrukcióval állítják elő.

Az első esetben a vázlatok pontjait digitalizálják, majd a pontokra folytonos görbét illesztnek, amelyek adott pontossággal közelítik (approximálják) a pontokat. Ezekből a görbékől aztán magasabb szintű eljárásokkal, mint pl. pásztázás, söprés, stb. felületet állítanak elő. A tervezői gyakorlat azt mutatja, hogy a görbék illetve felületek helyi részleteinek kidolgozásában a felhasználó sokszor kényszerül kézi módosításra. A kézi, pontatlan görbe illetve felületjavítás, de a nem megfelelő eljárások, és a tervezési hibák is, sokszor okoznak simasági problémákat. Az ezekkel az eljárásokkal előállított felületek globális alakjában sokszor marad valami „mesterséges”.

A szabadformájú geometria létrehozásának másik, elterjedtebb módja (különösen az autóiparban) a felületrekonstrukció, ami fizikai (fa vagy agyag) 3D modellek mérésével előállított adatpontjaiból indul ki. A felületrekonstrukció folyamatában a modell pontjait először digitalizálják, egy megfelelő háromszöghálót illesztnek a pontokra, majd egy alkalmas approximációs eljárással folytonos felületet hoznak létre, ez az elsődleges felület. A felületet ezután véglegesítik: minőségre és méreteltérésre ellenőrzik, majd simítják.

Az elsődleges felület approximációs eljárással történő előállítása a következőképp történik. Legyen minden \mathbf{p}_i adatponthoz egy (u_i, v_i) paraméterpárral és skalár (ω_i) súlyokkal definiált felületi pont rendelve. A súlyok ω_i az illesztés pontossága és a felület simaság közti arányt kifejező tényezők. A felületillesztés feladata annak a felületnek az előállítása, amely minimalizálja a következő legkisebb négyzetes kifejezést [Várady & Martin, 2002]:

$$F_{lsq}(S) = \sum_i \omega_i^2 \|\mathbf{S}(u_i, v_i) - \mathbf{p}_i\|^2 \quad (26)$$

A felület csomóvektorát, paraméterértékeit és a kontrollpont súlyokat rögzítik és az egyenletet a felület kontrollpontjaira megoldják. A (26) összefüggés azonban nem garantálja, hogy az elsődleges felület sima, lokális hibáktól mentes lesz. Megfelelően sima felületet létrehozni utólagos simítással, vagy az összefüggés különböző feltételekkel való kibővítésével (variációs módszer) lehet.

Az utólagos simítás az elsődleges felületet simító eljárásokkal fokozatosan kisimítja [Kaufmann & Klass, 1988], [Farin & Sapidis, 1989], [Hagen & Santarelli, 1992], [Hermann et al., 1997]. A variációs módszerek [Hagen & Schulze, 1990], [Bonneau & Hagen, 1994] [Greiner, 1994] fizikai alapú (pl. membrán energia, rugalmas lap energia, minimális görbület variáció) simasági kritériummal kibővített felületillesztési összefüggést alkalmaznak, ami egyszerre illeszt és simít is.

Az utólagos simító módszerek jellemzően megtartják a felület globális alakját, és csak a lokálisan módosítanak. A variációs módszerek többnyire globálisan javítanak, a felület alakjában nagyobb változások is megengedettek. A felület finom hibáinak feltárására és a javítására ezért az utólagos simító eljárások a megfelelőbbek [Dankwort et al., 2001].

A hibák a kézi módosításból, a tervezés hibáiból vagy még inkább a fizikai modell és a digitalizálás pontatlanságából erednek. Ezek a hibák nem kiküszöbölhetőek, vagy legalább is a közeljövőben nem várható hogy kiküszöbölhetőek lesznek. Inkább az ellenkezője várható, az, hogy az alapeometria egyre bonyolultabb lesz, amivel a hibák egyre kiterjedtebbé válnak, kezelésük pedig összetettebbé. A CAD programok fel kell hogy legyenek készülve ezekre a kihívásokra, olyan felülettervező eszközöket szükséges a tervező számára fejleszteni, amivel a felületek lokális hibái nagymértékben automatizáltan kezelhetőek, elősegítve ezzel a hatékony munkát.

2.4. Vizuális felületértékelő módszerek

A felület hibáinak javítását a hibák feltárása előzi meg. A hibák feltárása vizuális felületértékelő módszerekkel történik, amelyekkel a felületek minősége, az esztétikai illetve műszaki simasági követelményeknek való megfelelése értékelhető [Hahmann et al., 2008]. A következőkben a különböző módszereket ismertetem röviden, és értékelem a finom felületi hibák feltárásában történő alkalmazhatóságuk szerint.

Paramétervonalak

A paramétervonalak a felület konstans u illetve v vonalait jelenítik meg, a felület alakjáról így alkotott kép részletessége a paramétervonalak sűrűségével szabályozható. A módszer alkalmazása gyors, elsősorban a felület globális alakjának megjelenítésére vagy a közepes illetve nagyobb hibák feltárására használható jól.

Síkmetszetek

A módszer a felületet párhuzamos síkokkal metszi, az így létrejövő metszetgörbékkel a felület érintőlegességi hibáit lehet jól kimutatni. A finomabb felületi struktúrák értékeléséhez szükséges görbületi hibákat azonban nem képes megjeleníteni. A módszer leginkább tervezés közbeni gyors ellenőrzésre, a nagyobb hibák kimutatására alkalmas.

Árnyékolás

A módszer a felület pontjaihoz tartozó normálvektor és egy pontszerű fényforrás közti távolságból megvilágítottságot számol, majd a megvilágítottság függvényében a felület pontjaihoz különböző színárnyalatokat rendel. Érzékenyebb, mint a síkmetszetek módszere, de olyan formában mutatja az eredményeket, amelyek inkább csak a gyors ellenőrzési feladatokhoz megfelelők.

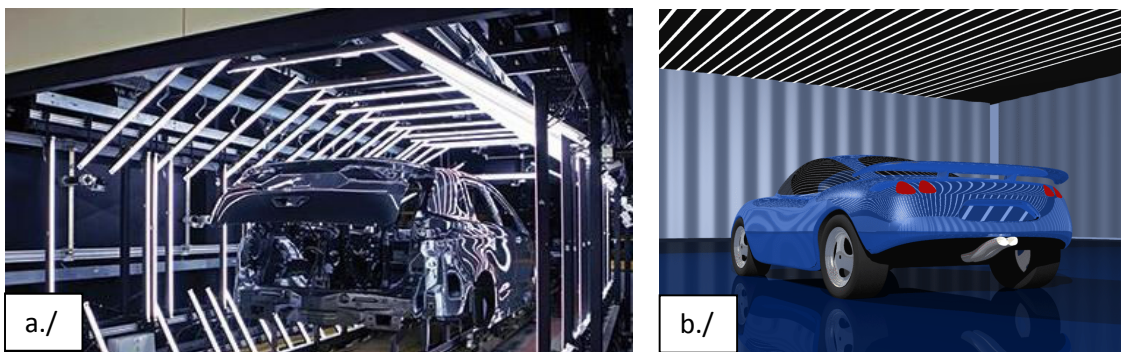
Görbületi térkép

A felület pontjaihoz tartozó főgörbületek összegét, különbségét vagy szorzatát jeleníti meg a felületen, különböző színárnyalatokkal. A színekből következtetni lehet a görbület változásának mértékére és előjelére. Alkalmazása inkább a felület globális simasági viszonyainak értékelésében elterjedtebb.

A felületértékelő vonalak módszerei vonalszerű illetve pontszerű fényforrások fényvisszaverődését használva a felület reflexióis tulajdonságait képezik le. Ide tartoznak a reflexióis vonalak, árnyékvonalak, fénykontúrok és a fényvonalak.

Reflexióis vonalak (reflection lines)

A reflexióis vonalak [Farin et al., 2002] vonalszerű fényforrások tükröződése a felületen. A reflexióis vonalakat a felület azon pontjai alkotják, amelyekhez tartozó normálvektor a nézőponttal és a vonalszerű fényforrás egy pontjával is ugyanakkora szöget zár be. Előnye, hogy a fényforrások a tervezői laboratóriumokban használt fénycsövekhez hasonlóan értelmezhetőek és a reflexióis vonalak (1.b. ábra) megegyeznek a valódi tükröződő felületen létrejövő reflexióis vonalakkal (1.a ábra). Az 1.a. ábra a Ford Motor Company Dearborn-i gyárában készült (Ford Edge külső felületeinek minőségellenőrzése).



1. ábra. Szabadformájú felületek reflexióis vonalai fizikai (a./) és virtuális (b./) közegben.

Hátránya, hogy reflexiós vonalak alakját a nézőpont is befolyásolja, és a felületmodell vizsgálata közbeni nézeti műveletek (elforgatás, transláció) a reflexiós vonalak újraszámítását igénylik, és megdrágítják az alkalmazásukat. Az alapos vizsgálat nehézkes a vonalak változása miatt, ezen felül, két különböző felhasználó, lehet, hogy sohasem fogja ugyanazokat a reflexiós vonalakat látni, és valószínű, hogy ugyanazon felület minőségét különbözően fogja értékelni.

Itt jegyzem meg, hogy Kaufmann és Klass [Kaufmann & Klass, 1988] gyorsan számolható, egyszerűsített, nézőpont független „pseudo-reflexiós”, vonalakat fejlesztettek ki. A vonalakat a felület síkmet-szet görbéinek azonos szög alatt látszódnó érintővektoraiból számolják. A vonalak a nézőpont függetlenség miatt ugyan gyorsan számolhatók, de jócskán eltérnek a valós reflexiós vonalak alakjától, ami miatt az értelmezésük, és így alkalmazásuk is nehézkes.

Árnyékvonalak (shadow lines)

Az árnyékvonalak [Andersson, 2005] a felületre vetülő vonalak árnyékát reprezentálják. Az árnyékvonalak a felület azon pontjaiból állnak, amelyekhez tartozó normálvektor merőleges a vonalszerű fényforrásra. Alakjuk a valódi reflexiós vonalaktól nagyban eltér, nehezen értelmezhető, használatuk nem terjedt el.

Fénykontúrok (isophotes)

A fénykontúrok a felület azonos fényintenzitással tükröződő vonalai [Poeschl, 1984], [Andersson, 1996]. A fénykontúrokat a felület azon pontjai alkotják, amelyekhez tartozó normálvektorok egy pontszerű fényforrásból nézve ugyanakkora szög alatt látszódnak. A pontszerű fényforrás kiküszöböli a nézőpontot, amivel leegyszerűsödik a felület vizsgálata közbeni nézeti műveletek számítása, de a pontszerű fényforrás miatt, a fénykontúrok alakja lényegesen eltér a valós reflexiós vonalakétól.

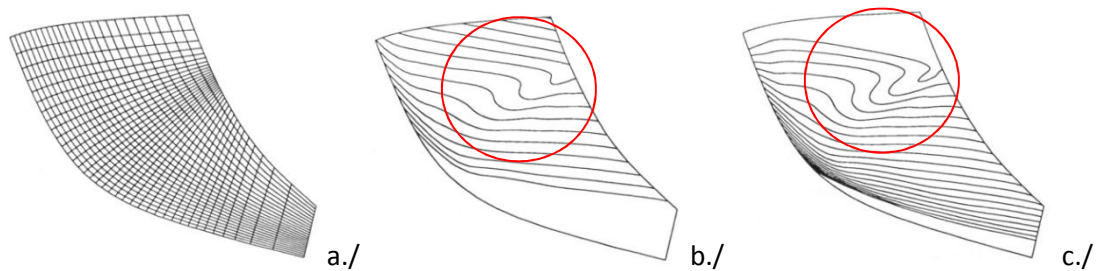
Fényvonalak (highlight lines)

A fényvonalakat 1990-ben kezdték el kifejleszteni, a Michigan-i Egyetemen [Chen, 1993] fejlesztés célja egy olyan felületértékelő módszer megalkotása volt, ami egyesíti a reflexiós vonalak és a fénykontúrok előnyös tulajdonságait: nézőponttól független, reflexiós vonalhoz hasonló értékelési vonalak. A fényvonal a felület azon pontjaiból áll, amelyekhez tartozó felületmerőleges metszi a fényforrás vonalát. A fényvonalak tehát kiküszöbölik a nézőpontot és meghagyják a vonalas fényforrásokat. Az eredmény olyan nézőponttól független értékelési vonalak, amelyek alakja nagyon közel áll a reflexiós vonalak alakjához. A fényvonalak valójában egyszerűsített reflexiós vonalaknak tekinthetők.

A reflexiós vonalakkal ellentétben alakjuk nem fog megváltozni, ha a tervező megváltoztatja a nézőpontját. Ebből következően a nézeti műveletek jól különválaszthatók a fényvonalak kezeléstől. A különválasztás két szempontból is hasznos. Egyrészt, a felhasználói munka szempontjából a tervező egyszerre csak egy dologra kell hogy koncentráljon, ami a munka hatékonyságát növeli. Másrészt hatékonyabbá teszi a nézeti műveletek számítását, mivel nincs szükség a fényvonalak újraszámítására a nézet megváltoztatásánál.

A fényvonalak számításában, ahogy fentebb írtam, szerepelnek a felület normálvektorai, ami érzékenyvé teszi őket a normálvektorok irányváltozásaira. A normálvektor az u és v szerinti parciális deriváltak vektoriális szorzata (25). Ebből adódóan, ha az első derivált függvények közül valamelyik nem folytonos (pl. éles törés van a felületen), az pozicionális hibaként jelenik meg a fényvonal alakjában. Ha a második deriváltak közül valamelyik nem folytonos, az érintőlegességi hibaként jelentkezik, és így tovább.

Ez a tulajdonság különösen alkalmassá teszi őket a lokális hibák feltárására [Chen, 1993], mivel a lokális hibáknál a felületi normálisok különösen változékonyak. Az alábbi ábrán egy köbös B-Spline felületet látható, aminek a központi régiójában egy kis horpadás van.



2. ábra. Példa a paraméter-, fény- és a reflexiós vonalak érzékenysége

A paramétervonalas ábrázolás (a./) nem képes arra, hogy a horpadást hatékonyan megjelenítse. A fényvonal struktúrában jelentkező szabálytalanság (b./) ugyanakkor egyértelműen jelzi a horpadást. A hibát a reflexiós vonalak is jelzik (c./). Vegyük észre, hogy a fényvonalak mennyire hasonlítanak a reflexiós vonalakra.

A fényvonalak a többi vizuális felületértékelő eljárással összehasonlítva, a leginkább alkalmasak a finom felületi hibák feltárására. A felület hibái eggyel nagyobb osztállyal jelennek meg a fényvonalakon, alakjuk nagyban hasonlít az iparban alkalmazott laborok fizikai reflexiós vonalaira, kezelésük a számítógépes környezetben azonban hatékonyabb, mint a reflexiós vonalaké.

Alakjuk nem függ a nézőponttól, a tervezés során jól szétválaszthatók a nézeti és a fényvonal kezelési műveletek. Gyorsabban számíthatók, mint a reflexiós vonalak. A "Class A" felületek (pl. járművek külső felületei) tervezésénél a görbületi hibák kimutatására gyakran használnak fényvonalakat [Farin & Hansford, 2002].

2.5. A genetikusan alapított algoritmusok alapsajátosságai

A felületek felületértékelő vonalakkal történő javításához a felület kontrollpontjai és a javított felületértékelő vonalak közötti komplex, nemlineáris összefüggést meg kell oldani. A jelenlegi módszerek ezt a feladatot klasszikus analitikus és numerikus szélsőérték kereső módszerekkel oldják meg.

Ezen módszerek alkalmazása a mérnöki számításokban bevett gyakorlat, de amíg sok esetben jól használhatók, a komplex, nemlineáris problémák esetén, mint amilyen a kontrollpontok és a fényvonalak közötti összefüggés, nem mindig adnak megfelelő eredményt. Az ilyen esetekben a sztochasztikus optimalizáló módszerek, mint a genetikusan alapított algoritmus (GA) jól alkalmazhatók, ésszerű számítási költségek mellett, megfelelően jó megoldást adhatnak [Deb, 2009].

A genetikusan alapított algoritmusokat a Michigan-i Egyetemen, John H. Holland és munkatársai fejlesztették ki [Holland, 1992], az algoritmust mérnöki problémák megoldásában először David E. Goldberg [Goldberg, 1989] alkalmazta. A következőkben a GA alapsajátosságait ismertetem, röviden.

A szervezetek a természetben evolúció útján adaptálódnak az új környezethez. A genetikusan alapított algoritmusok hasonló módon fejlesztenek ki megoldásokat az adott problémára. Megoldások halmazát (populáció) kezelik, és így egyszerre több irányban keresik a megoldást. A megoldásokat génekből álló kromoszómák kódolják, a gének tartalmazzák a megoldások változtatható paramétereit. A kódolás gyakran bináris (BCGA), ugyanakkor a folytonos változójú problémák esetében a valós kódolás (RCGA) eredményesebb [Herrera et al., 1998].

A genetikusan alapított algoritmusok keresési és megoldástere különvlik. A keresési tér a génekből álló kromoszómákat (genotípusok), a megoldástér az aktuális megoldásokat (fenotípusok) tartalmazza. A megoldások értékeléséhez a genotípusokat fenotípussá kell visszaalakítani.

Az algoritmus a megoldást az összes lehetséges kromoszóma alkotta keresési térben próbálja meg megtalálni. A keresés a *kezdeti populációval* indul, ami a keresési térből véletlenszerűen kiválasztott kromoszómák halmaza. Az egymást követő megoldás generációk elemeit genetikusan operátorok hozzák létre. A legtöbbet alkalmazott operátorok a *kiválasztás*, a *keresztelés* és a *mutáció*.

A genetikus algoritmus a megoldásokat egy előre meghatározott összefüggés, az ún. *fitness* (rátermetség) alapján értékeli, és rangsorolja. A kiválasztás operátor a populáció legjobb egyedeit rátermetségük alapján utódok létrehozásához választja ki. A kiválasztás része lehet az *elitizmus* stratégia is, amivel a legmagasabb rátermetségű egyedek változatlanul kerülhetnek be a következő generációba. Ez a művelet megőrzi az egyes generációk legjobb megoldásait, ami a keresés konvergenciáját tekintve létfontosságú lehet.

A keresztezés a következő (új) generáció kromoszómáit hozza létre. A művelet két (vagy több) jó szülő kromoszóma génjeit kombinálva hoz létre egy vagy több utódot, amelyek potenciálisan a szülő kromoszómáknál rátermettebbek lesznek. A binárisan kódolt GA esetében a kombináció rendszerint a különböző kromoszóma részek cseréjét az RCGA-nál aritmetikai típusú kombinációt jelent.

A mutáció operátor a kiválasztott kromoszómákban egy vagy több gént véletlenszerűen megváltoztat. Célja a populáció változatosságának fenntartása és a még feltáratlan kromoszómák beemelése a populációba. A mutációval megváltoztatott kromoszómák számát a mutációs arány határozza meg.

A keresési algoritmus addig fut, amíg a megállási feltétel nem teljesül. A megállási feltételt a megoldás minősége, vagy a maximális generációs szám határozzák meg. A 3. ábra a genetikus algoritmus működését szemlélteti. Az ábrán g a generációt, g_{max} a maximális generációs számot, $P(g)$ az aktuális generációhoz tartozó populációt jelöli.

```

Program GA
  g:=1 { generáció számláló }
  P(g) Kezdeti populáció létrehozása
  P(g) populáció értékelése

  Ciklus amíg (megállási feltétel) vagy (g = gmax) nem teljesül
    g:=g+1
    P(g) kiválasztás P(g-1)
    P(g) keresztezés
    P(g) mutáció
    P(g) populáció értékelése
  Ciklus VÉGE

Program VÉGE

```

3. ábra. A genetikus algoritmus pszeudo-kódja

A genetikus algoritmus elemeit, paramétereit mindig az aktuális problémához illesztve választják meg és hangolják össze úgy, hogy az algoritmus minél hatékonyabban (gyorsan és megbízhatóan) tudja elérni a megállási feltételt. A genetikus paraméterek közötti összefüggés illetve együttes hatásuk a keresésre összetett és bonyolult. A genetikus algoritmus problémához történő illesztése a paramétereinek változtatásával és az algoritmus tesztelésével lehetséges [Renner and Ekárt, 2003].

A GA konvergenciáját csak egyszerű, elemi genetikus algoritmusokra bizonyították (szkéma elmélet). Ezek szerint, a jó részmegoldások száma a populációban növekszik [Álmos et al., 2003]. A bonyolult, valós feladatokhoz igazított GA-nál matematikailag nem bizonyítható a konvergencia. A konvergenciát, az adott feladathoz igazított mérőszámokkal, a keresési folyamat értékelésével, reprezentatív problémák sokaságán kipróbálva lehet biztosítani. A korai konvergencia eredménye a pontatlan megoldás, a túl lassú konvergencia következménye a rossz keresési határfok. A konvergencia a keresési tér egy pontja környezetének felderítése (*exploitáció*) és a keresési tér ígéretes tartományainak felfedezése (*exploráció*) közti egyensúly fenntartásával szabályozható.

A GA erőssége hogy a megoldást közvetlen matematikai összefüggések nélkül képes megtalálni. A GA-t, jellemzően a következő esetekben használják [Álmos et al., 2003]:

- a tervezési változók száma nagy,
- a keresési tér óriási,
- a tervezési változók hatása a megoldásra bonyolult (nemlineáris) vagy
- a tervezési változók és a megoldás közötti összefüggés ismeretlen,

- a jó megoldás nem állítható elő egyszerű kombinációval,
- sok helyi szélsőérték.

Ezek alapján a GA alkalmas lehet az analitikus és numerikus módszerek kiváltására a kontrollpontok és a fényvonalak közötti komplex, nemlineáris összefüggés megoldásában. A kontrollpontok (tervezési változók) száma jellemzően nagy, pozíciójukat nagy precizitással szükséges meghatározni, ami a keresési teret óriásivá teszi. A kontrollpontok hatása az értékelési vonalakra bonyolult, a megoldás nem állítható elő egyszerű kombinációval.

2.6. Genetikus algoritmus a görbe- és felületjavításban

Azt hogy a GA alkalmas lehet a kontrollpontok és az értékelési vonalak közötti összefüggés megfelelő megoldására az is erősíti, hogy a GA-t előzőleg már számos görbe illetve felületjavítási problémában alkalmazták. Ebben a fejezetben ezen eljárásokat vizsgálom meg, és értékelem.

Genetikus algoritmust a görbejavításban először Márkus András és munkatársai alkalmaztak, az eljárást a [Márkus et al., 1995] cikkben közzétették. Az eljárásban a GA-t, szabadformájú síkgörbék fokszámának csökkentésére és simítására alkalmazzák. A génreprezentáció valós kódolású és a görbe kontrollpontjai szerepelnek benne. A fitness függvény az eredeti és a konvertált görbe reprezentatív izoparaméter pontjai közti távolságot és konvertált görbe simaságát értékeli. A simaság feltételeként a minimális görbületintegrált szabták meg. Az elfogadható görbék az eredeti görbe környezetében definiált sávon belül kell hogy maradjanak. A [Márkus et al., 1997] cikk B-Spline síkgörbék interpolációs problémáinak megoldásához közöl GA-t. A genetikus reprezentáció valós számokat illetve vektorokat. A kód hossza a görbe fokszámától illetve a komplexitástól függően változhat.

A keresztezés operátorra több lehetséges változatot fejlesztettek ki, amelyek az uniform és az egy-pontos módszer [Michalewicz, 1996] kibővített változatai. A keresés különböző szakaszaiban más-más keresztezést javasolnak. A mutáció operátor javasolt változatai a tabukeresés és a szimulált hűtés tulajdonságait egyesítik. A genetikus keresésben a túlságosan rossz egyedeket kizárják, a részben megsértőket pedig büntetik. A fitnessfüggvény javaslatuk szerint a simasági integrál és a büntetés olyan kombinációja, ami a büntetés csökkentése irányába viszi a keresést.

A [Renner & Vida, 2000] cikk söpört alakcsapadékos felületek rekonstrukciójával foglalkozik, amit egy változó sugarú lekerekítés felület példáján mutat be. Az alkalmazott genetikus algoritmus a söpréshez használt gerincgörbe kontrollpontjait és a keresztmetszeti kör sugarát optimalizálja úgy, hogy a kapott felület mért pontoktól való távolságnégyzete minimális legyen.

Munaux [Munaux, 2004] a GA alkalmazhatóságát vizsgálja mért pontokra illesztett térgörbék simításában. Az algoritmus a köbös B-Spline síkgörbékhez kifejlesztett módszerének térgörbékre alkalmazott változata. A génreprezentáció mindkét esetben bináris, és a görbék csomóvektorát tartalmazza. A fitnessfüggvény síkgörbékénél a görbületet, a térgörbékénél a görbületet és a görbe torzióját értékeli. Az algoritmus nem hozott megfelelően pontos eredményeket, nem konvergált olyan jól, mint a 2D esetben.

A görbülettől független torziós komponenssel együtt megnő a fitnessfüggvény értékelési ideje és a keresés többszemponturná válik, amit a felhasznált tervezési módszerrel [Roy et al., 2000] összeállított GA nem tudott megfelelően kezelni. Tapasztalata szerint a változók számának növekedésével a konvergencia sebessége jelentősen csökken. Az általa bemutatott összehasonlításban egy kilencváltozós problémát százszor tovább tartott megoldani, mint egy hasonló, de csak négy változót tartalmazó problémát, ami inkább a rosszul megszerkesztett GA-ra enged következtetni, nem pedig GA alkalmatlanságát mutatja. Ez jelentős hátrány, ami nagyban megnehezíti a módszer alkalmazását a mérnöki feladatokban nemegyszer több ezer pontból álló görbék globális simításában. Lehetővé teszi viszont alkalmazásukat a lokális, pl. az egyes B-Spline görbeszakaszok kezelésében.

Munaux a görbék simítására kidolgozott módszerében a simítás a csomóvektor módosításával, a görbékre felírt energiafüggvények minimalizálásával történik. Egy egyszerű applikációval, grafikusán,

interaktív felületen lehet változtatni a csomó kiosztást illetve figyelemmel kísérni a változtatás hatását a görbe alakjára és az energiafüggvényre. A módszer mérnöki feladatokban való alkalmazhatósága az alacsony automatizáltsága miatt bizonytalan.

A [Lomonosov & Renner, 2008] által kifejlesztett genetikus algoritmust B-Spline felületek, adott karakterisztikus görbékhez történő igazításában használja. A GA a görbe és a felület közti távolságot minimalizálja, az érintett felületrész kontrollpontjainak módosításával. A felület simaságát egy külön posztprocesszáló algoritmus biztosítja. Az algoritmus a felület kontrollpontjait módosítja, simasági mutatónak a görbületi integrál változását tekinti.

[Case et al., 2002] majd később [Keith et al. 2008] illetve [Yi et al., 2008] a felületet geometriai alakrajátosságokra bontja, amelyek funkcionális illetve esztétikai értékeléséhez GA-t alkalmaz. A fitnessfüggvényében funkcionális és esztétikai követelményeket: egyszerűség, stabilitás, puhaság, dominancia stb. szerepeltetnek. Az esztétikai fogalmakhoz alakrajátosságokat rendelnek [Giannini & Monti, 2002], amiket felhasználókkal kalibrálnak. Az egyes megoldások rátermettségét interaktívan vagy félig automatizáltan értékelik. A módszert a felület globális formáját alkotó alakrajátosságok szintézisében alkalmazzák. A GA-t, több görbe és felületsimítási feladatban alkalmazták sikeresen.

Következtetések

A felületek simításában a GA-t a globális simítási feladatokban alkalmazzák a lokális simítás problémái közül csak speciális feladatra fejlesztettek ki módszert (lekerekítés felület).

Az algoritmusok kódolása legtöbbször valós, a bináris kódolás alkalmazása nem járt sikerrel. A felületeket a génekben legtöbbször a kontrollpontok vagy a csomóvektorok reprezentálják. A kontrollpontokat tartalmazó reprezentációkat több sikerrel alkalmazták. Ennek az a valószínű oka, hogy a kontrollpontok a felület paraméterei közül a legnagyobb kihatással felület alakjára, és ezzel a módosításra is. A fitnessfüggvények simasági feltételeket is tartalmaznak, amelyet a felület görbületéből számolnak.

3. Felületek felületértékelő vonalakkal történő javítása

A felületek felületértékelő vonalakkal történő javításában direkt és indirekt módszereket alkalmaznak [Bosinco et al., 1998], [Dankwort, C.W., Podehl, G., 1998], [Dankwort et al., 2001]. A direkt megközelítésben a felületértékelő vonalakat javítjuk, majd a felületet a javított fényvonalaknak megfelelően módosítjuk. Az indirekt módszerrel a felületértékelő vonalak alakja csak közvetetten javítható, mivel a vonalak alakja helyett itt a felület differenciálgeometriai jellemzőinek (pl. főgörbületek) eloszlását írjuk elő (simítjuk ki), és a felület alakját ennek megfelelően módosítjuk [Rando & Roulier, 1991], [Higashi et al., 1999]. Ez utóbbi megközelítés hátránya, hogy ebben az esetben a felhasználó nem tudja pontosan milyen felületértékelő vonalakat kap majd eredményül. A javításban a vonalstruktúra (vonalak alakja és mintázata) jelentősen eltérhet attól, amit a tervező látni szeretne [Zhang, et al., 2001]. Ez a módszer inkább akkor használható jól, amikor a felület geometriájában még nagyobb változtatások is elképzelhetők. A lokális hibák javításában a direkt módszer a hatékonyabb mivel a felületértékelő vonalak minél precízebb javítása, előírása szükséges, nagyobb változtatások a felület geometriájában már nem elképzelhetők. Ezen felül, ha felület minőségének értékelésére felületértékelő vonalakat használunk ésszerűbb a direkt módszert alkalmazni és a javítást is ezeken a görbéken keresztül végrehajtani.

Direkt javítást alkalmazó módszerek kritikai értékelése

A következőkben már csak a direkt módszert alkalmazó felületjavító eljárásokat vizsgálom. A vizsgálatokban először röviden leírom a működésüket, majd kritikai szempontból megvizsgálom őket.

Az eljárásokat kronológiai sorrendben vizsgáltam meg. A direkt módszert először Klass [Klass, 1980] alkalmazott. A felület értékeléséhez reflexiós vonalakat használ, javításukat azonban nyitva hagyta és csak a felület javításával foglalkozott. A felületjavítás a reflexiós vonalak és a felület között felírt inhomogén lineáris differenciálegyenlet rendszer megoldásán alapul. Az eljárás fő problémája hogy az

egyenletrendszert csak egyszerűsítésekkel, közelítőleg lehet biztonsággal megoldani, ami miatt az eljárás csak kis hibanyagosság és kis hiba kiterjedési tartományban alkalmazható.

A problémát később a [Kaufman & Klass, 1988] cikkben leírt eljárás, egyszerűsített (pszeudo) reflexiós vonalak alkalmazásával próbálja meg feloldani. A felület javításában az eljárás egy \mathbf{R} pontot keres meg a felhasználó definiálta \mathbf{W} vektorokon úgy, hogy a szög a felület érintővektora és a javított reflexiós vonal érintője között ugyanaz maradjon, mint az eredeti reflexiós vonal esetén.

Az \mathbf{R} meghatározásához ugyanakkor nemlineáris differenciálegyenlet rendszer megoldása szükséges, amelynek megoldásához az egyenletrendszert szintén csak egyszerűsítésekkel, közelítőleg lehet biztonsággal megoldani. Az eljárás az előzőhöz képest gyorsabb ugyan, de ugyanúgy csak kis felületrész és kis hiba javítására alkalmazható és a kapott eredmények sem mindig elfogadhatók. Ezen felül, a pszeudo-reflexiós vonalaknak nincs közvetlen fizikai jelentésük, és a tervezésben nehezen értelmezhetők. Az előző eljáráshoz hasonlóan, ez a módszer sem foglalkozik a felületértékelő vonalak javításával.

A [Chen et al., 1997] által közölt eljárás felületértékelő vonalakként fényvonalakat alkalmaznak, és már nemcsak a felület, de a fényvonalak javításával is foglalkoznak. A javításhoz a Stewart [Stewart, 1991] által kifejlesztett „Direct Curve Manipulation” – nak elnevezett módszert alkalmazzák. Az eljárás a fényvonalakat egyenként javítja, a kijelölt hibás görbeszakasz alakját a felhasználó által kijelölt görbepont mozgatásával manipulálja, a görbe alakjának finomhangolását különböző további paraméterek iteratív módosításával éri el. A javítás folyamata egyáltalán nem automatizált, ami miatt alkalmazása hosszadalmas és fáradtságos. A módszer további hiányossága, hogy a fényvonalakat egyenként javítja, ami miatt a nagyobb vagy bonyolultabb felületrészeken történő alkalmazása hosszadalmas, nagy tervezői tapasztalatot igényel.

Az eljárásuk következő lépése a felület javítása. A javításhoz a fényvonal-pontok, és a felületi kontrollpontok módosítása közti összefüggést leíró nemlineáris egyenletrendszer megoldása szükséges. A megoldás pontossága nagyban függ a hiba nagyságától és a felület görbületi viszonyaitól. Ez annak a következménye, hogy az összefüggés a Taylor sor első tagjainak lineáris approximációján alapul.

Zhang és Cheng [Zhang & Cheng, 1998] által közölt eljárás a felület értékelésére szintén fényvonalakat használ. A fényvonalak javításában először a kijelölik a hibás fényvonalszakaszok végpontjait, majd a végpontok közé a hibás fényvonal szakaszokat helyettesítő köbös Hermite íveket szerkesztenek. A végpontok pontos kijelölése fontos részlete a javításnak, de nem közölnek rá módszert. A fényvonalak javítása itt is egyenként történik és csak elemien kisméretű javítást lehetővé tevő ívekkel.

Az eljárásuk következő lépésében a felületet a javított fényvonalakhoz igazítják. Az igazításhoz a fényvonal-pontok és a felület kontrollpontjai közötti összefüggést leíró nemlineáris egyenleteket oldják meg, amit a megoldáshoz linearizálnak. Ehhez a felület pontjait és első parciális deriváltjainak értékét becsülik meg, a javított fényvonalak mentén. A linearizálás következtében a fényvonal változtatás lehetséges mérete kicsire adódik, aminek következtében a módszer alkalmazása számos felhasználói beavatkozást igényel.

A [Nishiyama et al., 2007] cikkben közölt eljárás a felületértékeléshez szintén fényvonalakat számol, fényforrások azonban nem egyenesek és egymással párhuzamosak, hanem kör alakúak és koncentrikusak. A fényvonalak javítása teljes mértékben a fényvonalak u - v paraméterterbéli reprezentációján történik. A javítás első lépéseként a felhasználó interaktívan kijelöli a hibás fényvonal szakasz végpontjait, majd az eljárás a végpontok közti hibás szakaszt köbös Hermite ívvel helyettesíti.

A felület javításához a módszer először azonosítja a felület érintett kontrollpontjait. Ezt követően a szükséges kontrollpont módosításhoz a kontrollpontok és a fényvonal közti nemlineáris összefüggést Newton módszerrel megoldja.

A kör alakú fényforrásokkal létrehozott fényvonalai nehezen értékelhetők, mivel formájuk nagyban eltér a reflexiós vonalak alakjától. A fényvonalak javítására alkalmazott módszer úgy, mint az előzőek, csak egyszerűbb hibák esetén alkalmazható. Ennek az okai is ugyanazok: az interaktív kijelölés pon-

tatlansága és az, hogy az eljárás fényvonalanként javít, nem veszi figyelembe a fényvonalak struktúráját.

Az [Andersson,1996] cikkben közölt eljárás fénykontúrokat interpoláló felületet állít elő. A felület előállításához elsőrendű parciális differenciálegyenlet (EPD) megoldása szükséges. A felület szélének meghatározásához C1 folytonossági peremfeltételek illetve az EPD integrálása szükséges. További peremfeltételeket előírni, vagy az EPD megoldásához szükséges feltételekből kihagyni, nem lehetséges. Ebből kifolyólag, a felület lokális módosítása szintén nem lehetséges. Ezen felül a felület létrehozásához csak véges számú görbét használ, de nem ad egyértelmű leírást hogyan kell megfelelő felületmodellt előállítani.

A [Loos et al., 1999] által közölt eljárás a fénykontúrok hibáit a felületi fényintenzitás gradiensével reprezentálja. A fénykontúr mentén gradiens függvényt számol, amit másodfokú függvénnyel simít, a simított és az eredeti függvény között egy hibafüggvényt definiál, amit aztán a felület paramétereinek programozott módosításával minimalizál. Loos módszerében a másodfokú függvény az alkalmazhatóságot a rövid és egyszerű fényvonalak irányába tolja el.

Andersson [Andersson, 2005] cikkében a felület árnyékvonalakkal történő javításával foglalkozik. A felület simaságát nem a felületértékelő vonalak mentén, hanem a közöttük lévő távolság egyenetlenségeként értelmezi. Ez az elképzelés hasznos, mivel a fényvonalakat struktúra részeként kezeli, amit a merőleges távolságok jellegének megváltoztatása nélkül kell kisimítani. A cikk ugyanakkor nem tér ki a javítás részleteire, inkább csak a koncepciót fogalmazza meg.

A módszer a felületjavításhoz az árnyékvonalak és a felület módosítása közötti összefüggést lineáris elsőrendű hiperbolikus parciális differenciálegyenlettel írja le. Az egyenlet megoldásához számos egyszerűsítés és felhasználói interakció szükséges, ami miatt a módszer hatékonysága nagyobb hiba kiterjedések esetén erősen lecsökken.

A [Zhang & Lai, 2001] cikkben leírt eljárás pszeudo-reflexiós vonalakat alkalmaz. A vonalakat genetikus algoritmussal simítja, célja több alternatív megoldást létrehozni, amelyek megfelelően simák. A genetikus reprezentáció a reflexiós vonal egy pontjának elmozdítását meghatározó vektor, ami az adott pontban, az eredeti felület normálvektorával párhuzamos. Így a javítás mindvégig kapcsolatban marad az eredeti felülettel. A kromoszóma a hibás reflexiós vonal pontjainak elmozdulását meghatározó vektorokból áll. Az új reflexiós vonalat a keresztezések meghatározta elmozdulásokból generált új pontok határozzák meg. A módszer a vonalakat egyenként javítja, és nem veszi figyelembe, a vonalak egy struktúrájában egybetartoznak.

A pontok elmozdításának irányát a hibás felület normálvektorához köti, amivel a javítás minden egyes iterációs lépése a hibás felület normálvektorait veszi alapul a módosításhoz. A példaként bemutatott egyszerű esetekben az eljárás működőképes lehet, de ez a stratégia a bonyolultabb vonalak esetén gátolhatja az algoritmust a megfelelő minőségű megoldások előállításában. A cikk számos, az algoritmus alkalmazhatóságát és hatékonyságát befolyásoló genetikus paraméter alkalmazási részleteit nem közli (pl. kódolás, keresztezés, paraméterek, stb.). A közölt analízisből nem lehet arra következtetni, hogy a módszer a vonalak bonyolultságát tekintve mennyire általánosítható. A módszer fő hiányossága, hogy nem foglalkozik a javított vonalakból való felületet előállításával, nem küszöböli ki a [Kaufman & Klass, 1988] módszer felületjavítási fogyatékoságait.

A fényvonalak, vagy általában az értékelési vonalakhoz az eljárások a vonalakat csak diszkrét pontjaikban és csak közelítőleg határozzák meg. Ehhez a felületet u és v paramétervonalai mentén felosztják, majd az értékelési vonal definíciójának megfelelően diszkrét pontokból álló távolságfelületet számolnak. A felület szintvonalain elhelyezkedő pontok az értékelési vonalak pontjai. [Beier & Chen, 1994] és később [Jun-Hai et al. 2003]. Ennek a megközelítésnek két hátránya van. Az egyik, hogy a pontosságot globálisan, a paramétervonalak menti felosztást sűrítve érik el. Ez a pontosítási módszer drága, hiszen a sűrítésnél minden pontot újra kell számolni. A másik, hogy a tervező nem a pontok sűrítésében, hanem a pontosság közvetlen megadásában érdekelt.

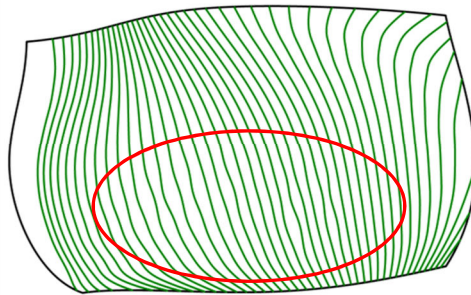
Következtetések

A meglévő módszerek hiányosságai a következő területeken jelentkeznek: a felületértékelő vonalak pontos és felhasználó által jól kezelhető számítása, hibás értékelési vonal szakaszok kijelölése és javítása és a javított vonalaknak megfelelő felület előállítás.

Sok módszer nem ad eljárást, vagy útmutatást hogyan kellene a hibás értékelési vonalszakaszokat kijelölni. Azok a módszerek, amelyek foglalkoznak ezzel a kérdéssel, a szakaszok végpontjait interaktívan, és ebből kifolyólag pontatlanul határozzák meg. Az interaktivitás fontos, mivel a felhasználó az egyéni, gyakorlatban szerzett tapasztalatok, és a felhasználói igényeknek megfelelően kell hogy eldöntse, hol van a jó és a hibás felület, illetve vonalstruktúra határa. Az interaktív kijelölés a bizonytalan felhasználói precizitás miatt ugyanakkor problematikus. Ha hibás részek maradnak ki a kijelölésből, a javítás eleve pontatlan lesz, ha jó részek is belekerülnek, akkor a felhasználó szándékával ellentétesen jó részeket is módosulhatnak. A felhasználó a gyors interaktív és pontos kijelölésben is érdekelt, amelyek az egymással ellentétes elvárások.

A módszerek a hibás értékelési vonal szakaszokat egyenként és egyszerű helyettesítő görbékkel javítják. A fényvonalak egyenkénti javításának koncepciója azért problematikus, mivel a javítás a szomszédos fényvonalak alakjára is kihat, ezért az egyenkénti javítás bonyolult, gyakorlatilag manuális művelet, amihez nagy felhasználói tapasztalat szükséges. Az egyszerű helyettesítő görbék erősen behatárolják a javítható értékelési vonal hiba nagyságát, kiterjedését és komplexitását.

Egy ilyen, nagyobb kiterjedésű hibára mutat példát a 4. ábra. A hibás fényvonal szakaszokat tartalmazó felületrészt piros görbével jelöltem.



4. ábra. Példa nagy kiterjedésű lokális felülethibára (piros görbével körberajzolva)

A javított értékelési vonalaknak megfelelő felület előállításában a fő probléma az értékelési vonalak pontjai és a felület kontrollpontjai közti nemlineáris összefüggés. Az összefüggést a módszerek egyszerűsítésekkel linearizálják, és klasszikus analitikus és numerikus szélsőérték kereső algoritmusokkal közelítőleg oldják meg. A közelítő megoldás következménye, hogy a módszerek alkalmazhatósága a hiba nagyságának és kiterjedésének növekedésével erősen csökken.

4. Célkitűzések

A szakirodalom elemzése alapján megállapítható, hogy a magas minőségű szabadformájú felületek lokális hibáinak javítására, a direkt, felületértékelő vonalak javítását alkalmazó módszerek a leginkább alkalmasak. A szakirodalom elemzéséből az is megállapítható, hogy a jelenlegi direkt módszerek, megelégednek az elemien kis kiterjedésű és méretű hibák javításával. A hibák mérete, kiterjedése és összetettsége (hiba jellemzői) ezzel ellentétben a mai felületeken sokszor széles határok között mozognak. Ennek az ellentétnek az a következménye, hogy a meglévő módszerek alkalmazása nehézkes, hosszadalmas, nagy felhasználói tapasztalatot igényel, a tervező sokszor inkább manuálisan javít.

- Céлом egy olyan új, felületmodellek lokális hibáinak javítására alkalmas direkt módszer megalkotása, amely széles határok között mozgó hibajellemzők esetén is jól alkalmazható. A módszert a lokális hibák feltárására leginkább alkalmas fényvonalakra szándékozom kidolgozni.

A módszert a jelenlegi módszerek hiányosságainak kiküszöbölésével szándékozom megalkotni. A hiányosságok, és a kiküszöbölés tervezett módjai a következők:

A jelenlegi módszerek nem foglalkoznak megfelelően a fényvonalak számítási pontosságának kérdésével.

- Céлом egy olyan fényvonal számító eljárás kidolgozása, amelyben a fényvonalak nagy pontossággal számíthatók, és a jelenlegi eljárásoktól eltérően a felhasználó közvetlenül írhatja elő a fényvonalak számítási pontosságát.

A módszerek hibás fényvonal-szakaszokat kijelölő eljárásai teljes mértékben interaktívak, és nem adnak megfelelő megoldást a szakaszok kijelölésének pontosítására. Ennek a következménye, hogy a fényvonalak javításából hibás vonalszakaszok maradhatnak ki, illetve a javítás során jó fényvonal szakaszok is módosulhatnak.

- Céлом egy újabb művelet beiktatásával pontosítani a kijelölési eljárást.

A jelenlegi módszerekben a felületértékelő vonalak javításánál a tervező legtöbbször a CAD rendszer olyan eszközeire kell hogy hagyatkozzon, amelyeket nem erre a feladatra fejlesztettek ki. A közreadott értékelési vonal javító módszerek egyáltalán nem automatizáltak, ami miatt alkalmazásuk hosszadalmas és fáradságos. A módszerek további hiányossága, hogy a fényvonalakat egyenként javítják, ami miatt a nagyobb felületrészekben történő alkalmazásuk nagy felhasználói tapasztalatot igényel.

- Céлом olyan eljárás kidolgozása, amely képes nagyobb kiterjedésű és összetettebb fényvonal hibák javítására is, nagyrészt automatizált, és a használatához nem kell nagy tervezői tapasztalat. Ezt olyan eljárással szándékozom elérni, amely a hiba értékelésébe és javításába a fényvonalak alakja mellé beveszi a fényvonalak mintázatát is, és a fényvonalak struktúráját javítja.

A fényvonalak és a fényvonalakat előállító felület paramétereinek közti összefüggés bonyolult, nemlineáris. A felület paramétereinek meghatározásához a jelenlegi eljárások sok egyszerűsítést alkalmaznak a számításokban, ami korlátozza az alkalmazhatóságukat, és alkalmatlanná teszi őket a nagyobb és kiterjedtebb hibák kezelésére.

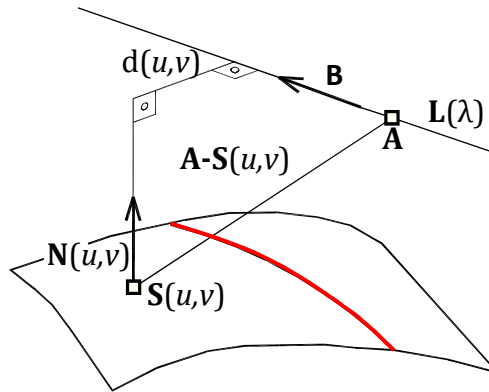
- Céлом egy olyan eljárás kifejlesztése, amely képes a javított fényvonalaknak megfelelő felület paramétereit megfelelő pontossággal és hatékonysággal meghatározni, széles értékhatárok között változó hibajellemzők esetén is. A feladatot a problémához illesztett genetikus algoritmussal szándékozom megvalósítani.

5. Módszer fényvonalak számítására

A fényvonal-pontok a felület azon pontjai, amelyekhez tartozó felület-normális és a lineáris fényforrás-egyenes egymást metszik, azaz a közöttük lévő merőleges távolság zérus. Legyen $L(\lambda)$ a fényforrást reprezentáló egyenes:

$$L(\lambda) = \mathbf{A} + \mathbf{B}\lambda \quad (27)$$

ahol λ az egyenes paramétere, \mathbf{A} az $L(\lambda)$ egyenes egy pontja, \mathbf{B} a fényforrás irányát meghatározó irányvektor (5. ábra).



5. ábra. A fényvonal pontjainak értelmezése

Mint ismeretes az $L(\lambda)$ fényforrás-egyenes és a felület $S(u,v)$ pontjához tartozó $N(u,v)$ normálvektor egyenese közti merőleges távolság:

$$d(u,v) = \frac{|(\mathbf{B} \times \mathbf{N}(u,v)) \cdot (\mathbf{A} - \mathbf{S}(u,v))|}{|\mathbf{B} \times \mathbf{N}(u,v)|} \quad (28)$$

Az $S(u,v)$ pont a fényvonalon van, ha teljesül [Chen, 1993]:

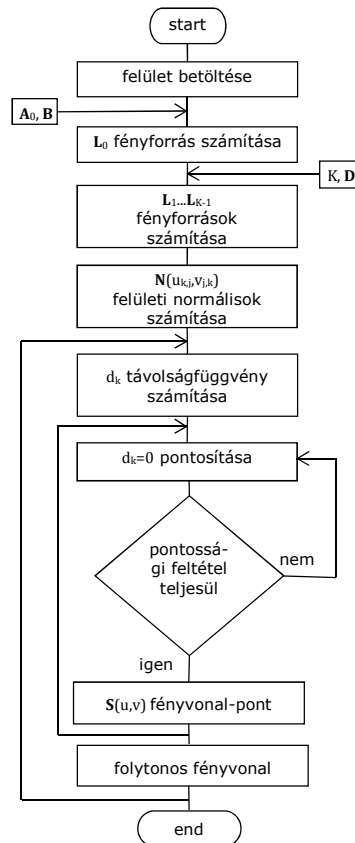
$$d(u,v) = 0 \quad (29)$$

A fényvonal pontjainak meghatározásához a (29) összefüggést a felület kontrollpontjaira kell megoldani. A számítás nemlineáris függvényeket tartalmaz, amelyekre legtöbbször nem lehetséges analitikus megoldást találni. A fényvonal számító eljárások ezért a fényvonalak pontjait csak közelítőleg határozzák meg. A lokális hibák feltárásához és javításához, különösen a kiterjedtebb, bonyolultabb hibák esetén a fényvonal-pontok minél pontosabb meghatározása szükséges. A pontosság növeléséhez a jelenlegi módszerek a pontok számát (u és v paraméter menti felosztás sűrűségét) növelik. A [Beier & Chen, 1994] és később [Jun-Hai et al. 2003] által közölt módszerek a fényvonal-pontok számításához a felületet u és v paraméterirányban egy előre meghatározott sűrűséggel felosztják, és az (28) összefüggést felhasználva diszkrét pontokból álló távolságfelületet számolnak. A távolságfelület vízszintes síkokkal metszik, és az így létrejövő szintvonalak azon pontjait tekintik fényvonal-pontoknak, amelyek a $d(u,v)=0$ összefüggést a leginkább kielégítik.

A módszer egyik hátránya hogy ez utóbbi műveletből kifolyólag a fényvonal-pontok meghatározási pontossága fényvonal-pontként más és más. Másrészt, a fényvonal-pontok helyét pontosítani csak indirekt módon, és költségesen lehet, az u és v irányban történő felosztás sűrűségének növelésével. A tervező a fényvonal-pont helyének adott pontossággal történő meghatározásában érdekelt, nem a felosztás sűrűségének változtatásában. A sűrítéssel a pontosság csak közvetve befolyásolható, nem tudni mekkora sűrítést kell alkalmazni az elfogadható pontossághoz. Ez utóbbi miatt a sűrítési eljárást rendszerint többször meg kell ismételni, ami rendkívül megdrágítja a módszert. Az általam ajánlott

megközelítés a pontok számának növelése helyett a pontok helyzetét pontosítja, a felhasználó által előírt pontossággal.

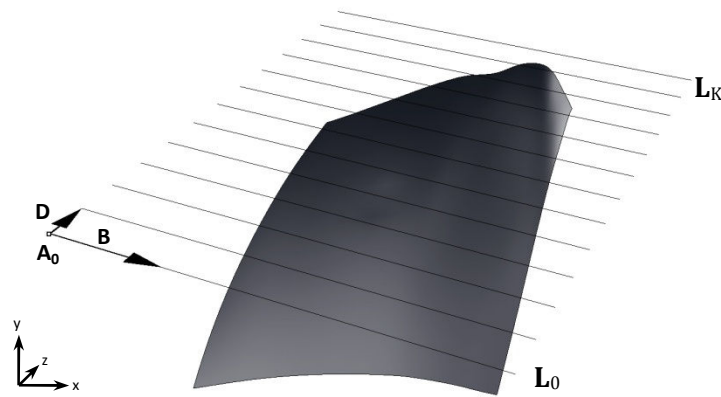
A módszerben a fényforrásokat egyenesekkel, a felületet diszkrét pontjaival és a pontokban értelmezett felület-normálisokkal reprezentálom. A fényforrásokból és a felület normálisokból a (28) összefüggést felhasználva diszkrét értékekből álló távolságfüggvényeket számolok. A függvények zérushelyei felületi pontokat határoznak meg, amelyek a (29) összefüggésnek megfelelően a fényvonal-pontok. A zérushelyeket a felhasználó által előírt tűréssel keresem meg, a 9. ábrán látható algoritmussal, majd meghatározom a hozzájuk tartozó felületi pontokat. A fényvonalak folytonos reprezentációjához a pontokra görbét illeszttek, amely a felhasználó által előírt pontossággal közelíti (approximálja) a pontokat. A módszer blokk diagramja a 6. ábrán látható.



6. ábra. Fényvonalakat létrehozó módszer blokk diagramja

5.1. Fényforrás-egyenesek számítása

A fényforrásokat reprezentáló egyenesek a felület felett, közös síkban, egymással párhuzamosan helyezkednek el (7. ábra). Az $\mathbf{L} = \mathbf{L}_{k=0} \dots \mathbf{L}_K$ fényforrás sorozat \mathbf{L}_0 első elemét a felhasználó az \mathbf{A}_0 ponttal és \mathbf{B} vektorral határozza meg. A sorozat többi elemét a \mathbf{D} távolságvektor és a fényforrások K száma határozza meg.



7. ábra. Fényforrásokot reprezentáló egyenesek számítása

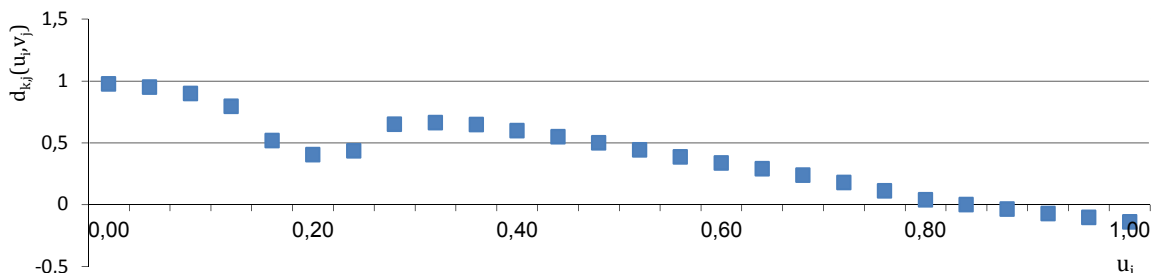
A felületminőség vizsgálata közben a \mathbf{D} vektorral a fényforrások síkja, és a fényforrások egymás közötti távolsága is változtatható. Az így létrehozott új helyzetekben előálló fényvonalak összehasonlításával részletesen megvizsgálható a felület minősége.

5.2. Fényvonal pontok számítása

A fényvonal-pontok számításában a felületet u és v izoparaméter vonalai mentén számolt $\mathbf{S}(u_i, v_j)$, $i=0..I-1, j=0..J-1$ diszkrét pontjaival és a pontokhoz tartozó felület-normálisokkal $\mathbf{N}(u_i, v_j)$ helyettesítem. A fényvonal-pontok számításához a (28) összefüggés helyett, az előjeles távolságot adó (30) összefüggést alkalmaztam, amivel a feladatot minimumkeresés helyett a könnyebben kezelhető gyökkeresési problémává alakítottam át:

$$d(u, v) = \frac{(\mathbf{B} \times \mathbf{N}(u, v)) \cdot (\mathbf{A} - \mathbf{S}(u, v))}{\|\mathbf{B} \times \mathbf{N}(u, v)\|} \quad (30)$$

A (30) összefüggésbe u_i és v_j paramétereket behelyettesítve diszkrét értékekből álló távolság felületet kapok. A felület zéruspontjainak behatárolásához az $u_i, i=0..I-1$ paramétervonalai mentén definiált $d_{k,j}(u_i, v_j), j=0..J-1, k=0..K-1$ távolságfüggvények gyökeit keresem meg. A 8. ábra egy példát mutat a $d_{k,j}$ függvényre. A függvényt a 7. ábrán látható felületen számoltam, az $L_{k=3}$ fényforráshoz és a $v_j = 0.24, j=8$ paramétervonalhoz tartozik. A függvény gyöke az $u_i = 0.8, u_i = 0.84$ értékek között található.



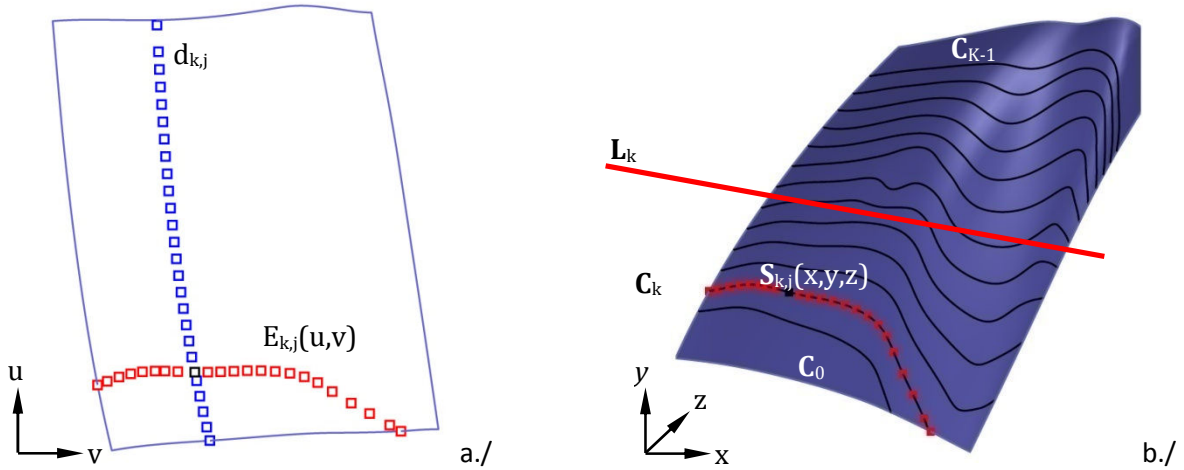
8. ábra. Példa a $d_{k,j}$ távolságfüggvényre

Mivel a függvények csak véges számú pontból állnak, a zérushelyeik csak közelítőleg ismertek. A fényvonal-pontok precízebb meghatározásához a zérushelyek pontosítása szükséges, amihez a 9. ábrán látható algoritmust alkalmazom. Az algoritmus intervallumfelező keresést alkalmaz, a keresési intervallum határait előjelfüggvénnyel, a $\text{sgn}(d_{k,j}(u_i, v_j)) + \text{sgn}(d_{k,j}(u_{i+1}, v_j)) = 0$ feltételből határolok be. A zérushelyet a felhasználó által megadott tűréssel határozom meg. A pontosítási ciklus biztonsági megállási feltétele a maximális ciklusszám.

ciklus amíg $i < I-1$
 $u := i \cdot \text{lépésköz}$
ciklus amíg $j < J-1$
ha $d_{k,j-1} > 0$ és $d_{k,j+1} < 0$ vagy $d_{k,j-1} < 0$ és $d_{k,j+1} > 0$ akkor
ciklus amíg $d_{k,i} > \text{tűrés vagy ciklusszám}$
 $v_{k,j} := ((v_{k,j-1} \cdot d_{k,j+1}) - (v_{k,j+1} \cdot d_{k,j-1})) / (d_{k,j+1} - d_{k,j-1})$
 $d_{k,j} := (L_k \times N(u,v) \cdot (A_k - S(u,v)))$
ha $d_{k,j} < 0$ és $d_{k,j-1} > 0$ akkor
 $d_{k,j-1} := d_{k,j}$
 $v_{k,j-1} := v_{k,j}$
elágazás vége
ha $d_{k,j} < 0$ és $d_{k,j+1} > 0$ akkor
 $d_{k,j} := d_{k,j+1}$
 $v_{k,j} := v_{k,j+1}$
elágazás vége
ha $d_{k,j} > 0$ és $d_{k,j-1} < 0$ akkor
 $d_{k,j-1} := d_{k,j}$
 $v_{k,j-1} := v_{k,j}$
elágazás vége
ha $d_{k,j} > 0$ és $d_{k,j+1} < 0$ akkor
 $d_{k,j} := d_{k,j+1}$
 $v_{k,j} := v_{k,j+1}$
elágazás vége
ciklus vége
kimenet: $E_{k,j}(u,v)$ zéruspont
ciklus vége
ciklus vége

9. ábra. A távolságfüggvény zérus helyét pontosító algoritmus

A pontosítások eredményeképp előálló $E_{k,j}(u,v)$ zérushelyeket (10.a ábra) az $S(u,v)$ felület képletébe helyettesítve a C_k fényvonalhoz tartozó $S_{k,j}(x,y,z)$ pontokat kapok a felületen (10.b ábra).



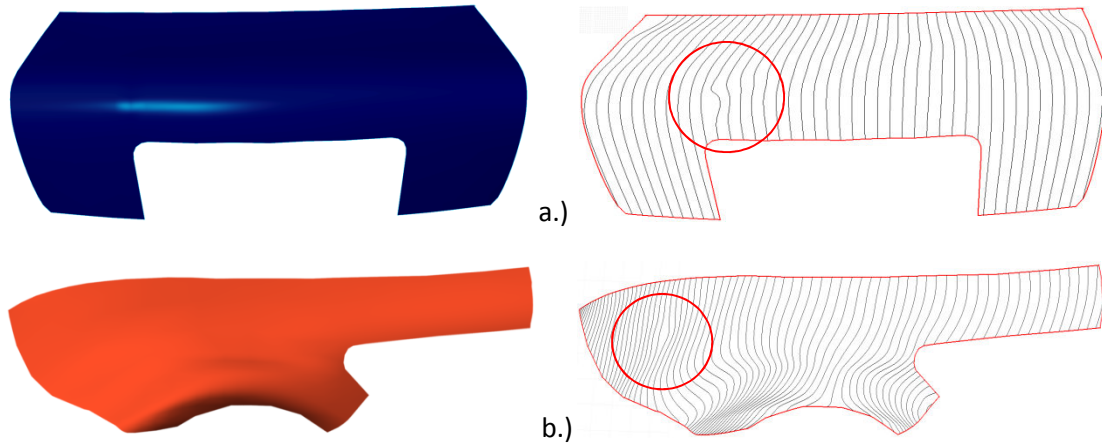
10. ábra. A távolságfüggvény zérus helyének (a./) illetve a fényvonal-pontok számítása (b./)

A folytonos reprezentációhoz az $S_{k,j}(x,y,z)$ pontokra adott pontossággal közelítő köbös B-Spline görbét illeszttek [Piegl & Tiller, 1997].

5.3. A fényvonal számító módszer alkalmazhatósága

A módszer alkalmazhatóságát számos felületen megvizsgáltam. A felületek funkciójukban, bonyolultságukban, és a hiba kiterjedésének nagyságában és összetettségében is jelentősen eltérnek egymástól. A 15.1 fejezett ábráin hibátlan felületek fényvonalai láthatók. A 68. ábrán közepes bonyolultságú,

a 69. ábrán komplex, a 70. ábrán változó mintázatú fényvonal struktúrát láthatunk. A módszer felülethibák feltárásában történő alkalmazhatóságát 15.3. fejezet ábrái szemléltetik. A 11. ábra két különböző méretű hibát tartalmazó felület példáját mutatja. A hibák helyét piros körrel jelöltem. Mindkét felület finom, az árnyékolt felületreprezentáció érzékenységi küszöbe alatti hibát tartalmaz. A nagyobbik hiba megközelíti ezt a határt (11.a), a kisebbik sokkal alatta van (11.b).



11. ábra. Példák fényvonal számító módszer alkalmazására (a.) csomagtartó b.) sárvédő)

5.4. Összefoglalás

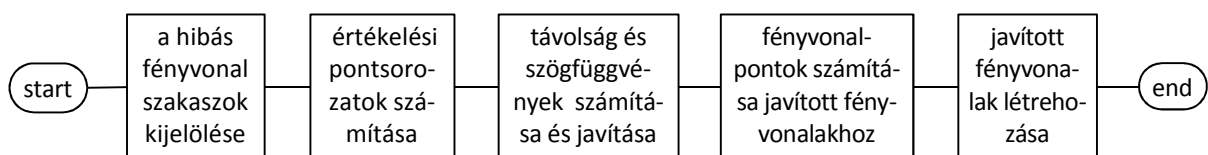
A fényvonalakat nagy pontossággal számító eljárást ismertettem. Az eljárásban a fényvonal-pontokat a felületi normálisok és a fényforrás-egyenesek között definiált távolságfüggvények zérushelyeiből számolom. A számítás egy közelítő és egy pontosító lépésből áll.

A közelítő lépésben a felületet az u és v paramétereinek mentén felosztom és a felosztásnak megfelelően felületi pontokat és a pontokhoz tartozó felületi normálisokat számolok. Ezt követően a felületi normálisokat és fényforrás-egyeneseket felhasználva, a fényvonalak összefüggéséből diszkrét értékekből álló távolságfüggvényeket számolok. A függvények zérushelyei csak közelítőleg határozhatók meg, mivel a függvények diszkrét értékekből állnak. A pontosításhoz behatárolom a zérushely két szomszédos függvényértékét, majd a felhasználó által előírt értékkel pontosítok. A következő lépésben a zérushelyekből fényvonal-pontokat számolok, majd a fényvonalak folytonos reprezentációjához a pontokra a felhasználó által előírt pontossággal görbét illesztetek.

A módszer alkalmazásához csak a felület CAD reprezentációját, a fényforrások helyzetét és irányát kell megadni. A módszer robusztus, gyors és bármilyen, CAD rendszerekben használatos matematikai felületreprezentáció esetén alkalmazható. A módszer kis, árnyékolási módszerrel nem felfedezhető felületi hibák esetén is megbízhatóan működik, alkalmazhatóságát számos ipari példafelületen vizsgáltam és igazoltam.

6. Módszer fényvonalak javítására

A felületjavítás következő lépése a hibás fényvonalak javítása. Az általam kidolgozott módszerben először az egymást követő hibás fényvonalak mintázata és alakja alkotta fényvonal-struktúrát értékelem majd javítom. A módszer működését a 12. ábra szemlélteti.

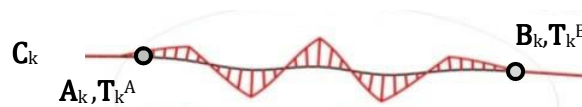


12. ábra. Fényvonalakat javító módszer blokk diagramja

Az eljárásban először kijelölöm a hibás fényvonal-szakaszok végpontjait, és a végpontokban meghatározom a fényvonalak érintőit. Ezután pontsorozatokat hozok létre az egymást követő hibás fényvonal-szakaszokon, majd a pontsorozat pontjaiból diszkrét távolság és szögfüggvényeket (hibafüggvények) számolok. A függvények a fényvonal struktúra hibáit reprezentálják. A következő lépésben a függvényeket kisímitom, és az új függvényértékekből a javított fényvonalakhoz pontokat számolok. Ezt követően a hibás fényvonal-szakaszokat a végpontjaik között kimetszem, majd a végpontokat, a végpontok érintőit és az új pontokat felhasználva helyükre javított fényvonal szakaszokat illeszték. A javított fényvonal-szakaszokat feltételes approximációt alkalmazva hozom létre. A javítás utolsó lépéseként a szomszédos jó és javított fényvonal-szakaszokat egy görbéné illesztem össze. A következő fejezetekben az eljárást ismertetem részletesen.

6.1. Hibás szakaszok végpontjainak kijelölése

A hibás fényvonal-szakaszok végpontjainak kijelölése egy előzetes, és egy pontosító műveletből áll. Az előzetes kijelölésben a felhasználó a CAD rendszer szolgáltatásait igénybe véve körberajzolja (pl. felületen történő görberajzolással) a hibás fényvonal-szakaszokat tartalmazó felületrészt (13. ábrán látható példán halványszürke görbével jelölve). Ez a kijelölés pontatlan, ami miatt jó fényvonal szakaszok is bekerülhetnek, illetve hibásak maradhatnak ki a javításból. A végpontok pontosításához a fényvonal és a rajzgörbe metszéspontjai közötti részen görbületi fésűvel grafikusán megjelenítem a görbületeszlást (a 13. ábrán piros vonalakkal jelölve). A bemutatott példán jól megfigyelhető hogy a jó és a hibás fényvonal-szakaszok görbületi eloszlása lényegesen eltér egymástól. Amíg a jó fényvonal-szakaszokon a görbületi fésű szinte belesimul a fényvonalba, a hibás szakaszokon jelentős ingadozás látható. A felhasználó a görbületi eltéréseket figyelembe véve pontosíthatja illetve azonosíthatja a végpontok helyét (13. ábrán körökkel jelölve).

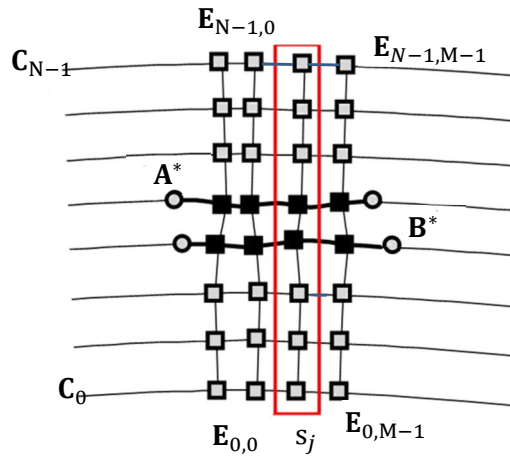


13. ábra. Hibás felületrész fényvonal szakaszainak előzetes kijelölése és a hibás fényvonal szakaszok végpontjainak meghatározása görbületi fésűk segítségével

A műveletben azonosítom C_k javításban résztvevő fényvonalat, a hibás fényvonal szakasz A_k kezdőpontját, a szakasz B_k végpontját, illetve a fényvonal ezen pontokban értelmezett T_k^A és T_k^B érintőit (13. ábra).

6.2. Fényvonal struktúra hibáinak értékelése

A fényvonal struktúra hibája az egymást követő fényvonalak mintázatából és az egyes fényvonalak alakhibáiból adódik össze, és a fényvonalak közötti rendszertelen távolság illetve irányváltozásban nyilvánul meg. A struktúra hibáját távolság és szögfüggvényekkel (hibafüggvények) számszerűsítem. A függvények létrehozásához először értékelési pontokból álló pontsorozatokat számolok. A 14. ábrán látható példán a hibás fényvonal szakaszokat vastaggal kiemelt görbék mutatják. Az s_j , $j=0..M-1$ pontsorozatok az A^* és B^* szélső végpontok között helyezkednek el, és lefedik a hibás fényvonal szakaszokat. A pontsorozatok $E_{3,j}..E_{N-4,j}$ pontjai (tömör négyzetekkel jelölve) a hibás fényvonal-szakaszok pontjai, a hibafüggvényeket ezeket a pontokat felhasználva számolom. A pontsorozatok $E_{0,j}..E_{2,j}$ és $E_{N-3,j}..E_{N-1,j}$ pontjai (üres négyzetekkel jelölve) a hibás fényvonalakkal szomszédos jó $C_0..C_2$ és $C_{N-3}..C_{N-1}$ fényvonalak pontjai, amelyeket majd a javítási műveletben a javított fényvonalak szomszédos, jó fényvonal-struktúrához történő illesztésében alkalmazok.



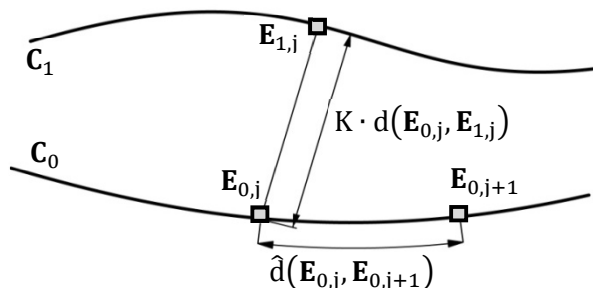
14. ábra. Értékelési pontsorozatok értelmezése

A pontsorozatok a C_0 jó fényvonal $E_{0,0} \dots E_{0,M-1}$ pontjaiból indulnak ki. A kezdőpontokat a C_0 görbületét illetve a C_0 és C_1 fényvonalak közötti távolságot figyelembe véve a következő összefüggéssel számolom:

$$\hat{d}(E_{0,j}, E_{0,j+1}) = K \cdot d(E_{0,j}, E_{1,j})$$

Ahol $E_{1,j}$ a következő fejezetrészben ismertetett módszerrel meghatározott értékelési pont. K -arányossági tényező, amellyel a pontsorozatok sűrűsége a felhasználói igényekhez igazítható. A K tényező csökkentésével sűrűsíthető, növelésével ritkítható a kezdőpontok közötti távolság, alapértelmezett értéke $K = 1$.

A kezdőpontok számítását a 15. ábra szemlélteti.



15. ábra. Az értékelési pontsorozatok kezdőpontjainak meghatározása

6.2.1. Pontsorozat számító módszerek

A pontsorozatok következő, $E_{k,j}$, $k=1 \dots N-1$ pontjait számító módszert több lehetséges változat közül, a következő szempontokat figyelembe véve választottam ki:

- a pontok által kijelölt nyomvonal alakja
- a pontokból számolt hibafüggvények alakja
- a simított hibafüggvények alakja
- a pontszámító eljárás költsége

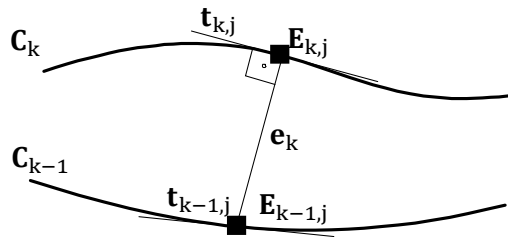
Nyomvonalakon az s_j , $j=0 \dots M-1$ pontsorozatok $E_{k,j}$, $k=0 \dots N-1$ értékelési pontjainak összekötésével előálló egyenes szakasz sorozatokat értem. Alakjuk jól használható a fényvonal struktúra vizuális értékeléséhez. A hibafüggvények definícióját a 6.2.2 fejezetben ismertetem részletesen. A leginkább alkalmas módszertől azt várom, hogy nyomvonalai követik a hibás fényvonal-struktúra hibáját, olcsón számítható, hibafüggvényei érzékenyen kimutatják a hibákat. A módszereket több fejezetben is érté-

kelem, a leginkább megfelelőt a 6.3.2 fejezet végén választom ki. Az értékelésekhez számos bonyolultságában és funkciójában jelentősen eltérő ipari felületet használtam fel. Az értékelések során kapott pontsorozatokat a 15.2. melléklet ábrái szemléltetik.

A módszer változatok mindegyike a sorozat \mathbf{E}_k pontjait a \mathbf{C}_{k-1} előző fényvonalon meghatározott \mathbf{E}_{k-1} pontból kiindulva számítja.

1. Módszer

Az \mathbf{E}_k pontot a \mathbf{C}_k és \mathbf{C}_{k-1} fényvonalak közötti legrövidebb merőleges távolságból számolom (16. ábra).



16. ábra. Értékelési pont számítása az 1. módszerrel

Legyen $\mathbf{E}_{k-1,j}$ a \mathbf{C}_{k-1} fényvonal értékelési pontja és $\mathbf{t}_{k-1,j}$ a \mathbf{C}_{k-1} fényvonal $\mathbf{E}_{k-1,j}$ pontjának érintője. Az $\mathbf{E}_{k,j}$ pont a \mathbf{C}_k fényvonal értékelési pontja, ha teljesül:

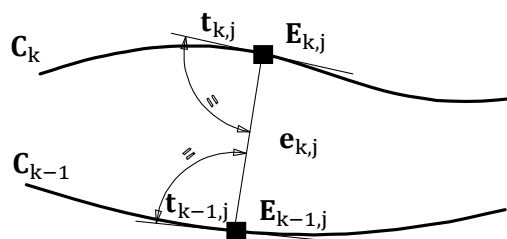
$$\min(\angle \mathbf{t}_{k-1,j} \mathbf{e}_{k,j} - \angle \mathbf{t}_{k,j} \mathbf{e}_{k,j})$$

azzal a feltétellel, hogy: $\mathbf{t}_{k,j} \cdot \mathbf{e}_{k,j} = 0$

Ahol $\mathbf{e}_{k,j} = \mathbf{E}_{k,j} - \mathbf{E}_{k-1,j}$

2. Módszer

Az $\mathbf{E}_{k,j}$ pontot a \mathbf{C}_k és \mathbf{C}_{k-1} szomszédos fényvonalak közti olyan legrövidebb távolságból számolom, amelynél teljesül, hogy az $\mathbf{E}_{k-1,j}$ és $\mathbf{E}_{k,j}$ pontokban értelmezett $\mathbf{t}_{k-1,j}$ és $\mathbf{t}_{k,j}$ érintők és a pontokat összekötő $\mathbf{e}_{k,j}$ szakasz közötti egyállású szögek egyenlők (17. ábra).



17. ábra. Értékelési pont számítása a 2. módszerrel

Az $\mathbf{E}_{k,j}$ pont a \mathbf{C}_k fényvonal értékelési pontja, ha teljesül:

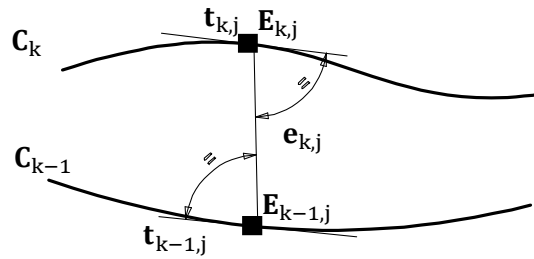
$$\min(\|\mathbf{E}_{k,j}\| - \|\mathbf{E}_{k-1,j}\|)$$

úgy hogy:

$$\angle \mathbf{t}_{k-1,j} \mathbf{e}_{k,j} - \angle \mathbf{e}_{k,j} \mathbf{t}_{k,j} = 0$$

3. Módszer

Az előző, 2. módszertől abban különbözik, hogy az egyállású szögek helyett a váltószögek szerepelnek (18. ábra).



18. ábra. Értékelési pont számítása a 3. módszerrel

Az $E_{k,j}$ pont a C_k fényvonal értékelési pontja, ha teljesül a következő:

$$\min(\|E_{k,j}\| - \|E_{k-1,j}\|)$$

azzal a feltétellel, hogy:

$$\angle t_{k-1,j} e_{k,j} - \angle t_{k,j} e_{k,j} = 0$$

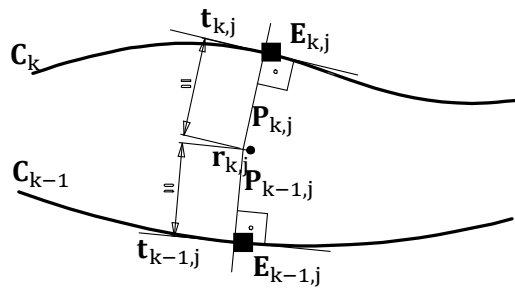
4. Módszer

Legyen $P_{k-1,j}$ az $E_{k-1,j}$ pont és a $t_{k-1,j}$ vektor és $P_{k,j}$ az $E_{k,j}$ pont és $t_{k,j}$ vektor definiálta normálisok, és legyen $r_{k,j}$ a normálisok metszésvonala (19. ábra). Az $E_{k,j}$ pont a C_k fényvonal egy értékelési pontja, ha teljesül:

$$\min(\|E_{k,j}\| - \|E_{k-1,j}\|)$$

azzal a feltétellel, hogy:

$$d(r_{k,j}, E_{k-1,j}) - d(r_{k,j}, E_{k,j}) = 0$$



19. ábra. Értékelési pont számítása a 4. módszerrel

A módszerek hibafüggvények szerinti értékeléséhez szükségünk van a hibafüggvényekre, a következő fejezetben a függvények számítási módszerét ismertetem.

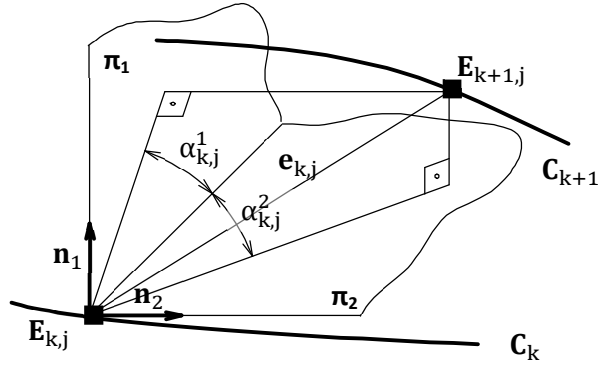
6.2.2. Távolság és szögfüggvények számítása

A távolság és szögfüggvényeknek (hibafüggvények) közvetlen geometriai jelentésük van, a hiba nagyságát és alakját mutatják. A hibák a függvények alakjában hirtelen és rendszertelen változásokként jelennek meg, a jó illetve javított fényvonal-struktúra függvényei egyenletesek vagy harmonikusan változóak.

A távolságfüggvény az $E_{k+1,j}$ és $E_{k,j}$ pontok között számolt $d_{k,j}$ távolságok sorozata:

$$d_{k,j} = \|E_{k+1,j}\| - \|E_{k,j}\| \quad (31)$$

A szögfüggvényt, a javítás új pontokat számító művelete miatt, két egymásra merőleges vetületével kezeljük. A vetületek előállításához Descartes-i koordináta-rendszert alkalmazunk, a koordináta-rendszer π_1 és π_2 síkjait (20. ábra) a később ismertetett n_1 és n_2 normálvektoraikkal határozzuk meg.



20. ábra. A szögfüggvények vetületek értelmezése

Az $\alpha_{k,j}^1$ és $\alpha_{k,j}^2$ szögfüggvény értékeket az $\mathbf{e}_k = \mathbf{E}_{k+1,j} - \mathbf{E}_{k,j}$ szakaszok és az \mathbf{n}_1 és \mathbf{n}_2 koordinátatengelyek közötti szögekből számoljuk:

$$\alpha_{k,j}^1 = \arcsin \frac{\mathbf{n}_1 \cdot \mathbf{e}_{k,j}}{|\mathbf{n}_1| \cdot |\mathbf{e}_{k,j}|} \quad \alpha_{k,j}^2 = \arcsin \frac{\mathbf{n}_2 \cdot \mathbf{e}_{k,j}}{|\mathbf{n}_2| \cdot |\mathbf{e}_{k,j}|} \quad (32)$$

A vetületek alkalmazásával felmerül, hogy bizonyos esetekben a szögek olyan vetületei jönnek létre, amelyek túlságosan eltorzítják a szöghibát és akadályozzák a javítást. Ennek az elkerülésére olyan koordinátarendszert kerestünk, amely a függvény vetületeit a legkevésbé torzítja, és amelyben a vetületek lehetőleg gyorsan számíthatók. Két koordinátarendszert vizsgáltunk meg: a felület koordinátarendszerét és a fényforrások definiálta koordinátarendszert.

A felület koordinátarendszerében az \mathbf{n}_1 és \mathbf{n}_2 vektorok a következőképp számíthatók:

$$\mathbf{n}_1 = \mathbf{X}, \quad \mathbf{n}_2 = \mathbf{Y} \quad (33)$$

A vetületek számítása itt egyszerű és gyors. Egyrészt a koordinátatengelyek eleve rendelkezésre állnak, másrészt a vetületek számítása leegyszerűsíthető, ha a (32) összefüggés helyett a következőket alkalmazom:

$$\alpha_{k,j}^1(x, y, z) = \alpha_{k,j}^1(x, y, 0)$$

$$\alpha_{k,j}^2(x, y, z) = \alpha_{k,j}^2(x, 0, z)$$

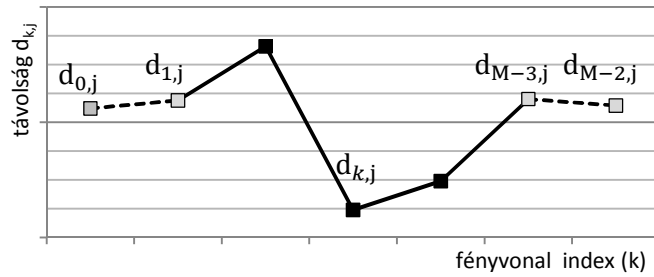
A vetületek alakja azonban ebben a koordinátarendszerben erősen függ a fényforrások helyzetétől, ezen felül, ha a \mathbf{B} vektor és a koordinátarendszer valamelyik vetületi síkja (π_1 vagy π_2) merőleges vagy közel merőleges egymásra, az egyik vetület túlságosan torz.

A fényforrások definiálta koordinátarendszerben az \mathbf{n}_1 és \mathbf{n}_2 vektorok a következőképp számíthatók:

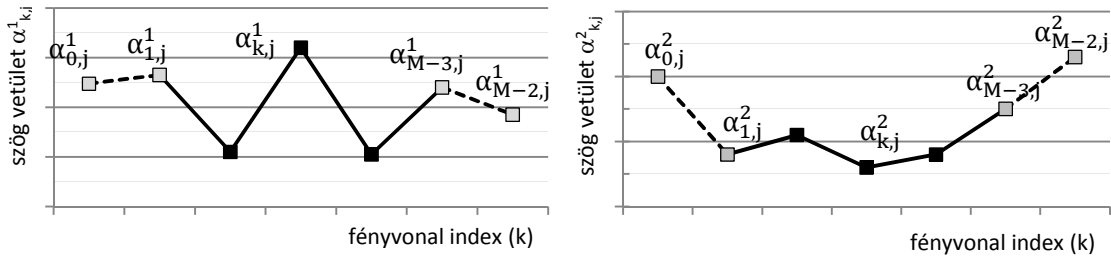
$$\mathbf{n}_1 = \mathbf{B}, \quad \mathbf{n}_2 = \mathbf{B} \times \mathbf{D} \quad (34)$$

Itt kénytelenek vagyunk a (32) összefüggést alkalmazni, és az \mathbf{n}_2 -t külön számítani kell. Ez, az előző esethez képest, időigényesebb számítást jelent, a vetületek alakja azonban csak a fényvonalak alakjától függ és jól értelmezhető geometriai jelentése van. Ebben a koordinátarendszerben az α^1 függvény alakja a hiba felületre merőleges irányú alakhibáját, az α^2 függvény pedig a "keresztmetszet vonalát" mutatja. A két koordinátarendszer tulajdonságait összevetve ez utóbbi alkalmazásánál maradtunk.

A hibás fényvonal-struktúra távolságfüggvényére illetve vetületi függvényeire mutat példát a 21. és a 22. ábra (A függvényértékeket a 14. ábra példájából számoltuk).



21. ábra. Példa a hibás fényvonal struktúra távolságfüggvényére

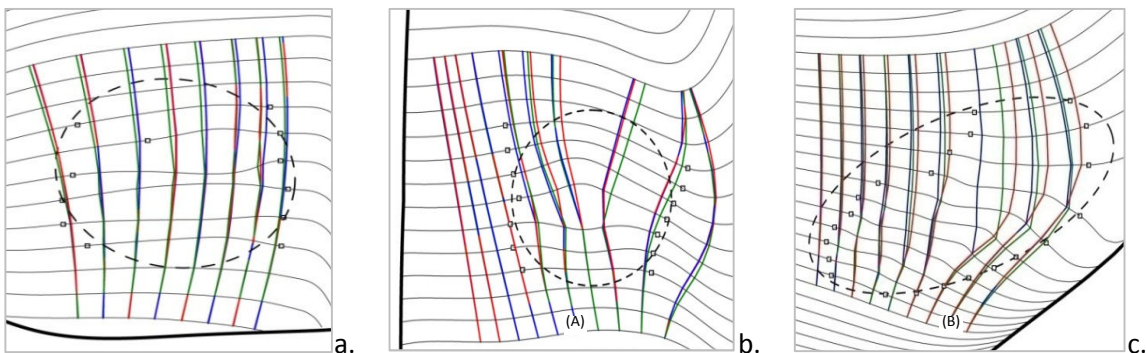


22. ábra. Példa a hibás fényvonal struktúra vetületi szögfüggvényeire

A függvények $d_{2,j} \dots d_{M-4,j}$ illetve $\alpha_{2,j}^1 \dots \alpha_{M-4,j}^1$ és $\alpha_{2,j}^2 \dots \alpha_{M-4,j}^2$ értékeit a hibás fényvonalszakaszok pontjaiból számoltuk (folytonos vonallal összekötött pontok). Ezen a szakaszon a függvényekben rendszertelen ingadozások, hirtelen és nagy változások figyelhetők meg. A függvény javítás feladata ezeket megszüntetni úgy, hogy a javított függvény egyrészt sima és harmonikus legyen, másrészt folytatólagosan illeszkedjen a szomszédos jó fényvonalakból számolt függvényértékekhez (szaggatott vonallal összekötött pontok). Az illesztési feltételek a távolságfüggvényénél a $d_{0,j}$, $d_{1,j}$ és $d_{M-3,j}$, $d_{M-2,j}$ illetve a szögfüggvényeknél az $\alpha_{0,j}^1$, $\alpha_{1,j}^1$ ill. $\alpha_{M-3,j}^1$, $\alpha_{M-2,j}^1$ és az $\alpha_{0,j}^2$, $\alpha_{1,j}^2$ ill. $\alpha_{M-3,j}^2$, $\alpha_{M-2,j}^2$ értékekből számolt érintőirányok, illetve a $d_{1,j}$, $d_{M-3,j}$ és az $\alpha_{1,j}^1$, $\alpha_{M-3,j}^1$ ill. $\alpha_{1,j}^2$, $\alpha_{M-3,j}^2$ csatlakozási pontok.

6.2.3. Pontsorozat számító módszerek nyomvonalaik szerinti értékelése

A nyomvonalaktól azt várom, hogy a hibás fényvonalszakaszokon haladnak, és hogy jól követik a hiba változásának irányát. A pontsorozat számító módszerek nyomvonalait számos különböző nagyságú, kiterjedésű és bonyolultságú hibával rendelkező felületen vizsgáltam, a vizsgálatokban kapott vonalak a melléklet 15.2. fejezetében láthatók. A 23. ábrán a vizsgálatokban kapott vonalakat értékelem, három reprezentatív példán (a., b., c.) keresztül.



23. ábra. Különböző bonyolultságú fényvonal hibák nyomvonalai

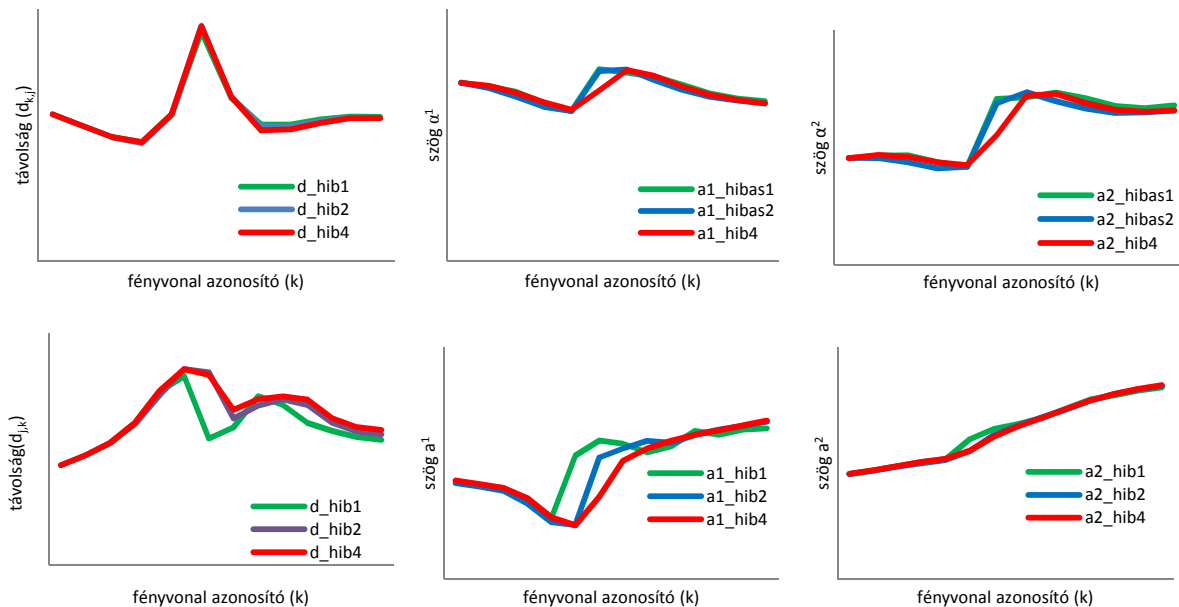
A zöld vonalak az 1. módszer, a kékek a 2. módszer illetve a pirosak a 4. módszer vonalait mutatják. A vonalak a hiba alakjához igazodva, végig a hibás fényvonalszakaszok (körökkel jelölt) végpontjai között haladnak. Kis hibák esetén mindegyik módszer jól működik, csaknem teljesen ugyanazokat a nyomvonalakat produkálja (24.a ábra). Nagyobb hibák esetén eltérés tapasztalható a módszerek vi-

selkedésében (24.b. és 24.c. ábra), vonalaik hajlamosak a hiba közepét megkerülni és a hiba peremén haladni, aminek a következménye a hibát kevésbé jól reprezentáló hibafüggvények.

Az eljárások nyomvonalait összehasonlítva az látható, hogy az esetek többségében a 2. módszer vonala volt a leginkább elfogadható. A vonalai mindegyik próbafelületen stabilak maradtak, és kevésbé kerültk ki a hiba közepét, amit a szögegyenlőség feltétel hatásának tulajdonítok.

6.2.4. Pontsorozat számító módszerek hibafüggvényeik szerinti értékelése

Az értékelésben a hibafüggvények alakját hasonlítottam össze, a függvényektől azt vártam el, hogy a hibákat érzékenyen, nagy változásként mutassák ki. Az értékelésben résztvevő pontok nyomvonalai és hibafüggvényeik a 15.2. melléklet ábráin láthatók. A vizsgálatokban kapott függvények alakját egy kisebb és egy nagyobb hiba pontsorozatain számolt függvénnyel szemléltetem. A pontsorozatok a 23. ábrán látható példa (A) és (B) nyomvonalaihoz tartoznak, a függvények a 24. ábrán láthatók. Kisebb hiba esetén (az ábra első sora) a függvények alakja mindegyik módszerben nagyon hasonló egymáshoz, maximumértékeik szinte ugyanakkorák és ugyanoda esnek. A nagyobb hibáknál (az ábra második sora) az alakbeli eltérések nagyobbak, amit elsősorban a távolság és az α^1 szögfüggvény esetén tapasztaltam. Mindent összevetve ugyanakkor elmondható, hogy a függvények érzékenysége mindkét esetben és mindegyik pontszámító módszernél nagyon hasonló.



24. ábra. Egyszerűbb és bonyolultabb fényvonal hibák távolság és szögfüggvényei

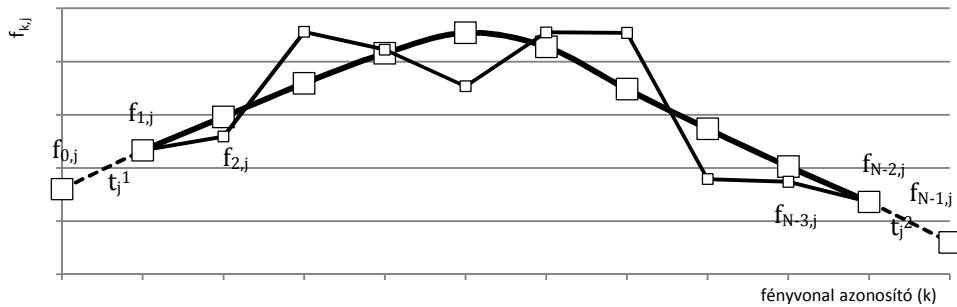
6.3. Fényvonal struktúra javítása

A fényvonal struktúra javítás következő lépésében a függvényeket kisítom, és a simított függvényekből a javított fényvonal szakaszok megszerkesztéséhez új pontsorozatokat számolok. A simításnál ügyelni kell a függvény simítottságának mértékére. A túlzott simítás következménye, hogy a javítás olyan geometriai elemeket mos el a felületen, amit a felhasználó megtartana, illetve a felület túl távolra kerül az eredeti a felülettől. A túlságosan mérsékelt simítás ugyanakkor elégtelen javítást eredményezhet.

Ezt a problémát úgy oldottam meg, hogy a simítási műveletben kiindulásként kísérletekkel meghatározott, alapértelmezett simítási paraméter értékeket alkalmaztam, amelyeket a felhasználó igényei szerint változtathat ezzel befolyásolva a simítás mértékét.

6.3.1. A távolság és szögfüggvények simítása

A távolság és szögfüggvények simításhoz (itt egységesen f -fel jelöltem őket) a jó fényvonal-struktúra kezdő és végpontjait reprezentáló $f_{1,j}$ és $f_{N-2,j}$ pontok között, approximációs B-Spline görbével helyettesítem (25. ábra, vastag görbe). Az approximáció feltételei az $f_{1,j}$ és $f_{N-2,j}$ függvényértékek, a skálázható t_j^1 és t_j^2 érintők, a görbe fokszáma, illetve a görbe kontrollpontjainak maximális száma. Az approximációs algoritmus olyan görbét hoz létre, amelyik az $f_{1,j}$ és $f_{N-2,j}$ pontokhoz G1 folytonossággal csatlakozik, és a görbe és a simítatlan függvény $f_{2,j} \dots f_{N-3,j}$ pontjai közötti távolságok négyzetének összege minimális. Az új, simított függvény pontjait az azonos ordinátákból határozom meg.

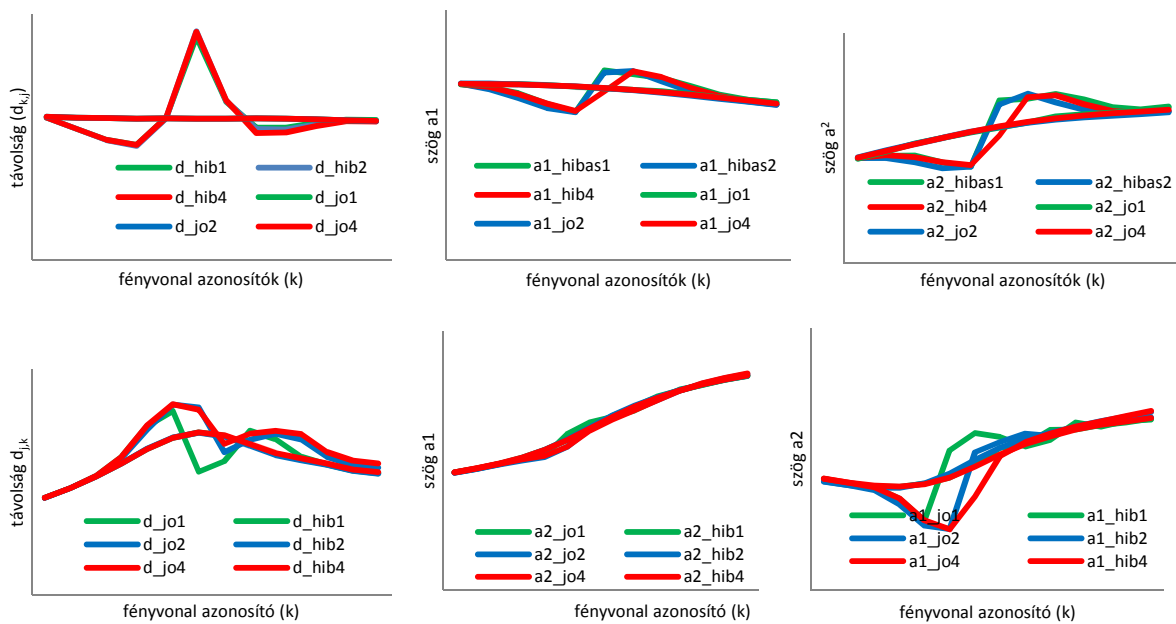


25. ábra. Példa hibafüggvény javítására

A simítási paraméterek, amelyeket a felhasználó változtathat a kontrollpontok száma és a görbe fokszáma. A kísérletekkel meghatározott alapértelmezett fokszám 4, a kontrollpontok száma 5. A viszonylag magas fokszám és kontrollpont szám oka az, hogy a simított függvények formája sokszor elnyújtott, összetett, és inflexiós pontot is tartalmazhat.

6.3.2. Pontsorozat számító eljárások értékelése simítófüggvényeikkel

A pontsorozat számító módszerek simítófüggvényeik szerinti értékeléséhez a pontsorozatok hibafüggvényeit és simítófüggvényeit hasonlítottam össze. A fenti példák (24. ábra) hiba- és simítófüggvényei a 26. ábrán egyszerre láthatók.



26. ábra. Egyszerűbb és bonyolultabb távolság és szögfüggvények és simított változataik

A példa javítófüggvényeihez az alapértelmezett paramétereket alkalmaztam. Látható hogy alakjuk a hibafüggvények különbözősége ellenére nagyon hasonló, ami a simítási eljárás robusztusságát mutat-

ja, illetve azt, hogy alakjuk elsősorban a simítás paramétereitől függ, nem a pontsorozat számítási módszertől.

A pontszámító módszerek vizsgálati eredményeit összegezve arra a következtetésre jutottam, hogy a fényvonalak javításához a leginkább a 2. módszer alkalmazható.

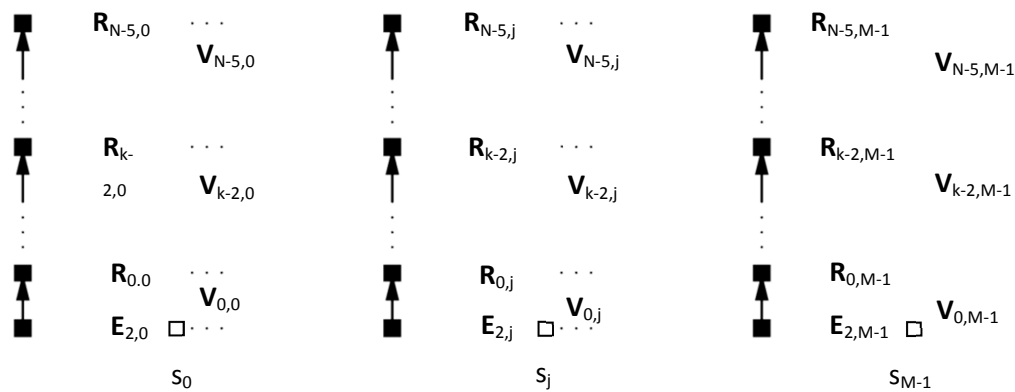
6.3.3. Pontok számítása a javított fényvonalakhoz

A javított fényvonalakhoz a pontokat a simított szög és távolságfüggvényekből számolt vektorok összegzésével számítom. Legyen $\mathbf{R}_{k,j}$ egy a javított fényvonalakhoz számolt pont. A pontot a $d_{k,j}$ és $\alpha_{k,j}^1$ ill. $\alpha_{k,j}^2$ függvényértékekből számolt $\mathbf{V}_{k,j}$ vektorok összegzésével állítom elő:

$$\mathbf{R}_{k,j} = \mathbf{E}_{2,j} + \sum_k \mathbf{V}_{k,j} \quad j = 0 \dots M, k = 2 \dots N \quad (35)$$

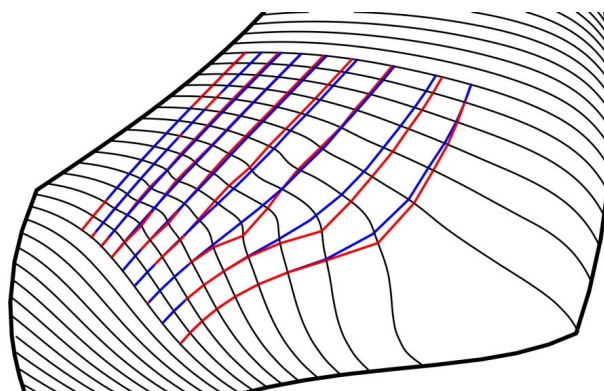
ahol $V_{k,j}(x) = d_{k,j} \cdot \cos(\alpha_{k,j}^1) \cdot \cos(\alpha_{k,j}^2)$, $V_{k,j}(y) = d_{k,j} \cdot \cos(\alpha_{k,j}^1) \cdot \sin(\alpha_{k,j}^2)$, $V_{k,j}(z) = d_{k,j} \cdot \sin(\alpha_{k,j}^1)$

A vektorok összegzése az s_j pontsorozatok $\mathbf{E}_{2,j}$ jó fényvonalon számolt pontjánál kezdődik, és az $\mathbf{R}_{N-3,j}$ javított pontig tart. Az összegzést a 27. ábra szemlélteti. A jó fényvonalakon lévő pontokat üres, a javított pontokat tömör négyzetek jelölik.



27. ábra. Pontok számítása a javított fényvonalakhoz

A 28. ábra az összegzés eredményeként előállt új pontok nyomvonalaira mutat példát (23.c. ábrán látható példafelület).



28. ábra. Példa a javított pontsorozatok nyomvonalaira

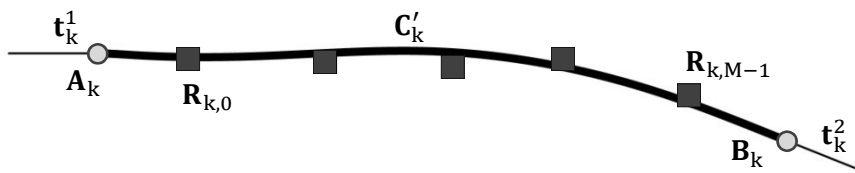
A javított függvényekből előállított új pontok vonalai (kék) mindenhol közel maradtak az eredeti pontok vonalaihoz (piros) és egy lehetséges jó fényvonal struktúra nyomvonalait mutatják. Az ábrán az is megfigyelhető, hogy minél kisebb a hiba, annál kisebb a hibás és jó pontok nyomvonalai között a különbség, és hogy a jó fényvonal-struktúrában (ábra első két pontsorozat vonala) a kettő ugyanazzá válik. Ez a módszer azon sajátosságát szemlélteti, hogy a jó fényvonal struktúrát nem módosítja. Ez egyrészt a javítás helyességét, másrészt a módszer robusztusságát igazolja.

6.3.4. Javított fényvonalak létrehozása

A javított fényvonalakat az új pontokra illesztett B-Spline görbéként állítom elő. Az illesztéshez feltételes lokális approximációs algoritmust alkalmazok [Piegl & Tiller, 1997], a feltételek a hibás fényvonal szakaszok végpontjai, és a végpontokban értelmezett érintőik iránya.

A görbe simasága illetve pontossága közti ellentétet a függvények simításánál ismertetett módszerhez hasonló módon, alapértelmezett értékek alkalmazásával oldom fel. A szabad paraméterek itt a görbe fokszáma és az illesztés pontossága. Az görbe alapértelmezett fokszáma 3. A görbe többi paramétere állandó: a csomó vektora egyenközű, a pontokhoz rendelt paraméterek az $\mathbf{R}_{k,j}$ pontok meghatározta húrhosszal arányosak.

A 29. ábra egy ilyen approximációval előállított fényvonalat mutat. A javított \mathbf{C}'_k fényvonalszakaszt telt vastag vonal, \mathbf{A}_k kezdő és \mathbf{B}_k végpontjait üres négyzetek jelölik. A végpontok \mathbf{t}_k^1 és \mathbf{t}_k^2 érintőinek irányát vékony vonalak mutatják.



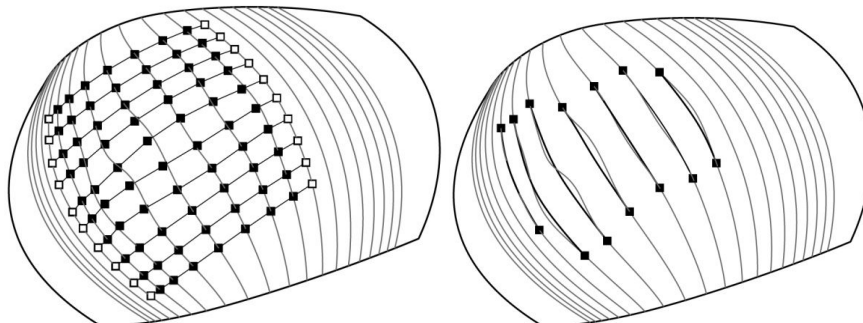
29. ábra. Az új pontokra illesztett fényvonal görbe

Az illesztés után a javított fényvonalak három külön görbéből állnak. A későbbi felületjavítási művelethez a görbék egy görbévé illeszttem össze. Ehhez először a javított szakasz $u \in [0, 1]$, paramétertartományát a kijelöléshez tartozó pontok $u_k \in [u_{A_k}, u_{B_k}]$ paraméter-értékeinek megfelelően átszámolnom (11), majd az eredeti, hibás görbe egyenletébe az új csomóvektort és kontrollpontokat beillesztve a három görbét eggyel helyettesítem. A módszert a „Rhino 5” NURBS modellező programhoz (©Robert McNeel & Associates, 2015) illesztettem.

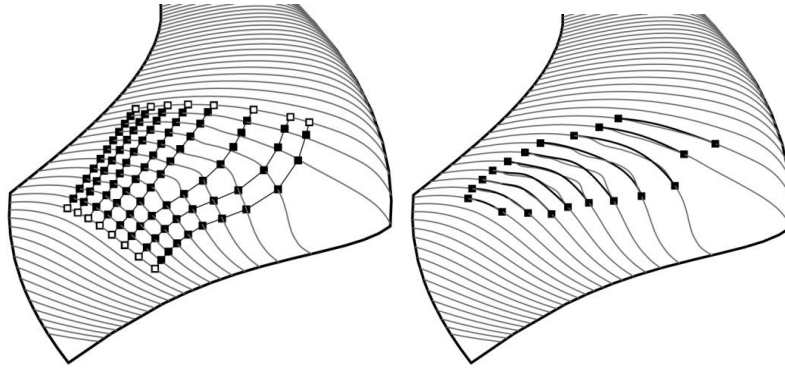
6.4. Alkalmazási példák

A módszer alkalmazhatóságát bonyolultságban és méretben egymástól jelentősen eltérő felületeken vizsgáltam és igazoltam. A 15.2 melléklet ábrái a felületek javítás előtti illetve javítás utáni fényvonalait mutatják. A javított fényvonalszakaszokat vastag, a hibásakat vékony vonalakkal jelöltem. Az ábrák tömör négyzetei az új pontokat, az üresek a jó fényvonalak pontjait jelölik.

A 30. és 31. ábra két különböző bonyolultságú fényvonal struktúra javítását mutatja. Az első felületen a hiba a fényvonalak alakjában látható, második felületen a fényvonalak mintázatában és alakjába is található hiba.



30. ábra. Példa fényvonal struktúra javítására



31. ábra. Példa fényvonal struktúra javítására

6.5. Összefoglalás

Fényvonalak javítására alkalmas módszert ismertettem. A módszerben a javítás két egymást követő művelet eredménye. Az első műveletben kijelölöm a hibás fényvonalszakaszok végpontjait, és az egymást követő fényvonalszakaszokon pontsorozatokat definiálok. A pontsorozatokon távolság és szögfüggvényeket hozok létre, amelyekkel a fényvonal struktúra hibái értékelhetők és javíthatók. A függvényeket a következő lépésben kisimítom, és a simított függvényértékekből a javított fényvonalkhoz új pontokat számítok. A javított fényvonalat a hibásakat helyettesítve, az új pontokra illesztett görbékként hozom létre. A módszer alkalmazhatóságát számos hiba nagyságban és kiterjedésben eltérő ipari felületen értékeltem és bizonyítottam. A módszert sikeresen beillesztettem a „Rhino 5” NURBS modellező programba (©Robert McNeel & Associates, 2015). Az approximációs feladatok kivitelezéséhez a „Solid Modeling Solutions ©” rendelkezésemre bocsájtott C++ rutinait alkalmaztam.

7. Módszer a felület javított fényvonalakhoz történő igazítására

A direkt felületjavítás a felületértékelési vonalakat kijavítja, majd megkeresi a javított vonalakat produkáló felületet. A felület előállításában a problémát a javított értékelési vonalak és a felület paramétereit közti bonyolult, nemlineáris összefüggés jelenti, ami a fényvonalakra is igaz. Az általam kidolgozott módszerben a problémát genetikussal kezeltem, amely közvetlen matematikai összefüggések nélkül is képes megoldást találni.

A genetikussal algoritmus paramétereit és operátorait többlépéses és több szempontot figyelembe vevő vizsgálatok és kísérletek alapján határoztam meg. Ebben a folyamatban először meghatároztam a felület genetikussal reprezentációját, és kiválasztottam a gének kódolási típusát. Ezt követően változatokat dolgoztam ki a fitnessz számítására, és a szakirodalom alapján kijelöltem a probléma megoldásában alkalmazható többi genetikussal paraméter számítási módját és az operátorok típusait. Mindezek alapján összeállítottam egy elsődleges működő algoritmust, majd az algoritmust sebességi és megbízhatósági szempontok szerint finomhangoltam. Az finomhangoláshoz a paramétereket és az operátorokat szisztematikusan változtattam, és az így előálló GA-k működését értékeltem. Az értékeléshez saját, a feladathoz illeszkedő mérőszámokat alkalmaztam.

7.1. A felületek genetikussal reprezentációja

A felület genetikussal reprezentációjában, a felület alakját meghatározó paraméterek közül, kizárólag a felület kontrollpontjait alkalmazom. A kontrollpontok a felület azon paramétereit, amelyek a felület alakját, és ezen keresztül a fényvonalak alakját is a leginkább meghatározzák.

A kontrollpontok egyedüli alkalmazása a génreprezentációban, egyszerű, könnyen kezelhető géneket eredményez, alkalmazásukkal a felület alakja közvetlenül és hatékonyan befolyásolható. Több paraméter (pl. kontrollpontok és a súlyaik) egyszerre történő változtatása a keresési tér dimenzióinak növekedése miatt, költségesebbé tenné a javítást.

A felületet a genetikus keresésben a c_κ kromoszóma reprezentálja, amely a g_λ génekből áll:

$$c_\kappa = (g_0, \dots, g_\lambda, \dots, g_{\Lambda-1}) \quad (36)$$

ahol

$g_\lambda = P_{i,j}(x, y, z)$	kromoszóma génjei (a felületjavításban érintett kontrollpontok)
λ	gén azonosítója a kromoszómában
Λ	gének (kontrollpontok) száma
κ	kromoszóma azonosítója a populációban

A lokális felületreprezentációknál, mint a B-Spline vagy a NURBS, a kontrollpontok hatása csak meghatározott felületrészre (felületfoltra) terjed ki (2.2 fejezet). A gének azokat a kontrollpontokat tartalmazzák, amelyekhez tartozó felületfoltok hibás fényvonalszakaszokat tartalmaznak.

7.2. A genetikai kódolás kiválasztása

A kódolás nagyban meghatározza az algoritmus alkalmazhatóságát, az algoritmus operátorait és paramétereit. A genetikus kódolást két fajta lehet: bináris (BCGA) vagy valós (RCGA). A következőkben a folytonos változójú problémákhoz történő alkalmazhatóságukat vizsgálom meg.

7.2.1. Bináris kódolás

A bináris kódolás egyik alapvető kérdése a változói intervallum diszkrétizálása. Legyen $[a_i, b_i]$ folyamatos intervallum, amit egy változó kódolására alkalmaznak. A kódolás során az $[a_i, b_i]$ intervallumból a $\{0, \dots, 2^{L_i}\}$ halmazba történő transzformációt hajtják végre (L_i a bitek száma a kódolásban) ami magába foglalja az $[a_i, b_i]$ intervallum ρ pontossággal történő diszkrétizálását:

$$\rho = \frac{b_i - a_i}{2^{L_i} - 1} \quad (37)$$

Az L_i paraméter meghatározza a keresési tér nagyságát, ugyanakkor behatárolja a megoldás pontosságát is (ρ az L_i -től függ). Ez problémákhoz vezethet, ha széles intervallumban nagy pontossággal kell az eredményt meghatározni. A [Janikow & Michalewicz, 1991] cikk ezt a következő példával szemlélteti: 100 változó esetén, amelyek a $[-500, 500]$ tartományban változik, és ahol hat tizedes pontosság szükséges, a bináris megoldás vektora 3000 ami 10^{1000} méretű keresési teret generál. A BCGA teljesítménye ezekben és ehhez hasonló esetekben gyenge lehet.

Ezen felül, a keresés kezdeti szakaszában a BCGA sok számítási erőforrást pazarol a kevésbé fontos tizedes jegyek értékelésére. Az optimum pontossága a legfontosabb tizedes jegyeken múlik, amíg ezek nem konvergálnak, a magasabb rendű tizedes jegyek javítás felesleges. Amikor a legértékesebb tizedes helyek konvergenciáját az algoritmus eléri, nem szükséges több erőbefektetés a finomításukra. Ezt az ideális viselkedést a BCGA nem tudja megvalósítani, mivel minden számjegyet hasonlóan kezel. Gyakran a kevésbé fontos számjegyek a keresés kezdetén egy véletlen szám felé konvergálnak, ami rontja a keresés hatékonyságát, és az elért pontosság sem lesz megfelelő.

Ha a binárisan kódolt paraméter értékészlete a kettes számrendszerből eltérő alapú, némely kódok feleslegesek lehetnek, mert olyan információt kódolnak, amik nem tartoznak a paraméter értéktartományába. Vegyük például azt a paramétert, amely 0.000 és 0.500 között változik. A paraméter kódolásához legalább 9 bit szükséges, így a 111110101 és a 111111111 közé eső elemek nem létező értékeket kódolnak. A felesleges kódok meglepő problémát jelent, mivel nem lehet garantálni, hogy a keresztezés vagy a mutáció olyan kódokat generálnak, amik nem feleslegesek. Az ilyen kódok kiküszöbölésére különböző mechanizmusok léteznek, amelyek viszont fontos genetikus információk elvesztését okozhatják.

7.2.2. Valós kódolás

A valós kódolás egyik legfontosabb sajátossága a kromoszómák valós számmal történő reprezentálása. A kromoszóma valós számokból álló vektor, hossza megegyezik a megoldásvektor hosszával, így minden gén egy változót reprezentál. A gének értékét a genetikus operátorok tartják a reprezentált

változó által meghatározott intervallumban. Lehetővé teszik, hogy a változók széles intervallumban változzanak, aminek a megvalósítása a bináris kódolásban állandó kromoszómahossz mellett csak az elérhető precizitás kárára volna lehetséges.

A valós paraméterek alkalmazásának további előnye a folytonos változójú függvények fokozatosságának kihasználási képessége. A fokozatosság itt azt jelenti, hogy a változó kis módosításához kis függvényváltozás tartozik. Ez a képesség az RCGA vonatkozásában a megoldások lokális finomítási képességét jelenti. A genetikus operátorok a finomítást egy alkalmazhatóbb és gyorsabb formában végzik mint a BCGA, ahol a finomítás a *Hamming* part miatt lehet nehézkes. A Hamming part az a jelenség, amikor a folytonos változó két egymást követő értékének reprezentációja mindegyik bit értékében különbözik (pl. 31=11111, 32=100000). Ennek az a következménye, hogy a *Hamming* távolság a két szomszédos érték között egynél nagyobb lesz, indokolatlanul eltávolítva egymástól a két értéket [Vancsay, 2001]. A valós kódolás további előnye hogy nincs szükség a bináris és a valós reprezentációk közti konverzióra, amivel az algoritmus időt takarít meg. A folytonos változójú problémák esetén, mint a kontrollpontok változtatása, a kódolások közül az RCGA jobban alkalmazható.

7.3. Keresési tér

A genetikus keresés során az új kromoszómaváltozatok a gének módosításával jönnek létre. A gének módosítása a mi esetünkben a génekben lévő kontrollpontok koordinátáinak módosítását jelenti. A módosítás értéktartománya génenként változik és a génhez tartozó keresési alteret határozza meg. A GA keresési tere ezen alterek halmaza.

A gének keresési alterei olyan folytonos geometriai térrészek, amelyekben a gének kontrollpontjai a genetikus keresés során elhelyezkedhetnek. A térrészek gömb alakúak, helyzetüket a gömb középpontjával, méretüket a gömb sugarával definiáltam.

A térrészek helyzete megegyezik a hibás felület kontrollpontjainak helyzetével és a keresés során nem változik. A térrészek ϵ_λ méretének meghatározásakor figyelembe vettem a kontrollponthoz tartozó felületfolt MAX.H legnagyobb fényvonal hibáját, és a kontrollpont befolyását a hibás felületrész alakjára. Ez utóbbit a b_λ aránytényezővel számszerűsítettem, a tényező értéke a felületfolton lévő fényvonal-pontok és az összes javításban érintett fényvonal-pont számának hányadosa. A térrészek méretét az eredményes genetikus kereséshez a χ szorzótényezővel igazítottam.

A térrészek méretét a következő összefüggéssel számolom:

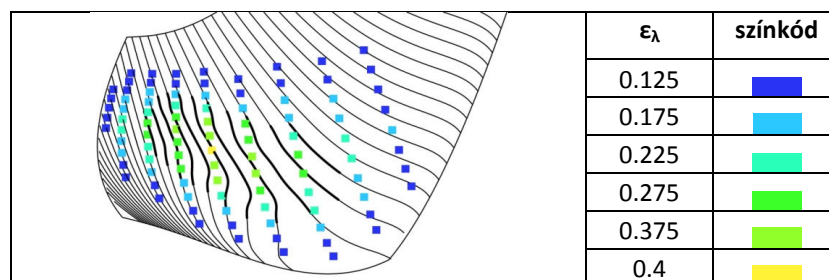
$$\epsilon_\lambda = \text{MAX.H} \cdot b_\lambda \cdot \chi \quad (38)$$

Ahol

MAX.H a hibás és a jó fényvonalak közti legnagyobb távolság
 χ szorzótényező

Az χ szorzótényezőt sebességi és megbízhatósági vizsgálatokkal határoztam meg. A vizsgálatokat és a vizsgálatok eredményeit a 9.1. fejezetben ismertetem.

A 32. ábra a keresési alterek és a fényvonal hibák mérete közti összefüggést egy példán szemlélteti. A négyzetek a hibás felületet reprezentáló kromoszóma génjeinek kontrollpontjait mutatják, a különböző színek a keresési alterek méretbeli különbségeit szemléltetik.



32. ábra. Példa a keresési alterek elhelyezkedésére és méretére (karosszériaelem felület)

A hidegebb színek kisebb, a melegebb színek nagyobb méretű alteret jelentenek (a színskála az ábra jobb oldalán látható). A jó fényvonalszakaszokat vékonyval, a hibásakat vastag görbékkel kiemelve ábrázolom. Látható hogy a hiba közepén, ahol a fényvonalak hibái a legnagyobbak és ahol a felület legnagyobb módosítása szükséges, ott a legnagyobbak a keresési térrészek is. A hiba pereme felé haladva a hibák fokozatosan csökkennek, egyre kisebb módosítás szükséges, a térrészek méretei ezt követve szintén csökkennek.

7.4. A kezdeti populáció létrehozása

A genetikus keresés akkor várható hogy hatékony és eredményes lesz, ha a keresés elején magas a populáció változatossága és a kromoszómák átlagos eltérése az eredeti, hibás felület kromoszómájától alacsony. Ezt a feladatot úgy oldottam meg, hogy a populáció kromoszómáit a javítandó felület c^0 kromoszómájából véletlen és közel szimmetrikus módosítások sorozatával hoztam létre. A művelet-hez a kontrollpontokat x, y, z koordinátairányonként külön, a következő összefüggéssel módosítom:

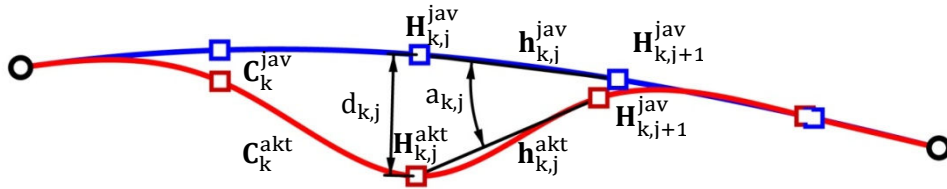
$$P_{\lambda}^k(x, y, z) = \begin{cases} P_{\lambda}^0(x, y, z) + \varepsilon_{\lambda} \cdot \rho_1 \cdot \rho_2 & \text{ha } 0.5 \leq \rho_2 < 1 \\ P_{\lambda}^0(x, y, z) - \varepsilon_{\lambda} \cdot \rho_1 \cdot \rho_2 & \text{ha } 0 < \rho_2 < 0.5 \end{cases} \quad (39)$$

ahol ρ_1, ρ_2 a $[0,1]$ intervallumból véletlenszerűen kiválasztott számok.
 ε_{λ} a P_{λ}^0 kontrollponthoz tartozó keresési térrész

Az összefüggéssel a kromoszóma $P_{\lambda}^0(x, y, z)$, $\lambda = 0 \dots \Lambda - 1$ génjeiben lévő kontrollpontok koordinátáit változtatom meg, koordinátairányonként. A kromoszómák változatosságát a ρ_1 , a közel változatlan átlagot a ρ_2 változó és a hozzá tartozó feltétel biztosítja.

7.5. Fitnessfüggvény

A cél- vagy fitnessfüggvénnyel a fényvonal struktúra hibáját fejezem ki. A függvényt a C_k^{jav} javított és a keresés során létrejövő C_k^{akt} aktuális fényvonalak közötti eltérésekből számolom. Az eltéréseket a fényvonalak közt értelmezett $d_{k,j}$ távolságokkal és $a_{k,j}$ szögekkel reprezentálom, amelyeket a $H_{k,j}^{akt}$ aktuális, illetve $H_{k,j}^{jav}$ javított fényvonalak pontjaiból állítok elő. A számítást a 33. ábra szemlélteti.



33. ábra. A $d_{k,j}$ távolságok és az $a_{k,j}$ szögek értelmezése

A fitnessfüggvényben a távolságok és a szögek az f_d távolság- és f_t szöghiba függvénykomponensként jelennek meg, a komponenseket a következő összefüggésekkel számolom:

$$f_d = \sum_{k=0}^{n_L-1} \sqrt{\left(\sum_{j=0}^{n_k-1} \left(d_{k,j} - \frac{1}{n_k} \sum_{j=0}^{n_k-1} d_{k,j} \right)^2 \frac{1}{n_k} \right)} \quad (40)$$

ahol

$d_{k,j} = |H_{k,j}^{akt} - H_{k,j}^{jav}|$ az aktuális, és a javított fényvonal-pontjai közötti távolságok a C_k fényvonalon.

n_k – a fényvonal-pontok száma a C_k fényvonalon

L – a fényvonalak száma

Az összefüggéssel a fényvonalak közötti távolságok szórását összegezem a fényvonalakra. A komponens a fényvonal struktúra javítottóságát a pontosság szempontjából reprezentálja.

$$f_t = \sum_{k=0}^{n_L-1} \sqrt{\left(\sum_{j=0}^{n_k-1} \left(a_{k,j} - \frac{1}{n_k} \sum_{j=0}^{n_k-1} a_{k,j} \right)^2 \frac{1}{n_k} \right)} \quad (41)$$

ahol

$$a_{k,j} = \arccos \left(\frac{\mathbf{h}_{k,j}^{\text{akt}} \cdot \mathbf{h}_{k,j}^{\text{jav}}}{\|\mathbf{h}_{k,j}^{\text{akt}}\| \|\mathbf{h}_{k,j}^{\text{jav}}\|} \right) \quad \text{az aktuális és a javított fényvonal egymást követő fényvonal-pontjaiból számolt } \mathbf{h}_{k,j}^{\text{akt}} \text{ és } \mathbf{h}_{k,j}^{\text{jav}} \text{ szakaszok közötti szög.}$$

$$\mathbf{h}_{k,j}^{\text{akt}} = \mathbf{H}_{k,j+1}^{\text{akt}} - \mathbf{H}_{k,j}^{\text{akt}}, \quad \mathbf{h}_{k,j}^{\text{jav}} = \mathbf{H}_{k,j+1}^{\text{jav}} - \mathbf{H}_{k,j}^{\text{jav}}$$

Az összefüggéssel a fényvonalak között értelmezett szögek szórását összegezem a fényvonalakra. A komponens a fényvonal struktúra javítottóságát formai szempontból reprezentálja.

A fitnessfüggvény megszerkesztésénél azt tapasztaltam, hogy a távolságfüggvény komponens a keresési tér azon részeire irányítja a keresést, ahol pontosságra jobb, de alakra kevésbé jó megoldások találhatóak. A szögfüggvény komponens ezzel ellentétesen viselkedik: alakra jobb, de pontosságra kevésbé jó megoldások felé viszi a keresést.

A genetikus keresés hatékonysága szempontjából az bizonyult helyesnek, ha a keresés előrehaladtával a két komponens arányát ennek megfelelően úgy változtatom, hogy a keresés elején a távolság, a végén a szögfüggvény komponens érvényesül. Az arány változását a w_d , távolság- és w_t súlyokkal valósítottam meg.

Legyen F^k a C^k kromoszóma fitnessfüggvénye, akkor:

$$F^k = w_d f_d^k + w_t f_t^k \quad (42)$$

A w_d és w_t súlyok meghatározására, a keresés során történő változtatására különböző stratégiákat dolgoztam ki, a leginkább megfelelőt, több szempontot is figyelembe vevő vizsgálatban határoztam meg. A vizsgálatokat és eredményeiket a 9.3. fejezetben ismertetem.

7.6. Megállási feltételek

A fitnessfüggvény feladata hogy a genetikus keresés olyan fényvonalakat, és ezen keresztül olyan felületet állítson elő, amelyik megfelel a javított felülettel szemben támasztott esztétikai és műszaki feltételeknek. Ezeket a feltételeket a genetikus keresésben megállási feltételek reprezentálják, amelyeket erre a célra létrehozott mutatókból számítok.

A mutatókat a javított és az aktuális fényvonalakat illetve a javított és az aktuális felületeket összehasonlítva számítom. Ez utóbbiakat csak a fényvonal javítás eredményeinek ellenőrzéséhez, a GA kifejlesztése során használtam. A GA kifejlesztéséhez alkalmazott vizsgálatokban a hibás felületeket mesterségesen, jó felületek kontrollpontjait módosítva állítottam elő. Ez azért volt szükséges, mert a felületek összehasonlításával számolt mutatókhoz a javított (jó) felület különben nem állna rendelkezésre. A következőkben a mutatók számítását és tartalmát ismertetem.

7.6.1. Megállási feltételekhez alkalmazott mutatók számítása

Az NRMSE.H mutató az aktuális és a jó fényvonal-pontok közti távolságok normalizált négyzetes közepe a fényvonalakra összesítve:

$$\text{NRMSE.H} = \sum_{k=0}^{n_L-1} \frac{\sqrt{\sum_{j=0}^{n_k-1} (\mathbf{H}_{k,j}^{\text{akt}} - \mathbf{H}_{k,j}^{\text{jo}})^2}}{n_k} \quad (43)$$

$$\left(\|\mathbf{H}_{k,j}^{\text{akt}} - \mathbf{H}_{k,j}^{\text{jo}}\| \right)_{\max} - \left(\|\mathbf{H}_{k,j}^{\text{akt}} - \mathbf{H}_{k,j}^{\text{jo}}\| \right)_{\min}$$

Ez a mutató azt fejezi ki, mennyire jól közelítették meg az aktuális felület fényvonalai a jó felület fényvonalait.

Az INF.H mutató a $\mathbf{K}_{k,j} = \mathbf{H}_{k,j}^{\text{akt}} - \mathbf{H}_{k,j}^{\text{jo}}$ pontokból álló diszkrét térgörbék I_k inflexió pontjainak számából [Paláncz, 2011] előállított átlag.

$$\text{INF.H} = \frac{\sum_{k=0}^{n_L-1} I_k}{n_L-1} \quad (44)$$

A mutatóval az aktuális fényvonalak formai minőségét mérem.

MAX.H az aktuális és a jó fényvonal-pontok között mért legnagyobb távolságok a fényvonalakra összesítve:

$$\text{MAX.H} = \sum_{k=0}^{n_L-1} \left(\left\| \mathbf{H}_{k,j}^{\text{akt}} - \mathbf{H}_{k,j}^{\text{jo}} \right\| \right)_{\max} \cdot \frac{1}{n_L-1} \quad (45)$$

A MAX.H mutatót ezért vezettem be, mert az aktuális és a javított fényvonalak közötti távolság eltérések sokszor olyanok, hogy a hiba legnagyobb része a hibás fényvonalszakasz egy kis szakaszára korlátozódik, ami az NRMSE.H mutatóval nem érzékelhető.

A következő mutatók az aktuális és a jó felületek közötti eltérést veszik alapul, és amiket csak a GA kifejlesztéséhez használtam. A NRMSE.S mutató a javított és a jó felület közti (fényvonalak mentén mért) merőleges távolságok négyzetes közepe a hibás fényvonalszakaszokra összesítve:

$$\text{NRMSE.S} = \sum_{k=0}^{n_L-1} \frac{\sqrt{\frac{\sum_{j=0}^{n_k-1} (\mathbf{H}_{k,j}^{\text{akt}} - \mathbf{S}_{k,j}^{\text{jo}})^2}{n_k}}}{\left(\left\| \mathbf{H}_{k,j}^{\text{akt}} - \mathbf{S}_{k,j}^{\text{jo}} \right\| \right)_{\max} - \left(\left\| \mathbf{H}_{k,j}^{\text{akt}} - \mathbf{S}_{k,j}^{\text{jo}} \right\| \right)_{\min}} \quad (46)$$

Az INF.S a $\mathbf{K}_k = \mathbf{H}_{k,j}^{\text{akt}} - \mathbf{S}_{k,j}^{\text{jo}}$ pontokból álló diszkrét térgörbék I_k inflexió pontjainak számából előállított átlag.

$$\text{INF.S} = \frac{\sum_{k=0}^{n_L-1} I_k}{n_L-1} \quad (47)$$

A mutatóval az aktuális felület formai minőségét mérem.

MAX.S az aktuális és a jó felület között mért legnagyobb merőleges távolságok a hibás fényvonalszakaszokra összesítve:

$$\text{MAX.S} = \sum_{k=0}^{n_L-1} \left(\left\| \mathbf{H}_k^{\text{akt}} - \mathbf{S}_k^{\text{jo}} \right\| \right)_{\max} \cdot \frac{1}{n_L-1} \quad (48)$$

A megállási feltételeket a mutatók százalékos vagy abszolút értékben értelmezett javítottságából számolom. A százalékos javítottságnál a megállási feltételek értéke a vizsgált felületen számolt mutatók 80%-os illetve a fényvonalain számolt mutatók 90%-os javítottsága. Az abszolút értékben számolt megállási feltételek esetén mindegyik vizsgált felületnél a legkisebb hibával rendelkező felületen számolt megállási feltételeket alkalmazom.

7.6.2. Biztonsági megállási feltétel

A biztonsági megállási feltétel feladata megakadályozni, hogy a genetikus keresés végtelen ciklusba kerüljön. Biztonsági megállási feltételként a generációk maximális számát alkalmazom. A vizsgálatokban a legjobb algoritmusok a megállási feltételek legtöbbször 60 generáció után kivétel nélkül mind elérték. Ez alapján a biztonsági megállási feltételt 100 generációban határoztam meg.

7.7. Szűrők

Tapasztalatom szerint a keresés előrehaladtával az aktuális és a javított fényvonalak között mért távolságok és szögek közül az elhanyagolható méretűek számaránya rohamosan nő. Ez a növekvő arány a fitnessfüggvényt (a benne lévő átlag és variancia összefüggés miatt) úgy befolyásolja, hogy egyre

kevésbé képes kimutatni a fényvonalak közötti különbségeket. Ennek a következménye a genetikus keresés konvergenciájának indokolatlan lassulása, illetve az, hogy az algoritmus csak gyengébb megoldásokat képes találni. A jelenség kiküszöbölésére a fitnessfüggvény számításából az elhanyagolható méretű távolságokat és szögeket kiszűröm.

A feladatra leginkább alkalmas szűrési módszert több szűrőt megvizsgálva választottam ki. A vizsgálatokban a Gauss (GAU), egy medián (MAD) és egy konstans (KON) szűrőt értékeltem, amelyeket az alábbi összefüggésekkel számoltam:

$$GAU = \frac{1}{n_k} \sum_{j=0}^{n_k-1} \left(|X_k - K_\lambda \frac{1}{n_j} \sum_{j=0}^{n_k-1} X_j| \right) \quad (49)$$

$$MAD = \text{median}_{n_k}(|X_k - K_\lambda \text{median}_j(X_j)|) \quad (50)$$

$$KON = K_\lambda \cdot \text{NRMSE.H} \quad (51)$$

Ahol X a távolság vagy szög hiba értéke, n_k a fényvonal-pontok száma a C_k hibás fényvonalszakaszon, K_λ a levágási érték szorozója. A szűrők vizsgálatát és a vizsgálat eredményét a 9.3.2. fejezet tartalmazza.

7.8. Reprodukció

A genetikus algoritmusban a kromoszómák fitness értékeinek számítását azok a genetikus operátorok követik, amelyek a következő generáció kromoszómáit hozzák létre. Ezek közül elsőnek a reprodukció operátort (elitizmus stratégia) alkalmazzák. Az operátor célja a legjobb (legmagasabb fitness értékkel rendelkező) kromoszómák genetikai információinak megőrzése. Ehhez az operátor a következő generációba változtatás nélkül bemásolja a legjobb kromoszómákat. A tapasztalat azt mutatta, hogy a GA akkor működött megfelelően, ha a reprodukcióval bekerült kromoszómák száma a populáció 3-5 % tették ki. A GA megtervezése során 5% -körüli értékeket alkalmaztam, így például az 50-es populációméretnél az elit kromoszómák száma kettő.

7.9. Skálázás

A következő generáció többi kromoszómája a fitness értékük alapján kiválasztott kromoszómák keresztezésével és az ezt követő mutációjával jön létre. Ahogy a 7.11. fejezetben látni fogjuk, a keresztezési operátor olyan keresztezések közül kerül majd ki, amelyek hajlamosak arra, hogy túl gyorsan csökkentsék a kromoszómák változatosságát, ami növeli a korai konvergencia esélyét, illetve a keresés késői szakaszában a populáció változatosságának indokolatlan csökkenését. A keresztezési operátoroknak ezt a hatását úgy egyensúlyozzák, hogy a keresztezés történő kiválasztás előtt a kromoszómák fitness értékeit skálázással újraszámolják.

A skálázásnak számos változata létezik [Dumitrescu et al., 2000], [Deb et al., 2002], a GA kifejlesztésében azoknak az alkalmazását vizsgáltam meg, amelyek a lehetséges keresztezési típusokat figyelembe véve megakadályozhatják a korai konvergenciát, illetve a populáció változatosságának indokolatlan csökkenését. A következőkben a megvizsgált skálázási algoritmusok összefüggéseit és sajátosságait ismertetem.

Az algoritmusok mindegyike először a populáció kromoszómáinak fitness értékeit nagyság szerint sorba rendezi, majd egy $\phi(\beta)$ skálázó függvénnyel, a β nagyság szerinti sorszámuknak megfelelően, megváltoztatja.

Egyszerű skálázás

A fitness értékét a β sorszámával helyettesíti:

$$\phi(\beta) = \beta \quad (52)$$

Ez a skálázás a generációktól független (static scaling), a hatását elsősorban a keresés későbbi szakaszában fejti ki, amikor a populáció változatossága jelentősen lecsökken.

Sorrendi skálázás

A fitness skálázott értéke másodfokú függvény szerint változik, szintén generációszámtól függetlenül:

$$\phi(\beta) = \frac{1}{\sqrt{\beta}} \quad (53)$$

Az egyszerű skálázástól eltérően a sorrendi skálázás igyekszik kiegyenlíteni a legjobb fitnessértékek közti különbségeket, és a legrosszabbaktól megszabadulni. Ez a skálázás a keresés kezdeti szakaszában fejt ki leginkább a hatását, ahol a fitness értékek közti nagy különbségek miatt a magas fitness értékű egyedek túl gyorsan elszaporodhatnak a populációban, ami megakadályozná, hogy az algoritmus a keresési tér bizonyos részeit elérje.

Lineáris skálázás

Ez a skálázás függ a generációszámtól (dynamic scaling), képes adaptálni magát a genetikus keresés különböző szakaszaihoz.

$$\phi(\beta) = p - \frac{2(\beta-1)(p-1)}{\beta_{\max}-1} \quad (54)$$

Ahol β_{\max} a populáció mérete, nem számítva az elit kromoszómákat.

A $p = 0$ közeli értéknél a skálázás a legjobb kromoszómák túlélését segíti. Az $p = 1$ közeli érték a kromoszómák túlélési esélyeit kiegyenlíti. A p ajánlott értékét a populáció F_{avg} átlag és F_{max} legjobb fitnessértékének hányadosából számolják:

$$p = 1 - \frac{F_{\text{avg}}}{F_{\text{max}}} \quad (55)$$

A skálázó algoritmusok vizsgálatát és a vizsgálat eredményeit a 9.5. fejezetben ismertetem.

7.10. Kiválasztás (szelekció)

A kiválasztás operátor a (skálázott) fitness értékek alapján kromoszómákat választ a keresztezés operátorhoz. A kiválasztást a fitness arányos (rulett kerék) algoritmussal valósítottam meg [Herrera et al., 1998]. Az algoritmus a fitnessértékek összegét a [0,1] tartománnyal felelteti meg, és a fitnessértékek arányában résztartományokra osztja, a résztartományok egy-egy kromoszómát reprezentálnak. A kiválasztás egy 0 és 1 között véletlenszerűen kiválasztott számmal történik, az a kromoszóma kerül kiválasztásra, amelyik tartományába a véletlen szám esik. A legnagyobb esélye a legnagyobb résztartománnyal rendelkező kromoszómáknak van, de a kevésbé jóknak is van esélye. A ϕ_i fitness értékű kromoszóma kiválasztásának q_i valószínűsége:

$$q_i = \frac{\phi_i}{\sum_{i=0}^{\beta_{\max}-1} \phi_i} \quad (56)$$

Ahol β_{\max} a kromoszómák száma a populációban.

Alternatívaként a versenyztetéses (tournament) algoritmust [Herrera et al., 1998] vizsgáltam meg. Előnye a fitness arányos kiválasztással szemben a kiválasztás esélyének szabályozhatósága. Erre azonban nincs szükség, mivel ez a fajta szabályozás a skálázás során már megvalósult.

7.11. Keresztezés

A keresztezés operátor feladata hogy a kiválasztott kromoszómákat felhasználva új kromoszómákat hozzon létre a következő generációhoz. A valós kódolású genetikus algoritmusokhoz számos keresztezés operátort fejlesztettek ki [Deb et al., 2002], tapasztalatom szerint a jelen genetikus algoritmushoz az aritmetikai típusú, kétszülős operátorok alkalmazhatók a legjobban. Ezek az operátorok az utódok létrehozásához egyfajta valószínűségi eloszlást vesznek alapul a szülő kromoszómák körüli tartományban. Egyes operátorok a szülő kromoszómák definiálta tartomány közepe körüli megoldá-

sokat, mások a szülők körüli megoldásokat részesítik előnyben. A következőkben a megvizsgált keresztezési operátorokat ismertetem.

Legyen $c_k^1 = (g_0^1, \dots, g_\lambda^1 \dots g_{\lambda-1}^1)$ és $c_k^2 = (g_0^2, \dots, g_\lambda^2 \dots g_{\lambda-1}^2)$ a két kiválasztott, keresztezésben résztvevő, szülő kromoszóma, és legyen $O_\tau = (o_0^\tau, \dots, o_k^\tau \dots g_{\frac{k}{2}-1}^\tau)$ a lehetséges utódok halmaza, ahol K a keresztezéssel létrehozandó kromoszómák száma a populációban, és τ az utód kromoszómák azonosítója.

Egyszerű aritmetikai keresztezés (Michalewicz)

Két utódot hoz létre: $O_\tau = (o_0^\tau, \dots, o_k^\tau \dots g_{\frac{k}{2}-1}^\tau)$ $\tau = 1, 2$

ahol

$$o_k^1 = \vartheta g_\lambda^1 + (1 - \vartheta)g_\lambda^2 \text{ és } o_k^2 = \vartheta g_\lambda^2 + (1 - \vartheta)g_\lambda^1$$

Az utódok szülőkhöz képesti helyzetét az ϑ paraméter határozza meg. Értéke a problémához igazítandó konstans. A vizsgálatokban a szakirodalomban [Herrera et al., 1998] ajánlott $\vartheta = 0.8$ értéket alkalmaztam.

BLX- α keresztezés (Eshelman)

Egy utód $O_\tau = (o_0^\tau, \dots, o_k^\tau \dots g_{\frac{k}{2}-1}^\tau)$, $\tau = 1$ jön létre, amit a szülők és az α tényező meghatározta $[c_{\min} - I\alpha, c_{\max} + I\alpha]$ tartományból választ ki véletlenszerűen a módszer, $c_{\max} = \max(c_k^1, c_k^2)$, $c_{\min} = \min(c_k^1, c_k^2)$, $I = c_{\max} - c_{\min}$.

Az α tényező módosításával szabályozható a tartomány mérete (minél szélesebb, annál nagyobb az keresési tér explorációja). Az $\alpha = 0$ érték esetén az egyszerű aritmetikai keresztezés műveleti tartományát kapjuk, míg $\alpha = 1$ esetén a tartomány méretét kétszeresére növeljük. Az α értéke tapasztalati, a jelen algoritmusban a geometriai alkalmazásokhoz közölt $\alpha = 0.2$ [Deb et al., 2002] értéket alkalmaztam.

BLX. α . β keresztezés (Eshelman)

A tartomány szélességének meghatározása ugyanúgy történik, mint a BLX- α módszernél, azzal a különbséggel, hogy a szülőkhöz képest aszimmetrikusan is elhelyezkedhet. Az aszimmetria a β tényezővel szabályozható.

$$[g_{\min} - I\alpha, g_{\max} + I\alpha\beta]$$

A β tényező értékét a vizsgálatok során határoztam meg, a legjobb eredményeket a $\beta = 0.4$ -gyel értem el.

Wright féle heurisztikus keresztezés

Egy utódot hoz létre, a populáció legjobb kromoszómájának génjeit a $\rho [0,1]$ véletlen változó szerint módosítja:

$$o_k = \rho(c_k^1 - c_k^2) + c_k^1$$

Az operátorok vizsgálatát és a vizsgálat eredményét a 9.4. fejezet tartalmazza.

7.12. Mutáció

A mutáció az új generáció mutációra kiválasztott kromoszómáinak egyik véletlenszerűen kiválasztott génjét változtatja meg. A gén új értékét a mutáció a génhez tartozó keresési tartományt (7.3. fejezet) felhasználva szintén véletlenszerűen választja ki. A tartomány mutációban alkalmazott mérete (a mutáció maximuma) a keresés során lehet állandó (eredeti méretű), vagy pedig generációnként csökkenő. A GA kifejlesztése során azt tapasztaltam, hogy az állandó méretű tartományt alkalmazó mutációk a keresés végén túl sokat változtattak a kromoszómákon, amelynek az eredménye a konvergencia megghiúsulása vagy gyengébb eredmények voltak. A GA kifejlesztésében ezért a generációnként csökkenő tartományt alkalmazó mutációkat vizsgáltam meg. A következő bekezdésben a megvizsgált mutációs algoritmusok összefüggéseit és sajátosságait ismertetem.

A mutációra kiválasztott kromoszómák számát a mutációs arány szabályozza. Az algoritmusban alkalmazandó mutációs arányt külön vizsgálatban határoztam meg, a vizsgálat eredményeit a 9.7. fejezet tartalmazza.

Mühlenbein féle mutáció

A mutáció maximuma a kromoszómák közötti különbség függvényében csökken. A mutált c'_k kromoszóma a következő összefüggéssel áll elő:

$$c'_k = c_k^0 \pm 0.1(c_k^{\max} - c_k^{\min})\delta \quad (57)$$

ahol

$$c_k^{\max} = c_k^0 + r_k, \quad c_k^{\min} = c_k^0 - r_k$$

r_k keresési tér mérete,

c_k^0 az eredeti kromoszóma,

δ a következő összefüggésből számolt véletlen szám:

$$\delta = \sum_{k=0}^{k=15} \alpha_k 2^{-k}.$$

ahol:

$$\alpha_k \in \{0,1\}, \quad p(\alpha_k = 1) = \frac{1}{16}.$$

Michalewitz féle nem egyenletes mutáció

A keresés előrehaladtával, a mutáció maximális értéke a következő összefüggés szerint változik [Michalewitz, 1996]:

$$c'_k = \begin{cases} c_k + \Delta(\gamma, c_k^{\max} - c_k) & \text{ha } 0.5 \leq \varrho < 1 \\ c_k - \Delta(\gamma, c_k - c_k^{\min}) & \text{ha } 0 < \varrho < 0.5 \end{cases} \quad (58)$$

ahol a ϱ értéke a $[0,1]$ intervallumból véletlenszerűen kiválasztott szám

$$\Delta(\gamma, \Delta c_k) = \Delta c_k \left(1 - \varrho^{\left(1 - \frac{\gamma}{\Gamma}\right)^{\varrho}}\right)$$

ahol

$$\Delta c_k = c_k^{\max} - c_k,$$

Γ generációk maximális száma,

γ az aktuális generáció sorszáma.

A vizsgálatok során alkalmazott paraméterértékek: $\Gamma = 100$, $\varrho = 0.8$. A mutációk vizsgálatát és a vizsgálat eredményeit a 9.6 fejezetben ismertetem.

8. A GA megtervezésének munkamódszere

A GA megtervezése során először egy működő alapváltozatot hozok létre, amit aztán lépésenként javítok. A lépésekben a GA egy-egy összetevőjét (operátorok, paraméterek) vizsgálom és határozom meg, amelyet innentől kezdve megfelelőnek tekintek, és továbblépek a következő összetevő meghatározására. Az összetevőket a GA változat keresési folyamatának alakulásával, a megállási feltétel elérésének sebességével és megbízhatóságával értékelem. Az értékelés több szempontot mérlegelve, mutatók és mérőszámok alapján történik.

8.1. A GA alapváltozat létrehozása

A GA alapváltozatban az összetevők lehető legegyszerűbb, leggyorsabban számítható változatait alkalmaztam, amelyeket a szakirodalom, és az előző fejezetben tárgyalt szempontok alapján választottam ki.

A GA kromoszómáit a javítandó felületrészre ható kontrollpontokból álló gének alkotják (7.1. fejezet). A gének, mivel a kontrollpontokat meghatározó koordináták folytonosak, valós kódolásúak. A keresési teret a fényvonalak hibáiból a (38) összefüggéssel határoztam meg, a χ szorzótényező értékét $\chi = 1$ -nek vettem. A fitness függvény a (40) és (41) összefüggésekkel definiált függvénykomponensek egyszerű, súlyozatlan összege. A megállási feltétel az NRMSE.H mutató 90%-os javíthatósága. A következő táblázat az alapváltozat összetevőit és az összetevők paramétereit foglalja össze:

összetevő	típus/számítás	paraméter
kódolás	valós (RCGA)	-
génreprezentáció	kontrollpontok	-
keresési tér	hibás és a javított fényvonalak közötti különbségből	$\chi=1$
kezdeti populáció létrehoz.	gének véletlen változtatásával	
fitness	távolság és szöghiba komponens összege	$w_d = 1$ $w_t = 1$
adatszűrők fitnesshez	-	-
megállási feltétel	NRMSE.H mutató 90%-os javíthatósága	-
biztonsági megállási feltétel	maximális generációs szám	100
reprodukción	igen	5%
skálázás	egyszerű	-
kiválasztás	fitness arányos	-
keresztelés	egyszerű aritmetikai (Michalewicz)	-
mutáció	Michalewicz féle nem egyenletes	$\Gamma = 100$ $q = 0.8$
mutációs arány		6%
populáció mérete		K=50

1. Táblázat. A GA alapváltozat összetevői és az összetevők paramétereit

A GA programkódjának megírásánál fontosnak tartottam, hogy jól strukturált és könnyen áttekinthető legyen, aminek érdekében a programkódot modulárisan építettem fel. Először alapfunkciók programkódjait írtam meg (pl. vektorműveletek, sorba rendezés stb.), amelyeket összeépítve létrehoztam a GA összetevők programkódjait, majd ezekből a GA-t. Az összetevők programkódjainak viselkedését és gyorsaságát a GA-ba történő beépítés előtt alaposan megvizsgáltam. A működő GA alapváltozat kísérletezéssel hoztam létre, amihez egy közepes méretű és kiterjedésű hibát tartalmazó tesztfelületet használtam.

8.2. A GA genetikus keresési folyamat szerinti értékelése

A GA összetevők megfelelőségét a GA sebességével (megállási feltétel eléréséhez szükséges generációk számával), megbízhatóságával, és a genetikus keresés folyamatának alakulásával értékeltem. A keresési folyamat értékeléséhez mutatókat hoztam létre, amelyek a kromoszómák változatosságát és a kromoszómák átlagát mérik. A következő részben mutatókat és keresési folyamat értékelésének módját ismertetem.

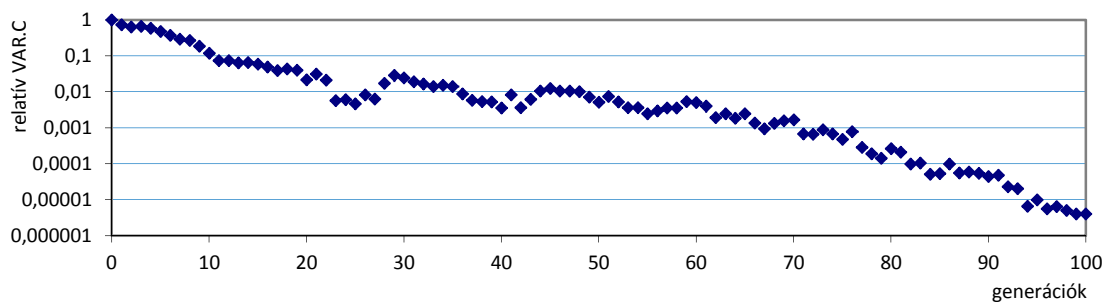
A keresés kezdeti szakaszában a GA feladata a keresési tér minél teljesebb felderítése (explorációja) annak érdekében, hogy az esélytelen régiókat kizárja, „megállapítsa” melyik régiók azok, amelyek jó eséllyel tartalmazhatják a megoldást. A kezdeti generációk kromoszómáinak változatossága (diverzitása) ezért nagy kell legyen.

A keresés következő szakasza, a felderítéssel „megszerzett” információk hasznosítása (exploitáció). A kromoszómák itt már csak kismértékben változtathatók, aminek a következtében a változatosságuk is sokkal kisebb, mint a keresés elején. A kromoszómák változatosságának csökkenése a jól működő GA esetén arányban áll a fitness javulásával.

A kromoszómák változatosságának mérésére a VAR.C mutatót hoztam létre. A mutató a változatosságot a gének varianciájából számolja, úgy hogy a kromoszómákban azonos helyen lévő gének (kontrollpontok) közti távolságok varianciáját összegezi a populációra:

$$\text{VAR.C} = \sum_{\kappa=0}^{K-1} \sum_{\lambda=0}^{\Lambda-1} \text{var}(g_{\lambda}(\kappa)) \quad (59)$$

Tapasztalatom szerint a jól működő genetikus keresés VAR.C függvénye, a fitness függvényhez hasonlóan, a keresés elején hirtelen csökken, majd a csökkenés sebessége az exploitációs szakasz elején nagymértékben lassul, és a függvény nulla közelében stabilizálódik. Egy ilyen függvényre mutat példát a 34. ábra (az értékskála logaritmikus, az értékek a kezdeti populáció értékeihez arányítottak). Tapasztalatom szerint azok a genetikus keresések bizonyultak a leginkább eredményesnek, amelyek VAR.C értéke a keresés végén sem csökkent teljesen nullára.

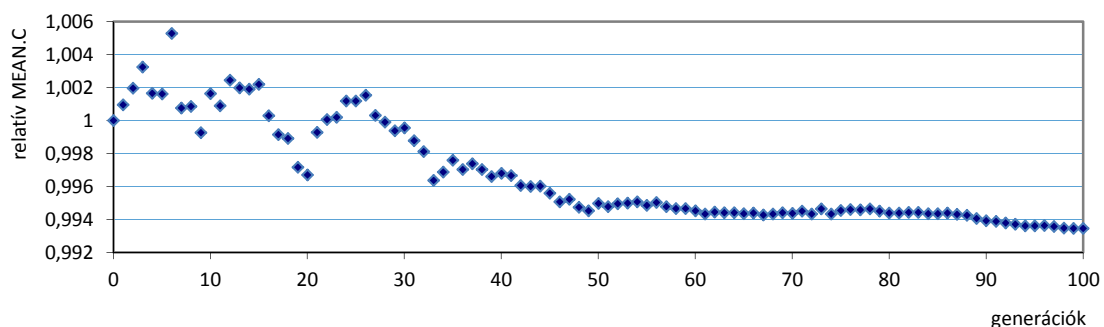


34. ábra. Példa a sikeres genetikus keresés VAR.C függvényére

A genetikus keresés folyamata akkor a leginkább hatékony, ha a kromoszómák átlaga a keresés során nem tér el jelentősen a kezdeti populációban számolt átlagtól [Herrera et al., 1998]. A kromoszómák átlagának mérésére a MEAN.C mutatót hoztam létre, értéke a kromoszómában azonos helyen lévő gének (kontrollpontok) közti távolságok átlaga a populációra összegezve:

$$\text{MEAN.C} = \sum_{\kappa=0}^{K-1} \sum_{\lambda=0}^{\Lambda-1} \frac{g_{\lambda}(\kappa)}{\Lambda} \quad (60)$$

A sikeres genetikus keresések MEAN.C függvénye, a keresés elején változik a legtöbbet, ami a magas VAR.C érték következménye. A kezdeti nagyobb változások után a függvény stabilizálódik, és beáll a megoldáshoz tartozó átlagértékre. A 35. ábra (az értékek a kezdeti populáció értékeihez arányítottak) a VAR.C bemutatásához használt keresési példa MEAN.C függvényét mutatja.



35. ábra. Példa a sikeres genetikus keresés MEAN.C függvényére

A függvény elején láthatók a magas VAR.C miatti hirtelen változások (vegyük észre hogy ezek csak pár százalékosak) ezt követi a stabilizációs szakasz, majd az 50-ik generáció körül a beállítás az új átlagértékre (0.994 körüli érték). Ennél a keresésénél a megoldást a 60 generáció környékén volt (megállási feltétel a NRMSE.H mutató 90%-os javíthatósága).

8.3. A GA sebesség és megbízhatóság szerinti értékelése

A vizsgálatok célja megállapítani, hogy a GA különböző összetevői, azok paraméterei hogyan befolyásolják az algoritmus sebességét illetve megbízhatóságát és megállapítani a GA ezen szempontok szerint megfelelő összetevőit, és azok paraméterértékeit.

A sebességet a megállási feltételek eléréséhez szükséges generációk számával, a megbízhatóságot több szempontot mérlegelve, mérőszámok alapján értékelem. A GA sebessége és megbízhatósága sokszor egymással ellentétesen követelmények: a sebességet csak a megbízhatóság kárára lehet növelni és fordítva. A GA kifejlesztésében azokat a GA változatokat részesítettem előnyben, amelyek megbízhatósága még elfogadható volt.

A következőkben a megbízhatóság értékelésében alkalmazott M1, M2 és M3 vizsgálati módszereket és a vizsgálati módszerekben alkalmazott mérőszámokat ismertetem. A módszereket és a mérőszámokat a tesztfelületek különböző nagyságú és kiterjedésű hibáinak javítása során szerzett tapasztalatok alapján dolgoztam ki.

Az M1 vizsgálat a megbízhatóság értékeléséhez a GA változat tesztfelületeken mért átlagos sebességét veszi alapul. Az átlagos sebességet az AVG.M1 mutatóval mérem, a mutató számításához a GA változatokat sebességük szerint sorba rendezem, és sorszámukkal helyettesítem.

$$\text{AVG.M1} = 1 - \frac{\sum_{i=0}^{n-1} N_i}{n \cdot M} \quad (61)$$

ahol

M	GA változatok száma
n	tesztfelületek száma
N_i	GA változat sorszáma a vizsgált tesztfelületen

Az M értéke vizsgálatonként más, így például a mutáció operátor vizsgálatában, ahol két mutáció változatot hasonlítottam össze M=2. A tesztfelületek száma állandó, n=10.

Minél nagyobb egy GA változat AVG.M1 értéke, megbízhatósága annál jobb. A megbízhatóság értékelésében a vizsgálati módszer figyelembe veszi, hányszor volt utolsó a GA változat, illetve volt-e olyan felület ahol nem érte el a megállási feltételt. Az első esetben annak függvényében hányszor volt utolsó annyi hellyel hátrébb sorolja a GA változatot. Ha van olyan eset ahol a GA nem éri el a megállási feltételt a GA változatot nem tekintem megbízhatónak. A vizsgálatban megállási feltételnek az NRMSE.H mutató 90%-os javítottságát tekintetem.

Az M2 vizsgálatban a megállási feltételnek alsó és felső határt definiáltam, és a megbízhatóságot aszerint értékeltem, hogy a GA képes volt-e az alsó határt elérni, illetve hány többletgenerációra volt szüksége a felső határ eléréséhez. Minél kevesebbre, a GA változatot annál megbízhatóbbnak tartottam. A GA változatokat mindegyik tesztfelületen értékeltem, a megbízhatóság eldöntésénél a legkedvezőtlenebb eredményeket vettem figyelembe. A megállási feltétel alsó határának az NRMSE.H mutató 90%-os, felső határának a mutató 95%-os javítottságát tekintetem.

Az M3 vizsgálatban a GA megbízhatóságát és sebességét egyszerre több megállási feltételt figyelembe véve értékeltem. A megállási feltételeket a 7.6.1. fejezetben ismertetett módon a következő mutatókból állítottam elő: NRMSE.H, INF.H, MAX.H illetve ezek felületen számolt változatai: NRMSE.S, INF.S, MAX.S. Megállási feltételnek a fényvonalakon számolt mutatók 90%-os a felületen számolt mutatók 80%-os javítottságát tekintetem.

A GA változatok megbízhatóságát itt az alapján döntöttem el, hogy milyen széles generációszám tartományban érték el a megállási feltételeket a különböző tesztfelületeken, és hogy hol volt a tartomány közepe. Minél kisebb volt a tartományok szélessége a GA változatot annál megbízhatóbbnak, minél kisebb értékű a tartomány közepe, a GA változatot annál gyorsabbnak tekintetem. Az értéke-

léshez a tartományokat közös diagramban ábrázolom (pl. 40. ábra). Az egyes sávok a tartományok szélességét és elhelyezkedését mutatják, a sávokat GA változatonként csoportosítottam.

A megbízhatóság értékelésében azt is figyelembe vettem, volt-e olyan eset ahol a GA változat nem érte el valamelyik megállási feltételt a maximális 100 generációszámán belül. Ha volt, a GA változatot nem tekintettem megbízhatónak. A GA változatokat mindegyik tesztfelületen értékeltem, és az értékeléseknél legkedvezőtlenebb eredményeket vettem figyelembe.

A javítás eredményeit vizuálisan is ellenőriztem, az ellenőrzéshez a jó és a javított fényvonalak alakját és egymástól való távolságát hasonlítottam össze, vizuálisan.

8.4. A GA megtervezésében alkalmazott tesztfelületek

A tesztfelületek kiválasztásának egyik szempontja, hogy nagyságukban, funkciójukban, bonyolultságukban jelentősen eltérőek legyenek, és hogy abból az ipari alkalmazási körből származzanak, amelyben a dolgozat elején ismertetett javítási problémák jelentkeznek.

A felületek kiválasztásának másik szempontja a lokális hibáik széles intervallumban változó nagysága, kiterjedése és komplexitása. A lokális hibák a felület finomstruktúra hibáihoz tartoznak, olyanok, amelyeket árnyékolt felületértékeléssel csak nehezen vagy egyáltalán nem lehet feltárni. A tesztfelületek hibáinak a nagyságát úgy választottam meg, hogy az árnyékolt felületértékeléssel már ne legyenek felfedezhetők. Azt hogy a hibák árnyékolás érzékenysége alattiak a „Rhino 3D Render” alkalmazással ellenőriztem, nagyságukat az NRMSE.H, MAX.H, NRMSE.S, MAX.S mutatókkal mérem. A tesztfelületek árnyékolt reprezentációi a 15.3. melléklet ábráin láthatók. A hibák közepes vagy kis kiterjedésűek, a kiterjedtségüket az AREA.S mutatja, ami a hibás fényvonalszakaszok alatti felület területe. A hibák komplexitását az INF.H mutatóval (44) mértem.

A felületek kiválasztásánál arra törekedtem, hogy a mutatóik értéktartományai minél szélesebbek legyenek. Az 2. táblázatban a kiválasztott felületek (S1-S10) hibáinak mutatói láthatók. Az egyes mutatók legkisebb és legnagyobb értéke közötti különbség 50% és 200% között mozog.

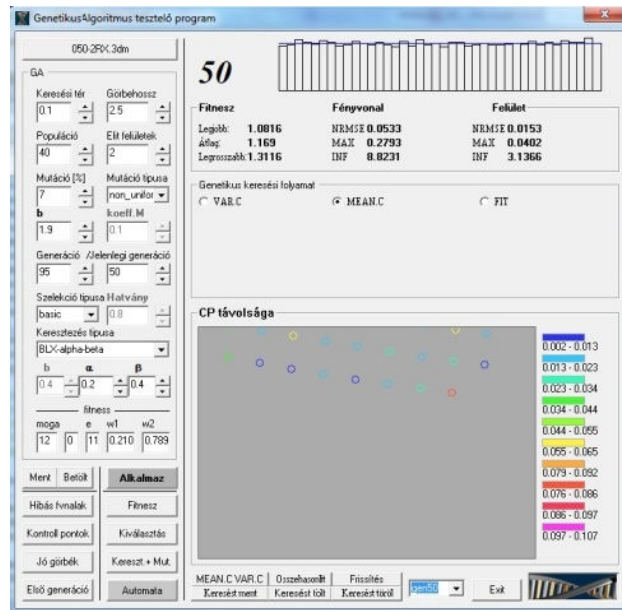
ID	Név	NRMSE.H	MAX.H	INF.H	NRMSE.S	AREA.S	MAX.S
S1	Audi csomagtartó f.	0.764	8.41	0.66	0.18	109981.4	1.17
S2	BMW sárvédő	0.048	0.21	0.44	0.025	375.7	0.135
S3	Ford csomagtartó f.	2.18	7.83	0.33	0.12	24099.7	0.586
S4	Szabadf. felület	1.21	6.05	0.55	0.009	10185.2	0.628
S5	Ford sárvédő	2.176	9.94	1.33	0.011	19703.0	0.80
S6	Térdprotézis (femur)	0.098	0.32	1.5	0.007	77.93	0.027
S7	Ford karossz. elem	2.35	8.08	1.82	0.19	91340.7	0.927
S8	Turbinalapát	0.18	0.51	0.2	0.008	103.8	0.039
S9	Propeller	0.41	1.87	1.1	0.041	4214.1	0.216
S10	FIAT sárvédő	0.88	5.50	1.52	0.24	15513.1	1.65

2. Táblázat. A tesztfelületek hibajellemzői

8.5. A genetikus algoritmus vizsgálatainak megvalósítása számítógépes környezetben

A tesztfelületeket „Rhino 3D NURBS modeler” felülettervező rendszerben kezelem, a felületek paramétereit a rendszer alkalmazásprogramozási interfészén (API) keresztül módosítom. A genetikus algoritmus és a vizsgálatok programkódjait ehhez az interfészhez illesztve írtam meg, VBA nyelven. A vizsgálatokat a hatékony megvalósítás érdekében, külön erre a célra létrehozott alkalmazással hajtom végre, a GA és a vizsgálatok programkódjait ezen alkalmazás kezelőfelületén keresztül működtetem. A programkódok paramétereik beállítása interaktív, ami meggyorsítja az adatbevitelt és csök-

kenti a hibás adatbevitel lehetőségét. A genetikus keresés műveletenként és generációként elemezhető. A műveleti lépések eredményei szöveges fájlokban a generációnkénti lépések eredményei szöveges fájlokban illetve, a kezelőfelületen megjelenítve, grafikus formában tanulmányozhatók. A 36. ábra az alkalmazás kezelőfelületét mutatja.



36. ábra. A genetikus algoritmust tesztelő alkalmazás kezelőfelülete

Az alkalmazás működtetése

A GA változat összetevői és az összetevők paraméterei a „GA” ablakkeretben adhatók meg. A fitness-függvény súlyai a „fitness” elválasztó vonal alatt található w1 és w2 mezőkben állíthatók be, a súlyok alakulása a keresés során ugyanezekben a mezőkben követhetők nyomon. A genetikus keresés a keret alján található gombokkal indítható és vezérelhető (léptetés, keresés mentése, betöltése, újraindítása).

Az keresés folyamata a jobb oldali keretben tanulmányozható. A keret a GA aktuális generációban mért jellemzőit (lásd 7.6.1 és 8.2 fejezet) mutatja grafikusán illetve számadatokkal. A kromoszómák fitnessértékei az első sorban, a generáció mutatói és mérőszámai a következő sorban láthatók. Az alattuk lévő keretben a keresés folyamatának mutatóit (VAR.C és MEAN.C) és a generációnkénti legjobb fitnessértékeket ábrázolom, grafikusán.

A keret közepén látható grafikus mezőben a populáció legjobb felülete és a jó felület kontrollpontjai közti távolságok generációnkénti változása tanulmányozható. A mező a jó felület kontrollpontjait mutatja körkként, felülnézetből (z koordináta tengely irányából nézve). A körök színei az említett kontrollpontok közötti távolságot reprezentálják, a színskála a mező jobb oldalán látható.

9. A GA megtervezése

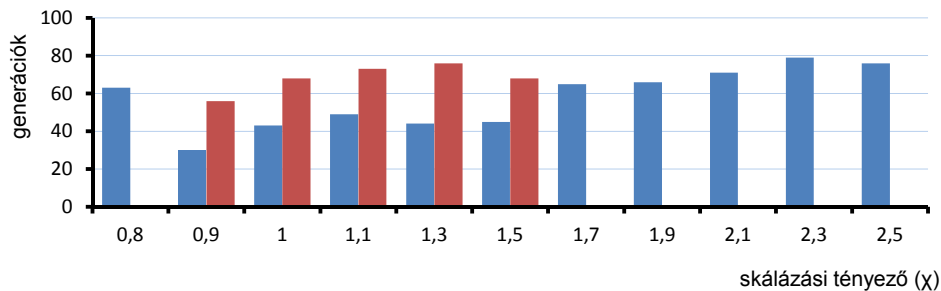
A megtervezés célja, olyan GA-t előállítani, ami megbízhatóan és gyorsan képes a megállási feltételeknek megfelelő megoldást találni. A tervezés során ehhez a GA összetevőit határozom meg. Ezek az összetevők a következők: keresési tér mérete, populáció mérete, kezdeti populáció létrehozási módszere, fitness függvény számítási módszere, skálázás módszere, keresztezési módszere, mutáció módszere és aránya.

Az előző fejezetben ismertettem a GA megtervezésének munkamódszerét a módszerben alkalmazott értékelési szempontokat és a hozzájuk kapcsolódó vizsgálatokat. Ismertettem a vizsgálatokban alkalmazott mutatókat, mérőszámokat és a tesztfelületeket, és kitértem a vizsgálatok megvalósításának számítógépes módjára is.

A következőkben a GA ezen munkamódszer szerinti megtervezését ismertetem. Az összetevők közül először a GA működését általában meghatározó paramétereket határozom meg: keresési tér nagysága és a populáció mérete. Ezt követi a felületjavítás feladathoz leginkább kapcsolódó összetevő, a fitnessfüggvény, majd a genetikus operátorok meghatározása.

9.1. A keresési tér nagyságának meghatározása

A keresési tér nagyságát a (38) összefüggést alkalmazva határozom meg. Az összefüggésben a MAX.H fényvonalak közötti legnagyobb távolság és a b_λ aránytényező közvetlenül számíthatók, a χ szorzótényezőt sebességi és M2 megbízhatósági vizsgálattal határozom meg. A 37. ábra az M2 vizsgálatban kapott generációs számokat mutatja a χ függvényében. A megállási feltétel alsó határának generációs számait kék, a felső határ generációs számait piros oszlopokkal jelöltem.



37. ábra. A GA sebessége (M2 vizsgálat) a χ skalázási tényező függvényében

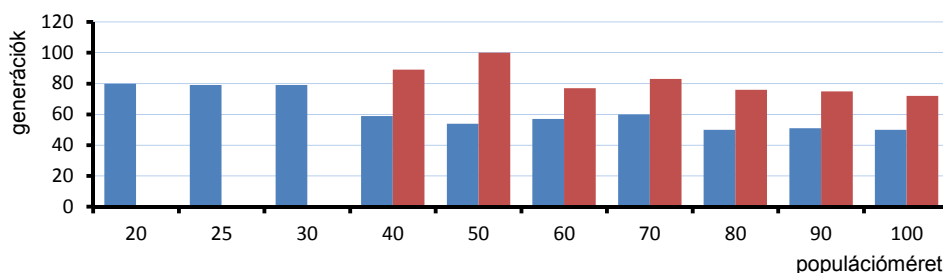
A GA a megállási feltétel alsó határát a $\chi=[0,8...2,5]$, a felsőt a $\chi=[0,9...1,5]$ tartományban érte el. A $\chi = 0,9$ értéknél az algoritmus mindkét határ szerint a leggyorsabb, a felső határhoz szükséges többlet generáció száma ennél az értéknél a legkisebb, és ez a legkisebb olyan érték is, ahol az algoritmus a megállási feltétel felső értékét elérte. Ezen érték felett a megállási feltétel alsó és felső határának generációs számai közel változatlanok, egészen $\chi = 1,5$ -ig. A $\chi = 1,5$ érték felett az algoritmus nem tudta elérni a felső határt, és a megállási feltétel alsó határához szükséges generációk száma is érezhetően nőni kezdett. A leginkább megfelelőnek a viszonylag gyors, és a megállási feltételt biztonságosan elérő GA-t eredményező $\chi = 1$ értéket választottam.

9.2. A populáció méretének meghatározása

A populáció méretének növelésével a GA gyorsabban és nagyobb biztonsággal találja meg az optimális megoldást [Herrera et al., 1998], ugyanakkor ez több függvényértékelést is jelent, amivel arányban nőhet a keresés időigénye. A populáció méretének csökkentésével csökken a keresés időigénye, de a jó megoldás megtalálásának egy határméret alatt, már nincs esélye. A cél olyan populációméretet meghatározni, amelynél az algoritmus megbízhatóan működik, és a keresés időigénye a lehető legkisebb.

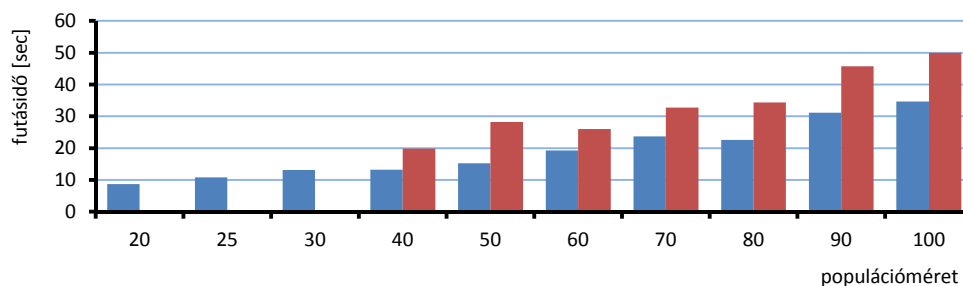
A populáció méretét sebességi és az M2-es megbízhatósági vizsgálattal határoztam meg. A vizsgálatokban először a megbízhatóság alapján behatároltam a populációméret (G) felső és alsó határát. A megbízhatóság szempontjából a populáció méretének nincs felső határa, felső határként a $G=100$ -as értéket alkalmaztam. Az alsó határt a felső határról kiindulva és a populáció méretét csökkentve határoztam meg. Alsó határnak azt a generációs számot tekintettem, ahol a GA még képes volt a megállási feltétel felső határát elérni.

A 38. ábra a megállási feltétel alsó illetve felső határához tartozó generációs számokat mutatja a populációméret függvényében. A GA a megállási feltétel alsó határát először a $G=20$ értéknél, a felsőt a $G=40$ értéknél érte el. A $G=40$ a méretnél a feltételhatárok generációs számai között viszonylag nagy a különbség, ami a populáció méretének növekedésével először tovább nő, majd lecsökken és stabilizálódik. A felső határ generációs számait vizsgálva az látható, hogy a $G=80$ értékig csökkennek, majd stabilizálódnak. Ezekből azt a következtetést vontam le, hogy a populáció méretét a megbízhatóság szempontjából a 40 és a 80 közé eső értékek közül érdemes választani.



38. ábra. A GA sebessége a populációméret függvényében (M2-es vizsgálat)

A 39. ábra a megállási feltételek elérésének időigényét mutatja a populációméret függvényében. A megbízhatósági tartományban $G=[40..80]$ a megállási feltételekhez szükséges idő lassan, de folyamatosan nőtt. Ebből arra lehet következtetni, hogy a populációméret növelése ebben a tartományban nem gyorsítja a keresés konvergenciáját vagy legalább is nem annyira, hogy az algoritmus futásidőjét csökkenteni tudná. Ezért a populáció méretét a megállási feltételt biztonsággal elérő tartomány elejéről $G=50$ értékre választottam.



39. ábra. A GA időigénye a populációméret függvényében (M2-es vizsgálat)

9.3. Fitnessfüggvény vizsgálata

A fitnessfüggvényt különböző összetevő súlyozási stratégiát alkalmazó változatok közül választottam ki. A változatok kidolgozásánál abból a tapasztalatból indultam ki, hogy az f_d távolság összetevő pontosabb, de kevésbé sima fényvonalakat produkáló felületek felé viszi a keresést, míg az f_t összetevő ezzel ellentétesen viselkedik (7.5. fejezet). Feltételezésem szerint, az eredményes javításhoz a keresés elején, amikor a populáció változatossága nagy, a pontosságot szükséges javítani. Később, amikor a változatosság lecsökken és a javítás már egy elfogadható pontosságot elért, a simaság javítására kell helyezni a hangsúlyt. A feltevés igazolására az olyan súlyozási stratégia változatokat dolgoztam ki, amelyek a két összetevő arányát úgy módosítják, hogy a fitness függvényben a keresés elején, amikor nagy a kromoszómák változatossága, a távolság-, a keresés végén, az alacsony változatosságnál, a szögfüggvény összetevő érvényesül. Az összetevők arányát w_d távolság és w_t szögkomponens súlyokkal módosítom.

Az F fitnessfüggvények alakja a következő:

$$F = w_d f_d + w_t f_t$$

ahol

w_d távolság összetevő súlya, $w_d = w_d^0 - \Delta w$

w_t szög összetevő súlya, $w_t = 1 - w_d^0 + \Delta w$

w_d^0 távolság komponens súlya a kezdeti populációban,

Δw súlykülönbség

A távolság és a szög dimenziója, értéktartománya különböző, a fitnessfüggvényben történő közös alkalmazásukhoz az értékeiket a következő összefüggések szerint normalizálom:

$$f_d = f_d \cdot \frac{1}{f_d^0}, \quad f_t = f_t \cdot \frac{1}{f_t^0} \quad (62)$$

ahol

f_d^0 és f_t^0 az eredeti, hibás felület fitnessfüggvényének távolság és szög összetevői.

A vizsgálatokban hat különböző súlyozási stratégiát alkalmazó fitnessfüggvény változatot (F1-F6) hasonlítottam össze. Az F1 és F2 fitnessfüggvények csak távolság illetve csak szögkomponenseket tartalmaznak. Az F3 változatban a súlyok aránya a keresés során végig ugyanaz. Az F4 változatban a súlyok aránya a keresés egy adott pontján megfordul. Az F1-F4 függvények előnye hogy gyorsan számíthatók, ugyanakkor jó eséllyel csak nagyon pontatlanul tudják követni a populáció változatosságát.

Az F5 és F6 függvényekben a súlyok arányának változtatását a populáció kromoszómáinak változatosságát mérő VAR.C értékéhez kötöttem. Az F5 változat a keresés elején csak a távolság összetevőt tartalmaz. Amint a kezdeti VAR.C érték egy bizonyos érték alá csökken, a távolságot a szög összetevő váltja fel. Az F6 fitnessfüggvényben a keresés kezdeti távolság/szög súly aránya a VAR.C csökkenésével fokozatosan fordul meg.

A súlyok arányát a függvényekben a Δw súlykülönbséggel változtatom. A súlyozási stratégiák paraméterértékeit és a Δw számítási szabályait a 3. táblázatban foglalom össze.

ID	paraméter(ek)	Δw számítási szabálya
F1	$w_d^0 = 1$	$\Delta w = 0$
F2	$w_d^0 = 0$	$\Delta w = 0$
F3	$w_d^0 = 0.75$	$\Delta w = 0$
F4	$w_d^0 = 0.8, a = 0.75$	$\Delta w = 0$ ha $\tau < a \cdot \tau_{\max}$ különben $\Delta w = 2w_d^0 - 1$
F5	$w_d^0 = 1, a = 0.15$	$\Delta w = 0$ ha $c_{\text{var}}^0 - c_{\text{var}}^{\tau-1} < a$ különben $\Delta w = 2w_d^0 - 1$
F6	$w_d^0 = 0.8$	$\Delta w = (2w_d^0 - 1) \cdot (1 - c_{\text{var}}^0 + c_{\text{var}}^{\tau-1})$

3. Táblázat. A fitness függvény változatok súlyozási stratégiái és számítási szabályaik

ahol

- a súlyarány csere együtthatója
- τ aktuális generáció sorszáma
- τ_{\max} generációk maximális száma ($\tau_{\max} = 100$)
- c_{var}^0 kezdeti populáció VAR.C értéke
- $c_{\text{var}}^{\tau-1}$ előző generáció VAR.C értéke

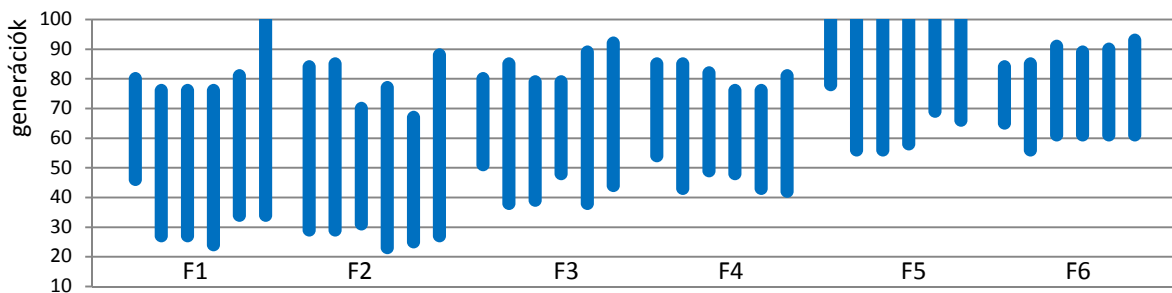
A leginkább megfelelő fitnessfüggvény változatot sebességi, M3-as megbízhatósági, illetve vizuális vizsgálatok alapján választottam ki.

Az M3 vizsgálat eredményeit a 40. ábra szemlélteti. A sávcsoportok egy-egy fitnessfüggvény változathoz tartozó eredményeket mutatják. A sávcsoponton belül az egyes sávok különböző mutatókból számolt megállási feltételek generációs szám tartományait ábrázolják, a következő sorrendben: NRMSE.H, INF.H, MAX.H, NRMSE.S, INF.S, MAX.S.

A GA változatok, az F5-öt és az F1-et alkalmazó eseteket kivéve, mindegyik felületnél, és mindegyik megállási feltételt elérték. Az F5 változatnál mindegyik megállási feltételnél volt legalább egyik olyan felület ahol a GA nem volt képes a megállási feltétel elérésére. Az F1 változatnál ilyen csak a MAX.S megállási feltételnél fordult elő.

Az értéktartományok sok esetben nagyon szélesek, mivel a különböző tesztfelületeknél a GA változat sebessége nagyban változott, ami miatt a megbízhatóságukat alacsonynak tekintettem. A leginkább széles értéktartományokat az F2 változatnál találjuk. A tartományok alsó értékét a többi változattal összehasonlítva az látható hogy az F2 változatnál a legalacsonyabbak. Ezzel a változattal az algoritmus jellemzően a legkisebb hibákat tartalmazó felületeken volt a leggyorsabb. A tartományok széles-

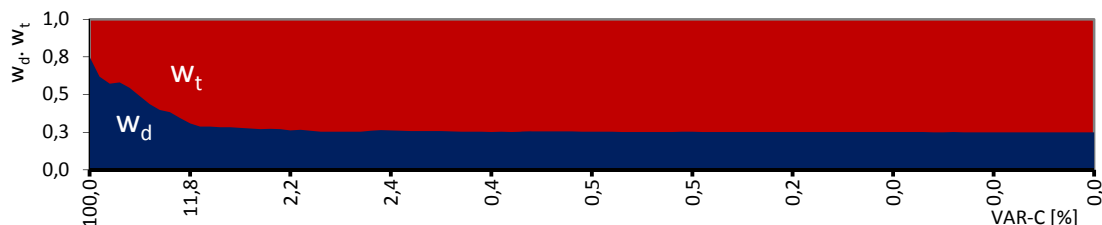
ségei az F3 és F4 esetén valamennyivel kisebbek, a legkisebb értéktartományokat az F6 változatnál láthatjuk. Az F6 tartományok közepe viszonylag alacsonyan helyezkedik el (az algoritmus ezzel a fitnessváltozattal közepesen gyorsnak tekinthető).



40. ábra. Fitness függvények M3 vizsgálatának eredményei

Összegezve elmondható hogy a kizárólag szög illetve távolság komponenst alkalmazó távolságfüggvények (F1, F2) sebessége jellemzően széles tartományban mozgott, amiből arra következtethetünk, hogy a sebességük nagyban függ attól, milyen felületen alkalmazzák őket. Az F2 fitnessfüggvénye csak szög összetevőt tartalmaz, és a kis hibák javításában a leginkább eredményes. Az F1 fitnessfüggvény csak távolság összetevőt tartalmaz, sebessége az F2-höz hasonlóan széles tartományban változik, de nem volt mindig képes elérni a megállási feltételt. A konstans, illetve az adott generációsárhoz kötött arányváltás stratégiái (F3, F4) megbízhatóbbak, kevésbé széles tartományban változik a generációk száma. A populáció változatosságához kötött két adaptív módszer (F5, F6) közül csak az F6 működött megbízhatóan, és közel kétszer olyan jól teljesített, mint az összes többi. Mindent összevetve a leginkább megfelelőnek a komponensarányt fokozatosan változtató F6 stratégia látszik a leginkább megfelelőnek, amit a fényvonalak vizuális értékelés is megerősített.

A 41. ábra az F6-os függvényváltozat szerinti súlyarány változás egy tipikus példáját mutatja. A kere-



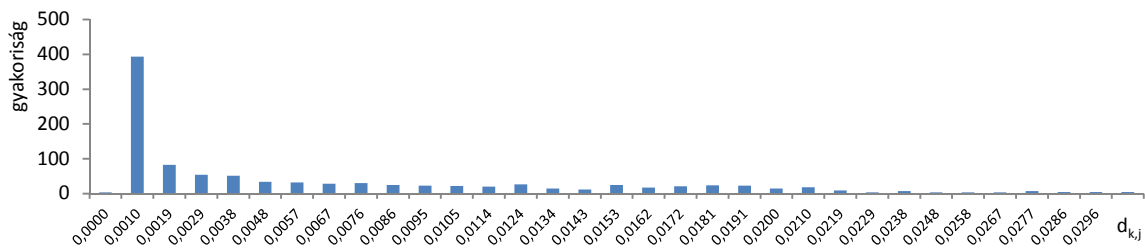
41. ábra. Példa az F6-os függvény szerinti összetevő súlyozásra (S6-os felület)

sés elején a távolság összetevő a domináns ($w_d=0.8$), ami azonban a kromoszóma variancia gyors csökkenésével hamar megfordul ($VAR.C=2.3$, illetve $\tau = 28$ érték körül). Ez a rövid dominancia és a fennmaradó befolyás azonban elég volt ahhoz, hogy a megoldás pontosságát megfelelően megnövelje, majd a keresés későbbi szakaszában a szög összetevő dominanciája mellett meg is tartsa.

9.3.1. Szűrő alkalmazása a fitnessfüggvényben

A fitnessfüggvény (40), (41) összefüggéseiben a hibák átlaga illetve varianciája szerepel. Ezért, a kis és elhanyagolható méretű hibák magas aránya elmosza a hibás fényvonalak közötti különbséget, és így a fitnessfüggvény képtelenné válik arra, hogy a kromoszómák (illetve felületek) közti különbségeket megfelelően kifejezze (érzékenysége lecsökken). Ez nagyban csökkenti a kiválasztás hatékonyságát, aminek a következménye a keresés indokolatlan lassulása és a GA megbízhatóságának csökkenése.

A probléma jellemzően a keresés későbbi szakaszaiban jelentkezik. A keresés elején a hibák döntő többsége nagy, az elhanyagolható méretű hibák száma alacsony. Később, ahogy a megoldások egyre hasonlóbbá válnak, ez megváltozik, és az elhanyagolható méretű hibák száma generációról generációra rohamosan nő, a keresés végére az elhanyagolható hibák kerülnek döntő többségbe. A 42. ábrán látható hisztogram erre a helyzetre mutat példát. Látható hogy (a nullát leszámítva) a legkisebb hiba (0.0010 mm) gyakorisága a legmagasabb.



42. ábra. Példa a $d_{k,j}$ távolság értékeinek eloszlására a keresés késői szakaszában

A probléma megoldásához a fitnessfüggvény számításából szűréssel kizártam a kis és az elhanyagolható méretű hibákat. A megfelelő szűrést sebességi és megbízhatósági vizsgálatokkal, több szűrési változat közül választottam ki. A változatok a 7.7. fejezetben ismertetett GAU és MAD illetve KON szűrőket alkalmazzák. A vizsgálatokat két részre osztottam. Az első részben a kiszűrt fényvonal-pontok száma és elhelyezkedése alapján meghatároztam a szűrés paramétereinek értelmezési tartományát. Ezt követően az egyes szűrők leginkább megfelelő paraméterértéket határoztam meg, sebességi, megbízhatósági és vizuális vizsgálatokkal. A GA-ban történő alkalmazásra a leginkább megbízható és gyors GA változatot eredményező szűrőt választottam ki.

9.3.2. A szűrők paramétereinek értelmezési tartománya

Az értelmezési tartományt a szűrőváltozatok működőképessége alapján határoztam meg, amit a kiszűrt fényvonal-pontok száma és elhelyezkedése alapján vizsgáltam. A működőképesség egyik feltétele, a szűrt pontok egytől vagy nullától eltérő aránya: nem tekintetem működőképeseznek azokat a szűrési változatokat, amelyek mindegyik fényvonal-pontot vagy egy fényvonal-pontot sem szűrték ki. A működőképesség másik feltétele hogy a szűrt pontok a hibás terület szélén, a szűretlen pontok a legnagyobb fényvonal hiba környékén koncentrálódjanak. A feltételek teljesülését a keresés elején (kezdeti generáció) és végén (megállási feltétel generációja) is ellenőriztem.

Az egyes szűrők változatai a levágási értékben különböznek. A levágási érték a K_λ tényezővel befolyásolható. A vizsgálat első részében a K_λ azon tartományát határoztam meg, ahol a szűrő működőképese volt. A szűrőket az előző vizsgálatokban alkalmazott tesztfelületeken vizsgáltam. A vizsgálattal behatárolt értéktartományokat az 4. táblázat tartalmazza.

	GAU	MAD	KON
K_λ	0.5 - 3	0.25-2	0.05-0.5

4. Táblázat. A megvizsgált szűrők működési tartományai

9.4. Keresztezés vizsgálata

A keresztezést először a GA alapváltozatban alkalmazott módszer paramétereit változtatva értékeltem, majd az így előállított legjobb módszert további keresztezési módszerekkel hasonlítottam össze.

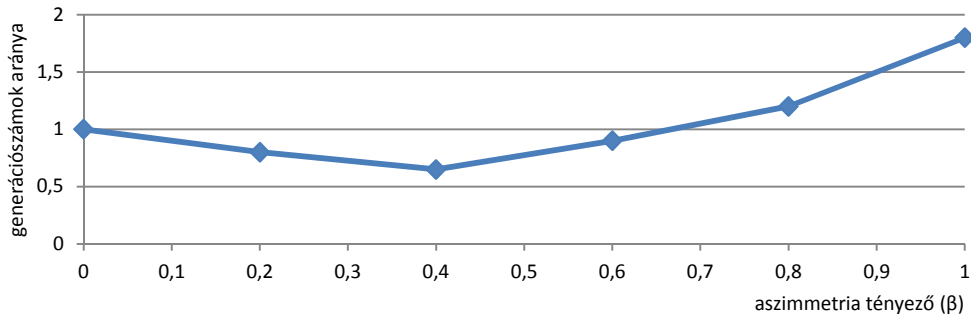
A GA alapváltozata a BLX. α módszert alkalmazza. A módszer az utód kromoszómákat a szülők meghatározta tartomány közepére szimmetrikus tartományból választja ki. A módszer változtatható paramétere a tartomány mérete, ami az α változóval befolyásolható (7.11. fejezet). Értéke az alapváltozatban $\alpha = 0.2$ (a kiválasztás tartománya a szülők meghatározta tartományhoz képest 20%-al nagyobb).

A vizsgálatok során először azt ellenőriztem, hogy létezik-e az $\alpha = 0.2$ -nél megfelelőbb érték is, amivel a GA teljesítménye növelhető. Az ellenőrzés az M1-es vizsgálattal történt, amiben az $\alpha = [0.15, 0.2, 0.25, 0.4]$ értékeket vizsgáltam meg. Az eredmények azt mutatták, hogy a GA az eddig alkalmazott $\alpha = 0.2$ -t értékkel teljesít a legjobban. A részletes eredmények az 5. táblázatban láthatók.

α	0.15	0.2	0.25	0.4
M1	0.37	0.75	0.54	0.21

5. Táblázat. A BLX- α keresztezés α paraméterének értékelése (M1-es vizsgálat)

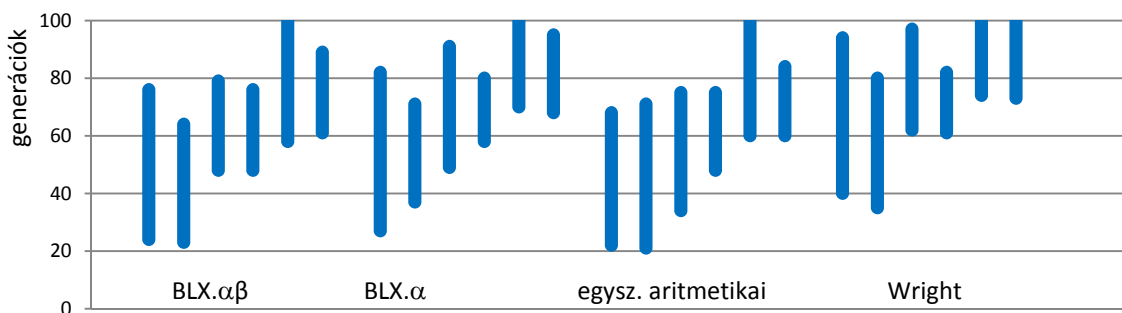
A BLX- α keresztezést, a módszer aszimmetrikus kromoszóma tartományt (a kiválasztás és a szülők tartományainak közepe eltérő) alkalmazó BLX- $\alpha\beta$ változatával összehasonlítva is megvizsgáltam. Az aszimmetria mértékét a β paraméter határozza meg (7.11 fejezet). A vizsgálatban az α -t állandó $\alpha = 0.2$ értéken tartottam, a BLX- $\alpha\beta$ változatait a $\beta = [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$ értékeket alkalmazva hoztam létre. A változatokat a megállási feltétel eléréséhez szükséges generációk számával értékeltem. Az értékelés eredményét a 43. ábra szemlélteti.



43. ábra. A kromoszóma tartomány aszimmetriájának hatása a GA sebességére

Az ábra a két módszer generációszámainak arányát mutatja a β függvényében. A $\beta = 0$ helyen a BLX- $\alpha\beta$ módszer BLX- α -val egyenértékű, a két módszer ugyanazt az eredményt adja. Az aszimmetria növelésével a generációszámok a $\beta = [0 \div 0.4]$ értékeknél először csökkennek, majd gyorsulva nőnek, a $\beta > 1$ érték felett a konvergencia meghiúsult. A vizsgálat alapján a BLX- $\alpha\beta$ módszer β paraméterét $\beta = 0.4$ -re választottam.

A leginkább megfelelő keresztezés meghatározásához a BLX- α és BLX- $\alpha\beta$ módszereket a tőlük teljesen eltérő, egyszerű aritmetikai és a Wright módszerrel (7.11 fejezet) hasonlítottam össze. Az összehasonlításhoz az M3-as vizsgálati módszert alkalmaztam, az eredményeket a 44. ábra szemlélteti. Az értékelésből a Wright módszert kizártam, mivel több megállási feltételt sem tudott a biztonsági megállási feltétel generációszámán belül elérni. A többi módszer generációszámai viszonylag széles tartományban mozognak, ez a leginkább az egyszerű aritmetikai keresztezésre igaz. A BLX- $\alpha\beta$ és a BLX- α eredményeit összehasonlítva, nagyjából ugyanazt a sebességet láthatjuk, azzal hogy az előbbinél a generációszámok legtöbbször kisebb tartományban mozogtak.



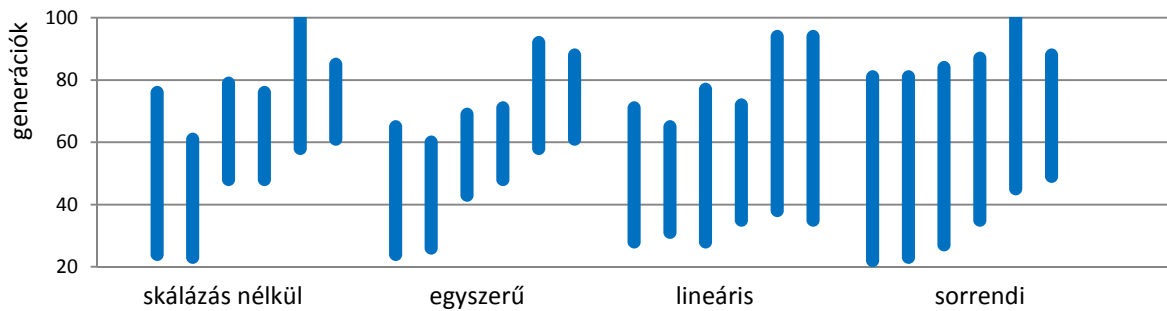
44. ábra. Keresztezési algoritmusok M3 vizsgálatának eredményei

Az M3-as vizsgálatok alapján a GA-ban történő alkalmazásra a BLX- $\alpha\beta$ módszert választottam.

9.5. Skálázás vizsgálata

A skálázás vizsgálatának célja megállapítani, javítható-e a GA teljesítménye, ha a keresztezéshez történő kiválasztás előtt a fitnessz értékeket skálázzuk. A vizsgálatban, a 7.9. fejezetben ismertetett ská-

lázási algoritmusok alkalmazását teszteltem, a GA teljesítményét az M3-as módszerrel értékeltem. A vizsgálat eredményeit a 45. ábra foglalja össze. Az első sávcsoport a skálázás nélküli, az utána következő csoportok sorrendben az egyszerű, lineáris és a sorrendi skálázást alkalmazó GA teljesítményét mutatják.



45. ábra. Skálázási algoritmusok M3 vizsgálatának eredményei

Az ábrából látható, hogy a BLX. $\alpha\beta$ keresztezést alkalmazó GA teljesítményét (44. ábra első sávcsoportja) a megvizsgált skálázási módszerek jelentősen befolyásolják.

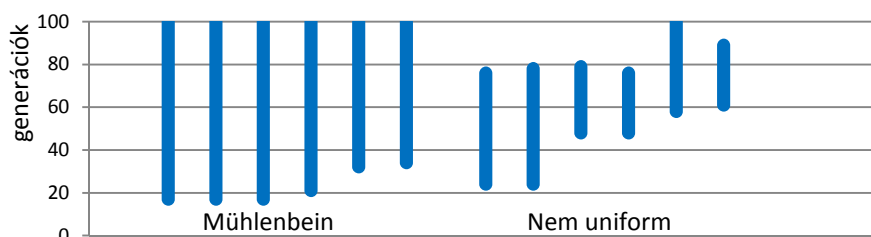
A lineáris és a sorrendi skálázás jelentősen gyengítette az algoritmus megbízhatóságát, mivel a generációs számok tartományai mindegyik megállási feltételnél érezhetően nagyobbak lettek. Ezen felül arra is van példa, hogy a GA nem érte el a megállási feltételt a biztonsági feltétel megszabta generációs számon (100 generáció) belül. Az egyszerű skálázással a GA jobb eredményeket produkált, a megállási feltételek többségénél csökkent a tartományok mérete és egyik esetben sem nőtt jelentősen.

A sebesség szempontjából a GA mindegyik skálázás esetén jobb lett. A lineáris és a sorrendi skálázással a generációs számok középértékei sokkal lejjebb kerültek, és az egyszerű skálázás generációs számainál is tapasztalható csökkenés.

Mindent összevetve az egyszerű skálázás alkalmazását tartom megfelelőnek, mivel nem csökkentette a megbízhatóságot, és a GA sebességén is javított.

9.6. Mutáció vizsgálata

A vizsgálatban a „Mühlenbein” és az alapváltozatban alkalmazott „non uniform” mutációt alkalmazó GA teljesítményét hasonlítottam össze. A vizsgálatához az M3-as módszert alkalmaztam, a kapott eredményeket a 46. ábra foglalja össze.

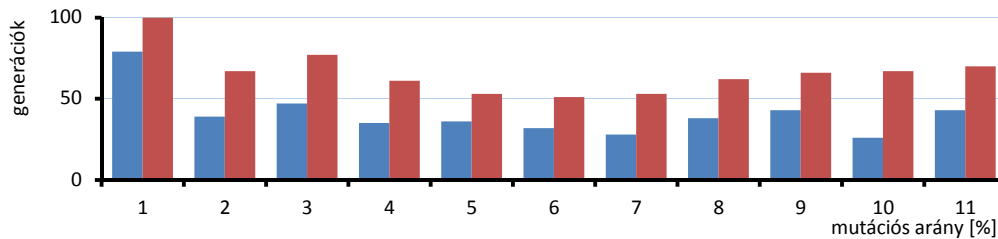


46. ábra. Mühlenbein és a nem uniform mutáció összehasonlítása M3 vizsgálatával

Az ábrából látható, hogy a Mühlenbein mutáció generációtartományai nagyon szélesek. Ezen felül mindegyik megállási feltételnél volt legalább egy felület, ahol a GA nem volt sikeres. A tartományok közepe valamelyest alacsonyabbak, mint a nem uniform mutációnál, de a sikertelenségek miatt ez nem sokat számít. A vizsgálat egyértelműen azt mutatta, hogy a két mutáció közül a nem uniform módszert lehet csak biztonsággal alkalmazni.

9.7. Mutációs arány vizsgálata

A GA alapváltozatában a szakirodalomban javasolt $\mu = 7\%$ -os mutációs arányt alkalmaztam, a vizsgálat célja az arány értékének finomhangolása. A vizsgálathoz az M2-es módszert alkalmaztam, a $\mu = [1..11]\%$, $\mu \in \mathbb{N}$ értékekkel. Az eredményeket az 47. ábra mutatja.



47. ábra. A GA sebessége a mutációs arány függvényében

Látható hogy a vizsgált tartományban a GA mindenhol elérte a megállási feltétel alsó és felső határát is. A GA mindkét megállási feltétel határt az 5 és a 6% között érte el a leggyorsabban, ezen felül a felső határ eléréséhez szükséges többletgeneráció száma is itt volt a legkisebb. A vizsgálat ezen eredményei alapján a mutációs arányt 7-ről 5 és 6% közé, 5.5%-ra csökkentettem.

10. A hibajellemzők hatása a GA működésére

A GA paramétereinek és operátorainak előző fejezetekben ismertetett módon történt meghatározásával olyan GA-t állítottunk elő, amelyik a különböző gyakorlati felületeken megbízhatóan és gyorsan működik. Ez a fejezet arra keresi a választ, hogy a kifejlesztett GA milyen műszaki problémák megoldására és milyen hatékonyan alkalmazható, milyen összefüggés van a hibák és a GA viselkedése között. Ennek meghatározására azt vizsgáljuk, hogy a fényvonalak különböző hibajellemzői: a hiba nagyság, kiterjedés és komplexitás széles határok között történő változtatása hogyan befolyásolja a GA működését.

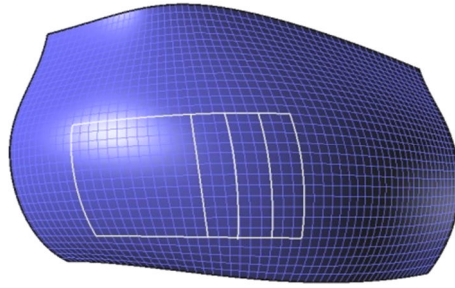
10.1. A vizsgálatok munkamódszere

A vizsgálatokban a hiba nagyságát az NRMSE.H, kiterjedését az AREA.S, komplexitását az INF.H mérőszámokkal mértem. A hibajellemzők hatását egymástól elkülönítve vizsgálom. A hatásuk egymástól történő elkülönítésére az egyes vizsgálatokban csak a vizsgált jellemzőt változtatom, a többi közelítőleg állandó értéken tartom. Jellemzőként négy különböző nagyságú hiba hatását hasonlítottam össze. A hibák nagyságai az árnyékolt megjelenítés érzékenységi határa közelében vannak, vagy az alattiak. A hibák területe a felület méretéhez képest lokális vagy közepesen nagy kiterjedésű (0. fejezetében megfogalmazott problémáknak megfelelő). A hibák komplexitása az egészen egyszerűtől a közepesen bonyolultig terjed. A jellemzők értéktartományait a 6. táblázat tartalmazza.

	NRMSE.H	AREA.S	INF.H
értéktartomány	0.21-0.56	8112-18187	0.80-1.8

6. Táblázat. A hibajellemzők értéktartományai

A vizsgálatokat és a vizsgálatok eredményeit külön erre a célra létrehozott felületen mutatom be. A felület geometriájának jellege, mérete az ipari felületek méretéhez és geometriájához igazodik. A vizsgálatokhoz a hibákat jó felületek kontrollpontjait módosítva hoztam létre. A kontrollpontok struktúrája szabályos és megfelelően sűrű, hogy a hibajellemzők értékeit minél pontosabban és hatékonyabban be tudjam állítani. A példafelület alakját az 48. ábra mutatja. A hibákat tartalmazó felületrészek határait fehér vonalakkal jelöltem, a halványkék vonalak a felület kontrollpont hálóját mutatják.



48. ábra. A vizsgálatok eredményeinek bemutatására szolgáló példafelület

A GA-t a megállási feltételek eléréséhez szükséges generációk számával, illetve a jó és a javított felületek közti távolságokból számolt távolságtérképekkel (MAP) értékeltem.

Megállási feltételek a felület NRMSE.H mutatójának 90%-os javíttósága (N90) és a legkisebb hibát tartalmazó felület NRMSE.H mutatójának 90%-os javíttósága (NM90). Ez utóbbi megállási feltételt arra használtam, hogy megvizsgálhassam mennyivel több generáció szükséges ugyanazon jó felület előállításához, különböző hibajellemző nagyságok esetén. Az eltérést a generációs számok Δ terjedelmével és σ szórásával értékeltem:

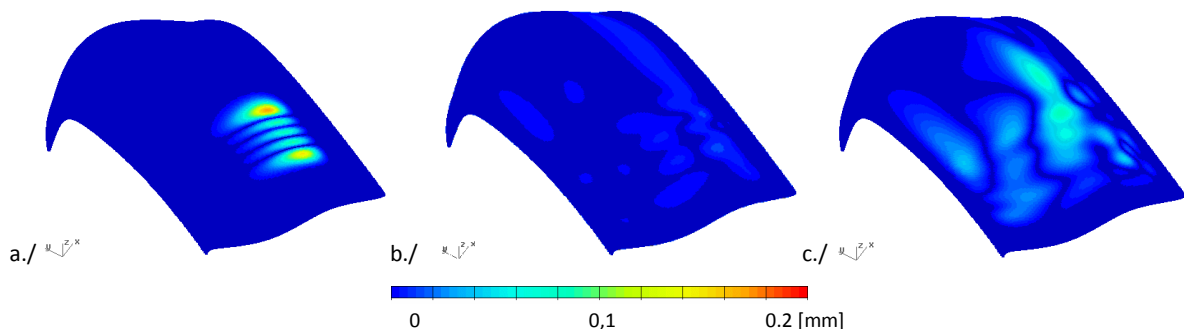
$$\Delta = g_{\max} - g_{\min} \quad , \quad \sigma = \sqrt{\frac{\sum (g - \bar{g})^2}{n - 1}}$$

ahol g NM90 megállási feltételhez tartozó generációs szám
 \bar{g} NM90 megállási feltételhez tartozó generációs számok átlaga
 g_{\max}, g_{\min} NM90 megállási feltételhez tartozó legnagyobb/legkisebb generációs szám
 n GA futtatások száma

A távolságtérképeken ábrázolt távolságok, felületek közötti merőleges távolságok, amelyeket a jó felületen felvett pontok és a másik (a hibás vagy a javított) felület között számítok. A pontokat rendeztem, adott számban, paraméterirányonként veszem fel, a vizsgálatokban irányonként 250 pontot alkalmaztam. A térképek a távolságokat a pontokhoz rendelt pixel színével jelenítik meg. A kutatásban alkalmazott színskála 42 színt különböztet meg, a hidegebb színek kisebb, a melegebbek nagyobb távolságot mutatnak.

A felületek közti távolságtérképekkel vizuálisan értékelhetők a javítások. A MAP térképek a hiba és a javítás értékelésének eszközei, amelyek segítségével részletesebb kép alkotható a megoldandó feladról és a megoldás eredményességéről. Ellenőrizhető, hogy a hibajellemzők méretétől függetlenül a megoldásfelületek mindenhol a felületen ugyanoda és a jó megoldáshoz konvergáltak-e.

A vizsgálatokhoz a következő távolságtérképeket alkalmaztam: jó és a hibás (MAP.1), jó és a javított (MAP.2) illetve javított (MAP.3) felületek közötti távolságtérkép. A 49. ábra a példafelületen mutatja be a térképeket.



49. ábra. Példa a távolságtérképekre a./ MAP.1, b./ MAP.2 és c./ MAP.3

A MAP.1 térképből a megoldandó feladat jellege olvasható ki. A képen jól kivehető a hiba helye, kiterjedése, nagyságának változásából formája, a legnagyobb hiba helye és formája illetve az, hogyan csökken a hiba a környező jó felületrészbe nullára.

A MAP.2 térkép a jó felület és a javított felületek közti legnagyobb távolságokat mutatja. Tanulmányozásával eldönthető mennyivel és hogyan térnek el a javított felületek a jó felülettől, a javított felületek nem csak a fényvonalak mentén, hanem mindenhol közel vannak-e a jó felülethez. A térképről látható hogy a javítás következtében a hiba körüli hibátlan felületrészek alakja is megváltozik, fontos ellenőrizni, hogy a hibátlan felületrészeket a javítás ne rontsa el, és a módosulás a megengedett érték alatt maradjon.

A MAP.3 térképek a javított felületek közötti legnagyobb távolságokat mutatják. A térképpel az vizsgálható hogy a felületek a hibajellemzők nagyságától függetlenül ugyanoda konvergálnak-e.

10.2. Az algoritmus robusztusságának vizsgálata

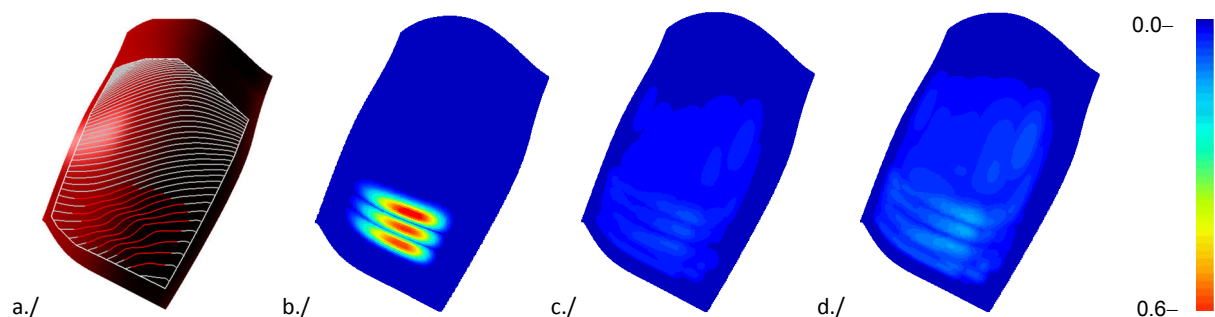
A genetikus algoritmus működése nem determinisztikus, számos sztochasztikus eleme (kezdeti populáció létrehozása, keresztezés, mutáció, stb.) van. Ennek az egyik következménye, hogy a GA futtatásonként ugyanazt a problémát eltérő sebességgel oldja meg, és hogy ugyanarra a problémára egymástól valamelyest eltérő megoldásokat talál meg. A vizsgálat célja meghatározni, hogy a kifejlesztett GA mennyire robusztus: sebessége és megoldásai mennyivel térnek el egymástól, ugyanannál a feladatnál. Ennek a megállapítására a GA futtatását egy feladaton többször megismételtem, majd megvizsgáltam a javított felületek egyezőségét illetve összehasonlítottam a keresések sebességét. A GA viselkedését akkor tekintettem robusztusnak, ha az eltérések mindkét szempont szerint elhanyagolhatóak voltak.

A felületek (megoldások) egyezőségét MAP távolságtérképekkel vizsgáltam, a megállási feltétel eléréséhez szükséges generációs számok közti eltérést a generációs számok Δ terjedelmével és σ szórásával mértem.

Az alkalmazott megállási feltétel: a felület NRMSE.H mutatójának 90%-os javítottsága (N90). A vizsgálatban a legnagyobb hibaméterű és kiterjedésű hibát tartalmazó felületeket javítottam (6. táblázat).

A példafelületet az 50. ábra felülnézetből mutatja, hogy a fényvonalak közötti eltérések jól látszódnak és a távolságtérképek könnyebben értékelhetők legyenek. Az 50.a ábra a felületet árnyékoltan megjelenítve és a fényvonalával mutatja. A hiba az árnyékolt megjelenítéssel nem fedezhető fel, a fényvonalak azonban kimutatják (élénkpiros színű fényvonalak). A javításban érintett felületrész hátárát és a többi érintett fényvonalat fehérrel jelöltem.

A 50.b ábra a felület MAP.1 távolságtérképét mutatja, láthatjuk a hiba helyzetét, és hogy a hiba három, egymástól elkülönülő, keresztirányba elnyúló hibacsúcs köré összpontosul. A legnagyobb távolság a hibás és a jó felület között 0.6 mm.

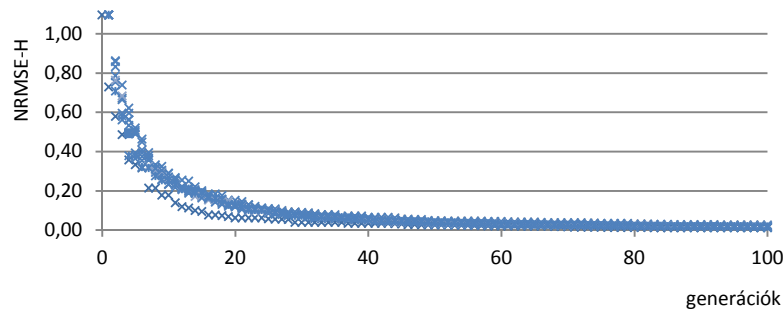


50. ábra. A hibás felület fényvonalai és a felület MAP.1, MAP.2, és MAP.3 távolságtérképei

A felület javítását kilenc alkalommal ismételt meg. Az 52c. ábrán a vizsgálat MAP.2 térképe látható, ami a jó és a javított felületek közötti legnagyobb távolságokat mutatja. A távolságok sokkal kisebbek, mint a későbbi hibajellemző nagyság vizsgálatokban kapott távolságok, ami azt mutatja, hogy az algoritmus nagy biztonsággal mindig ugyanazt a megoldást találta meg.

Az 50d. ábrán a vizsgálat MAP.3 térképe látható, a térkép a javított felületek közötti legnagyobb távolságokat mutatja. A távolságok nagyságát és eloszlását tanulmányozva látható, hogy a távolságok a javított hibacsúcsok környezetében valamennyivel nagyobbak. Ebből arra következtettem, hogy a GA robusztusságát a javított hiba nagysága befolyásolja ugyan, de ennél a hibaméretnél ez még elfogadható mértékű. Ez utóbbit az bizonyítja, hogy az érintett felület többi részén a különbségek kisebbek, és homogén eloszlásúak.

A 51.a ábra a megismételt keresések NRMSE.H mutatóit ábrázolja a generációk függvényében. Látható hogy mindegyik esetben nagyon hasonlóan változik a keresés során. A genetikus folyamat elején mutatkozó különbségek az első generáció magas kromoszóma-változatosságából adódnak, ami azonban a keresés előrehaladtával gyorsan lecsökken. A generációszámok eltérése: $\Delta = 8$, $\sigma = 3.7$.

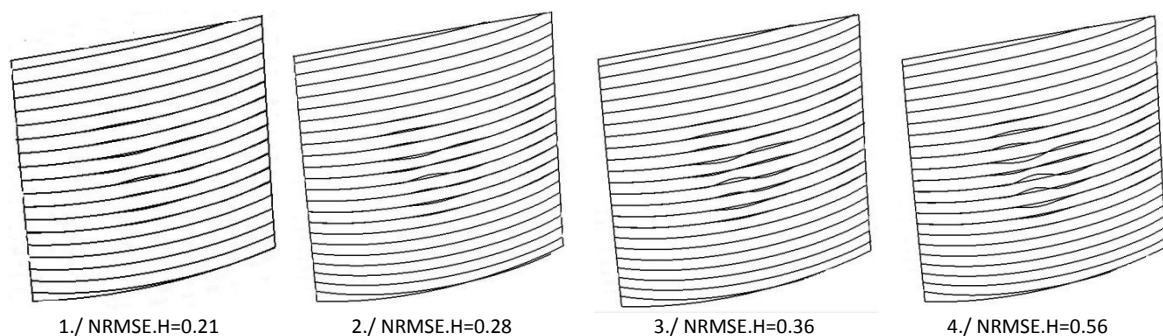


51. ábra. Az NRMSE.H mutató alakulása a keresések során (robusztusság vizsgálat)

A vizsgálatok alapján megállapítható hogy a GA viselkedése robusztus. A hiba mindegyik esetben hasonló módon csökkent, a megállási feltétel generációszámának terjedelme, szórása alacsony. Az eredményfelületeket megvizsgálva az látható, hogy mindegyik keresés gyakorlatilag ugyanazokat a felületeket produkálta.

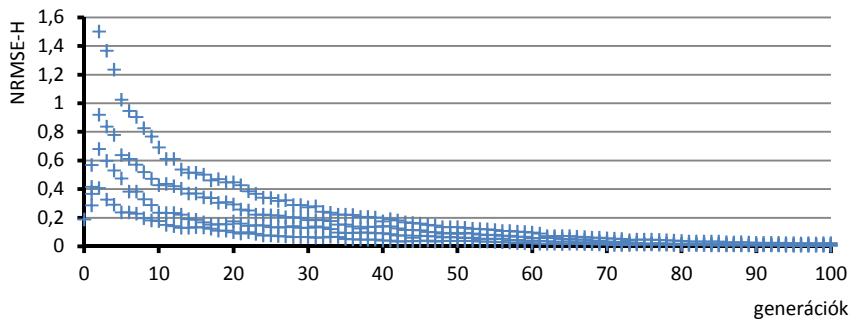
10.3. Hibanagyság hatása

A hibanagyság hatását a 6. táblázatban megadott $NRMSE.H = (0.21-0.56)$ értéktartományban vizsgáltam, legnagyobb hiba kiterjedés ($AREA = 18187mm^2$) és komplexitás ($INF.H = 1.8$) mellett. Azért választottam a másik két hibajellemző értékét a legnagyobbra, mivel így vizsgálható a leginkább kedvezőtlen körülmények között a hibanagyság. Az 52. ábra az érintett felületrészek hibás és jó fényvonalait egyszerre mutatják. A legnagyobb hiba kevéssel az árnyékolt reprezentáció érzékenységi határa alatt van, a legkisebb hiba ennek a fele.



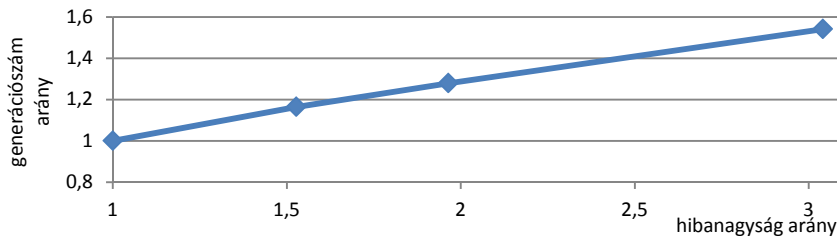
52. ábra. Hibanagyság vizsgálat jó és hibás fényvonalai

A vizsgálatokban először azt értékeltem, hogyan változik a különböző hibák NRMSE.H értéke a genetikus keresés során. Az 53. ábrán az NRMSE.H értékeket közös diagramban ábrázoltam. Látható, hogy a kezdeti nagy különbségek ellenére az értékek gyorsan csökkennek, majd kiegyenlítődnek és közel ugyanoda konvergálnak. A megállási feltétel mindegyik hibanagyság esetén a legkisebb hiba 90%-os javított-ságához tartozó érték ($NMH=0.018$). A megállási feltételhez tartozó generációszámokat a GA hibamérettől függetlenül mindegyik esetben közel ugyanolyan gyorsan érte el. Eltérések: $\Delta = 33$, $\sigma = 13.9$.



53. ábra. Az NRMSE.H alakulása a keresések során (hibanagyság vizsgálat)

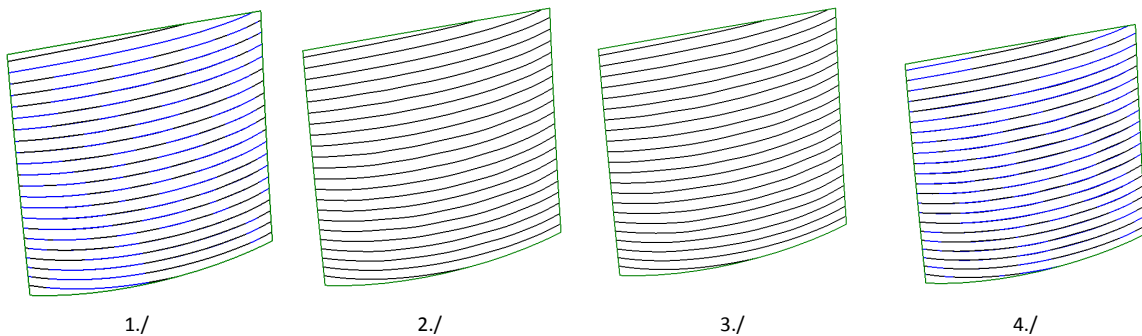
Az 54. ábra az NMH90 megállási feltétel eléréséhez szükséges generációszámokat mutatja a hibaméret függvényében. A tengelyeken a generációszámok és a hibák arányai szerepelnek.



54. ábra. A megállási feltétel eléréséhez szükséges generációk száma a hibanagyság függvényében.

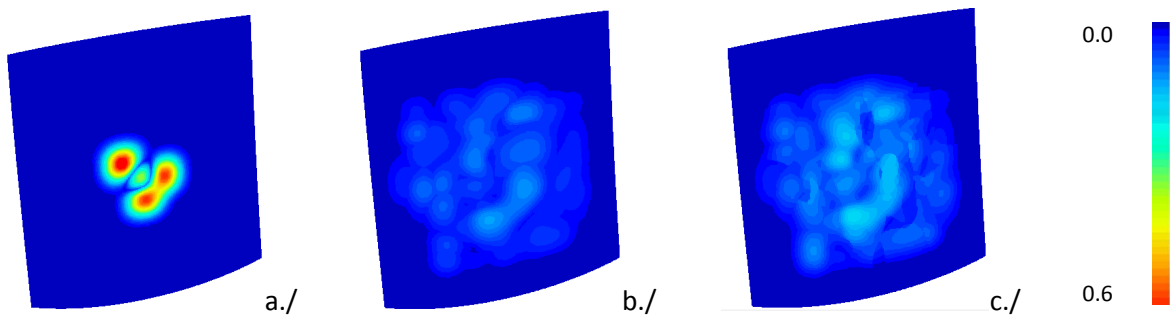
A diagramból látható, hogy a generációk száma a hibanagyság növekedésével arányosan nő. A legkisebb és a legnagyobb hiba javítását összehasonlítva azt látjuk, hogy a háromszor akkora hiba javításához is csak másfélszer több generációszámra volt szükség.

Az 55. ábra az érintett felületrész javított (feketével jelölve) és a jó (kézzel jelölve) fényvonalait egyszerre mutatja. A javított fényvonalak helyzetét és alakját vizuálisan megvizsgálva láthatjuk, hogy mindenhol közel vannak a jó fényvonalakhoz és alakjuk gyakorlatilag megegyezik a jó fényvonalak alakjával.



55. ábra. Hibanagyság vizsgálat jó és javított fényvonalai

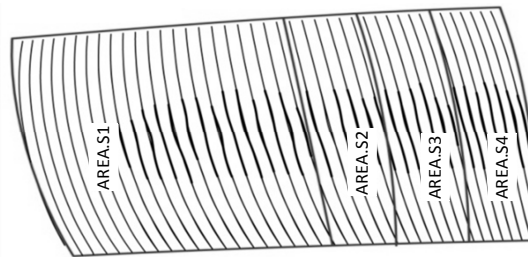
A javítások távolságtérképeit az 56. ábra mutatja. A MAP.1 térképen (56.a ábra) látható hogy az eredeti hiba viszonylag kis területet foglal el, és több egymás melletti hibacsúcsból áll. A MAP.2 térkép (56.b ábra) azt mutatja, hogy a javított felületeken a hibacsúcsok kisimulnak, és a fennmaradó hibák egyenletesen, nagyobb területen oszlanak el (56.b ábra), a javított felületek mindenhol közel vannak a jó felülethez, és a közöttük lévő távolság a felületrész peremének irányába fokozatosan nullára csökken. A javított felületek egymás közötti távolságát megvizsgálva az látható, hogy szintén hasonló módon közel vannak egymáshoz (56.c ábra).



56. ábra. A hibanagyság vizsgálat távolságtérképei

10.4. Hibakiterjedés hatása

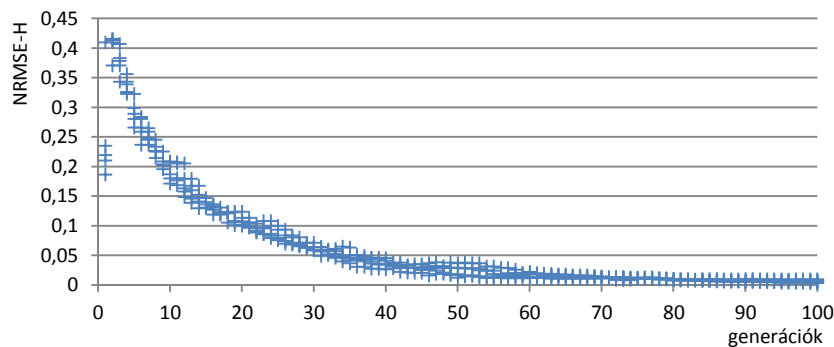
A hiba kiterjedtségének hatását az 6. táblázatban megadott $AREA.S = (8112-18187\text{mm}^2)$ értéktartományban vizsgáltam, legnagyobb hiba méret ($NRMSE.H = 0.56 \text{ mm}$) és komplexitás ($INF.H = 1.8$) mellett. Azért választottam a másik két hibajellemző értékét a legnagyobbra, mivel így a hiba kiterjedése a leginkább kedvezőtlen körülmények között vizsgálható. Az 57. ábrán az érintett felületrészek (vastag szürke görbékkel határolva) jó és hibás fényvonalai láthatók. A legkisebb ($AREA.S1$) és a legnagyobb ($AREA.S4$) felület közötti különbség 175%, a másik két hibajellemző változása 10% alatti.



1./ $AREA.S1=8112 \text{ mm}^2$ 2./ $AREA.S2=10127 \text{ mm}^2$ 3./ $AREA.S3=12142$ 4./ $AREA.S4=18187 \text{ mm}^2$

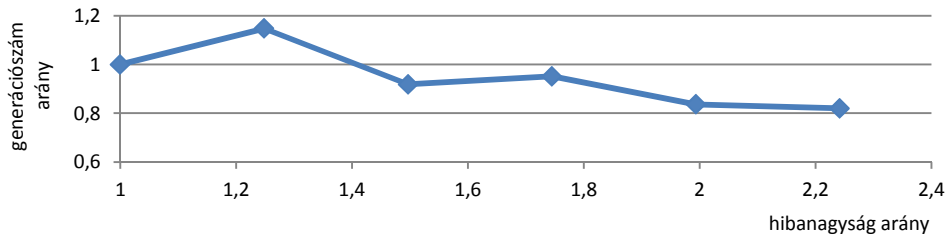
57. ábra. A Hibakiterjedés vizsgálat jó és hibás fényvonalai

Az 58. ábra az $NRMSE.H$ mutató alakulását mutatja genetikus keresés során. Látható, hogy értékei még gyorsabban váltak hasonlóvá, mint a hibanagyság vizsgálatban. Ez annak a következménye, hogy a kezdeti értékek, mivel a hibanagyság állandó, szinte ugyanazok. A megállási feltételt ($NMH=0.018$) az algoritmus kis szórással ($\sigma = 7.3$), kis terjedelemmel ($\Delta = 11$) mindegyik esetben ugyanakkor érte el.



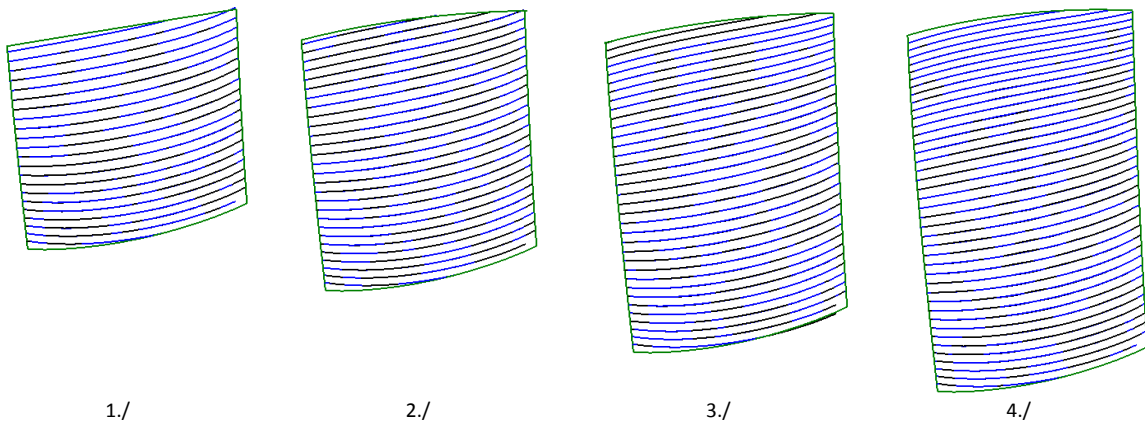
58. ábra. Az $NRMSE.H$ alakulása a keresések során (hiba kiterjedés vizsgálat)

A 59. ábra az $NMH90$ megállási feltételhez tartozó generációk számát a hiba területének függvényében ábrázolja. Látható hogy a terület növelésével a generációszám először nő, majd folyamatosan csökken. Ha a legkisebb és a legnagyobb területű hiba javításához szükséges generációk számát összehasonlítjuk, az látható, hogy míg a terület nagysága 220%-al nőtt a generációk száma 20%-al csökkent.



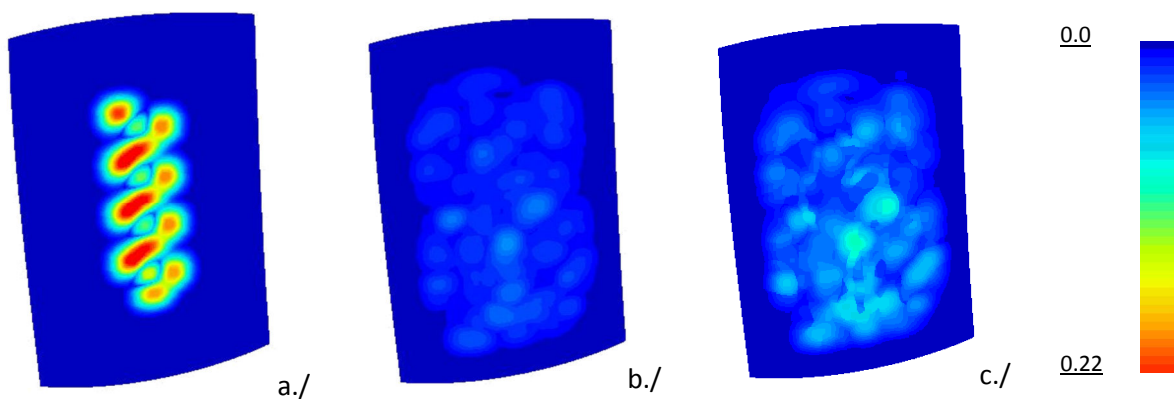
59. ábra. A megállási feltétel eléréséhez szükséges generációk száma a hiba kiterjedésének függvényében

A javított és a jó fényvonalakat összehasonlítva (60. ábra) látható hogy a GA kis eltéréssel a jó fényvonalakat állította elő.



60. ábra. A Hibakiterjedés vizsgálat jó és javított fényvonalai

A felületek távolságtérképeit megvizsgálva látható, hogy a hibás felületrészekon viszonylag egyenetlen a hibaeloszlás, a hibás felületrészeket együtt ábrázolva, hibacsúcsok sorozata rajzolódik ki (61.a ábra). A javítás után a csúcsok kisimulnak, és a visszamaradó hibák közel egyenletesen oszlanak el, illetve a javított felületrész peremének irányába fokozatosan nullára csökkennek (61.b és 61.c ábra). A javított felületek MAP.2 illetve MAP.3 térképei azt mutatják, hogy az eredményként kapott felületek mindegyike közel van a jó felülethez és a közöttük lévő eltérések elhanyagolhatók.

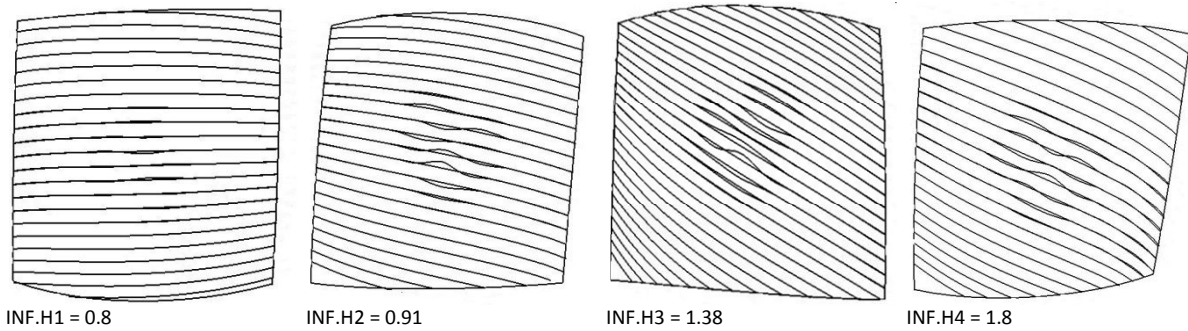


61. ábra. A hibakiterjedés vizsgálat távolságtérképei

10.5. Fényvonal-komplexitás hatása

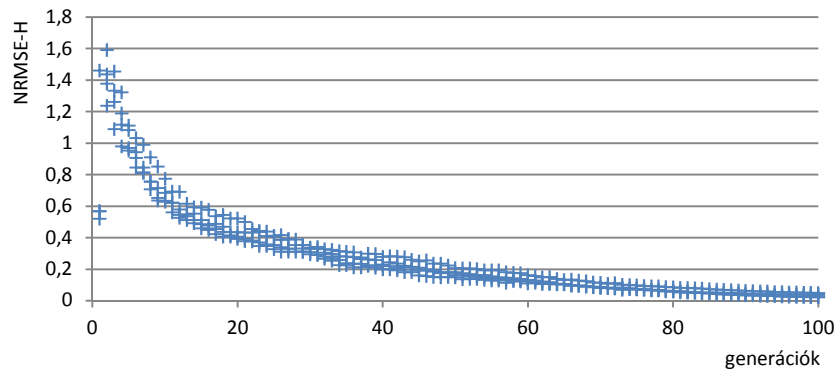
A komplexitásának hatását az 6. táblázatban megadott $INF.H = (0.8-1.8)$ értéktartományban vizsgáltam, legnagyobb hiba méret ($NRMSE.H = 0.56mm$) és kiterjedés ($AREA.S = 18187mm^2$) mellett. A különböző komplexitású fényvonal struktúrákat a fényforrások elforgatásával állítottam elő. Ezt a módszert azért tartottam jónak, mert egyrészt nem változtatja meg a hiba nagyságát és kiterjedését,

másrészt mivel a fényvonal struktúra erősen függ a felület és a fényforrások kölcsönös helyzetétől, a fényforrások helyzetének kis változtatása a fényvonal struktúrában jelentős változást okoz. A 62. ábra az így előállított fényvonal struktúrákat szemlélteti.



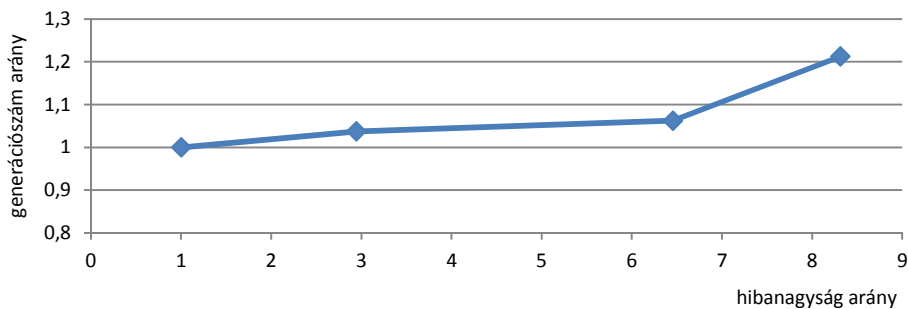
62. ábra. A komplexitás vizsgálat jó és hibás fényvonalai

A felületek NRMSE.H mutatói a genetikus keresés során végig nagyon közel haladnak egymáshoz (63.a ábra), és a javításhoz szükséges generációs számok eltérései is alacsonyak: $\Delta = 17$, $\sigma = 7.5$.



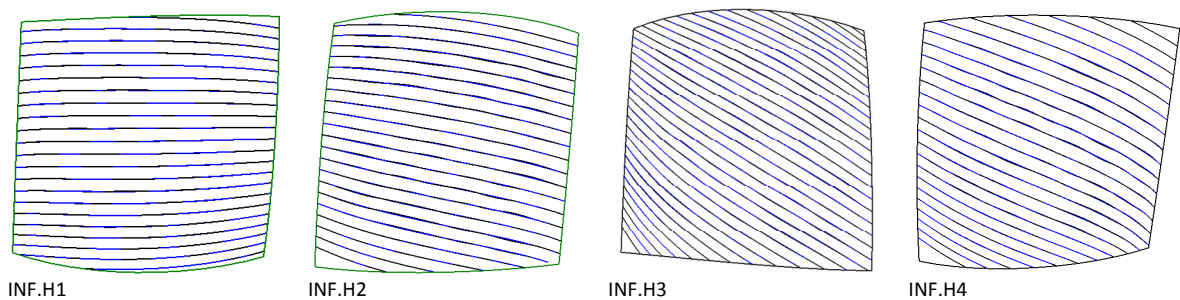
63. ábra. Az NRMSE.H alakulása a keresések során (komplexitás vizsgálat)

A 64. ábra az NMH90 megállási feltételhez tartozó generációk számát a hiba komplexitásának függvényében ábrázolja. Látható, hogy a komplexitás növekedésével a generációs szám alig változik. A legnagyobb és a legkisebb komplexitású hiba javítását összehasonlítva a hétszeres komplexitás is csak 20%-os generációs szám növekedést eredményezett.



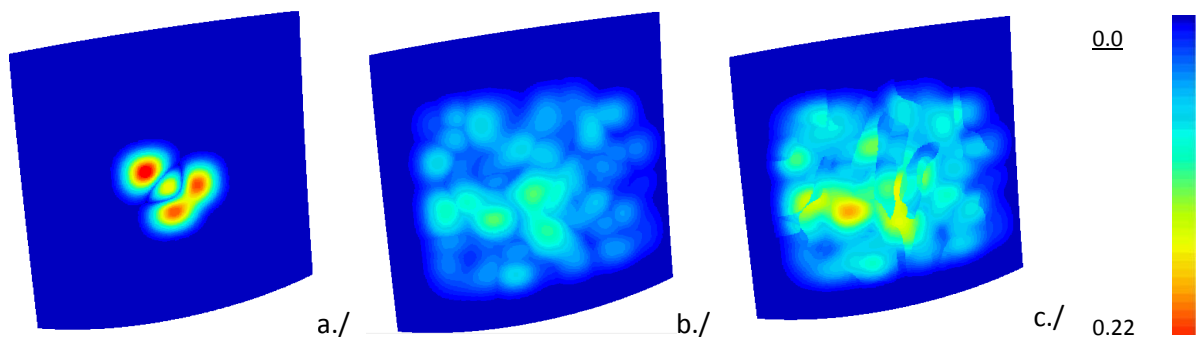
64. ábra. A megállási feltétel eléréséhez szükséges generációk száma a hiba komplexitás függvényében

A javított fényvonalakat a jókkal összehasonlítva azonban azt láthatjuk, hogy az előző vizsgálatokban kapott eredményekhez képest csak közepes a fényvonalak minősége (65. ábra).



65. ábra. A komplexitás vizsgálat jó és javított fényvonalai

A felületek távolságtérképeit (66. ábra) megvizsgálva látható, hogy a hiba csúcsai (66.a ábra) a javítás után kisebbek lettek ugyan, de nem tűntek el, sőt megszorodtak (66.b ábra). Emiatt a felület egyetlen maradt, ahogy az a fényvonalak alakjából már sejthető is volt. Ha a megoldások egymástól való távolságát nézzük (66.c ábra) a maradék hibák még inkább kifejezettebbek.



66. ábra. A komplexitás vizsgálat távolságtérképei

10.6. Összefoglalás

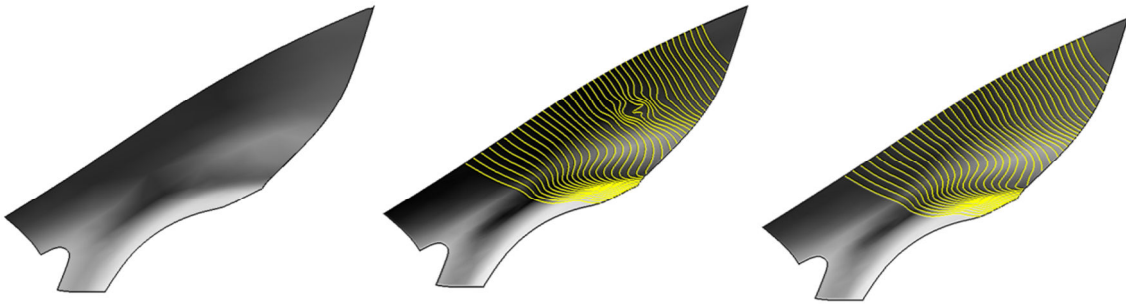
Összességében elmondható hogy a vizsgálatokban a javított felületek a hibajellemzők nagyságától függetlenül ugyanoda, a jó felülethez konvergáltak. A javított felületek mindenhol közel voltak egymáshoz és a jó felülethez is. A javítás eredményeként a hibacsúcsok kisimultak és a maradék hibák a felületrész peremének irányába fokozatosan nullára csökkentek.

A hibaméret vizsgálatánál azt láthattuk, hogy javításhoz szükséges generációk száma a hiba nagyságával érezhetően nőtt. A hiba kiterjedés vizsgálatnál ezzel szemben a kiterjedés növelésével a szükséges generációk száma alig változott, sőt csökkent. A komplexitás vizsgálatban a fényvonalak összehasonlítása és a távolságtérképek is a vártnál nagyobb maradék hibákat mutattak a javított felületeken. Ez a fényvonalak csökkent érzékenységevel magyarázható, ami annak az eredménye, hogy a fényforrások síkja és a fényvonal-pontok felületmerőlegesei túl kis szöget zárnak be egymással (a fényvonalak alakja ilyenkor „megnyúlik”). A fényvonalak akkor mutatják a legérzékenyebben a hibát, ha az említett szög a legtöbb fényvonal-pontban közel merőleges. A felhasználó olyan pozícióba kell hogy helyezze a fényforrásokat, ahol ez megvalósulhat.

11. Alkalmazási példák

A kifejlesztett GA alkalmazhatóságát számos felület javításán próbáltam ki, és igazoltam. A felületek ipari termékek formatervezett felületei, amelyek jelentősen eltérnek egymástól geometriai bonyolultság és funkció szempontjából, hibajellemzőik széles határok között mozognak.

A fejezet a hibás, és a hibajavítás eredményeként előállt felületeket mutatja be, ábrák segítségével (**Hiba! A hivatkozási forrás nem található.** melléklet). Az ábrák a felületeket árnyékoltan és fényvonalakkal együtt mutatják, a 67. ábra a melléklet ábrái közül mutat egy példát (autókarosszéria sárvédő eleme).



67. ábra. Sárvédő elem javítás előtt (a., b.) és után (c.)

A 67.a és 67.b ábrákon a hibás felület látható árnyékoltan, illetve árnyékoltan a fényvonalakkal együtt. A 67.c ábra a javított felületet ábrázolja árnyékoltan, a fényvonalakkal együtt.

Az alkalmazási példafelületeket úgy választottam ki, hogy minél jobban reprezentálják a GA szándékolt felhasználási területét.

A hibák nagysága a finom felületi hibák tartományában van, az árnyékolt felületeken a hibák nem, vagy csak nehezen felfedezhetők fel. Kivételek a 87., 88., és 89. ábra felületei, ahol a hibák olyan nagyok, hogy az árnyékolt felületen is láthatók. A fényvonalak ezzel szemben mindegyik esetben jól ki tudták mutatni a hibákat.

A hibák különböző fényvonal eloszlást produkáló felületeken helyezkednek el. A 79,80,85-88,90 ábrák felületein a hiba egymáshoz közel állandó távolságra lévő fényvonalakon, a 78, 81-84,89 ábrák felületein egymástól lassan és folyamatosan változó távolságra lévő fényvonalakon helyezkedik el.

A hibák kiterjedése jelentősen változik, és a kisebbektől (pl. 85. ábra) a felület nagyobb részét is érintő méretűekig (80., 84. és 86. ábrák) terjed.

A hibás fényvonalak összetettsége a felületeken az egészen egyszerűtől (pl. 80) az összetettebbekig terjed, a leginkább összetett fényvonalak a 83. és a 87. ábrán láthatók.

A hibás és a javított felületek fényvonalait mutató ábrákat összehasonlítva, vizuálisan értékelhető a felületjavítás eredményessége. Látható hogy a javítással kivétel nélkül mindegyik felületnél jelentősen csökkent, vagy megszűnt a hiba.

12. Összegzés (tézisek)

A magas minőségű, szabadformájú felületmodellek lokális hibáinak javítása nem teljesen megoldott, aminek a fő oka, hogy a jelenlegi javító eljárások megelégednek az elemien kis kiterjedésű és méretű lokális hibák javításával. A hibák mérete, kiterjedése és összetettsége (a hiba jellemzői) ezzel ellentétben a mai felületeken sokszor széles határok között mozog. Ennek az ellentétnek az a következménye, hogy az eljárások alkalmazása nehézkes, hosszadalmas, a tervező sokszor inkább manuálisan javít.

A kutatás célja olyan lokális hibák javítására alkalmas módszer kidolgozása volt, amely széles határok között változó hibajellemzők esetén is jól alkalmazható.

Az eljárásban a geometria leírásához a CAD rendszerekben széleskörűen alkalmazott NURBS (Non-uniform Rational B-Spline) reprezentációt alkalmaztam. A NURBS a B-Spline reprezentáció általános formája, a CAD rendszerekben jelenleg alkalmazott geometriai reprezentációs módszerek közül ez adja a legtöbb alakmódosítási lehetőséget, különösen alkalmas a szabadformájú geometria leírására.

A NURBS reprezentáció szakaszos polinomiális függvényt alkalmaz a geometria reprezentálásához, amit számos paraméter határoz meg. A geometria alakjára a polinom együtthatói, amelyek térbeli pontok (kontrollpontok), hatása a legnagyobb. A kidolgozott eljárásban a javításhoz szükséges geometriai módosításokat kizárólag a kontrollpontok változtatásával értem el.

Megvizsgáltam a lokális felületi hibák feltárására alkalmas felületértékelési módszereket. A vizsgálat eredményeként a fényvonalak módszerét (highlight lines) találtam a leginkább megfelelőnek, amely több módszer előnyét egyesíti. A fényvonalak nagyban hasonlítanak az iparban alkalmazott felületminőség értékelő laborok fizikai reflexiós vonalaira, kezelésük a számítógépes környezetben hatékonyabb, mint a reflexiós vonalaké mivel olcsóbb a számításuk, alakjuk nem függ a nézőponttól, a tervezés során szétválaszthatók a nézeti és a fényvonal kezelési műveletek.

Megvizsgáltam az értékelési vonalak, közöttük a fényvonalak alkalmazhatóságát a felületek javításában. A lehetséges két javítási módszer közül (direkt és indirekt) a direkt módszert találtam alkalmasnak. A direkt módszerben az értékelési vonalakat javítják, majd a felületet a javított vonalaknak megfelelően módosítják. Ez azért előnyös, mivel közvetlenül befolyásolhatók a vonalak formája és eloszlása (struktúrája), amelyek alapján a mindenkori felület minőségét eldöntjük. Az indirekt módszerrel a felületértékelő vonalak alakja csak közvetetten javítható, a felhasználó nem tudja pontosan milyen vonalakat kap eredményül, a vonalak alakja jelentősen eltérhetnek attól, amit a tervező látni szeretne. Ez a módszer inkább akkor használható jól, amikor a felület geometriájában még nagyobb változtatások is elképzelhetők.

Ezt követően a direkt módszert alkalmazó felületjavító eljárásokat vizsgáltam meg, és feltártam a hibáikat. A következőkben az eljárások hibáit és a hibák kiküszöbölésére tett javaslataimat illetve a hibák kiküszöbölésére kidolgozott eljárásokat ismertetem.

A javító eljárások nem foglalkoznak az értékelési vonalak (fényvonalak) számításának pontosságával. Könnyű belátni ugyanakkor, hogy a hibák feltárásának és a javításának eredményességéhez hozzá tartozik a fényvonalak pontossága. Munkám során a fényvonalakat nagy pontossággal előállító eljárást dolgoztam ki. Az eljárásban először fényvonal-pontokat határozok meg, a pontok meghatározása egy közelítő és egy pontosító lépésből áll. A közelítő lépésben a felületet u és v paramétervonalai mentén értékelve behatárolom a fényvonal-pont helyének intervallumát. A pontosító lépés az intervallumon belül keres, és a felhasználó által megadott tűrésnek megfelelően meghatározza a pont helyét. Ezt követően a fényvonal folytonos reprezentációjához a pontokra B-Spline görbét illesztnek.

A módszer robusztus és bármilyen, CAD rendszerekben használatos matematikai felületreprezentáció esetén alkalmazható. A módszer árnyékolási módszerrel még nem felfedezhető finom felületi hibák esetén is megbízhatóan működik, alkalmazhatóságát számos ipari felületen vizsgáltam és igazoltam.

A direkt módszerekben a hibás fényvonal szakaszok végpontjainak kijelölése a felhasználó aktív bevonásával, interaktívan történik. A jelenlegi eljárások ugyanakkor nem nyújtanak támpontot a végpontok megfelelően pontos kijelöléséhez. A pontatlan kijelölés nagyban befolyásolhatja a javítás eredményét mivel így a javításból hibás fényvonalak maradhatnak ki, vagy a javítás jó fényvonalszakaszokat is módosít.

Az általam javasolt végpont kijelölési művelet két részműveletből áll. Az első részműveletben a felhasználó a felületen interaktívan körberajzolja a hibás fényvonalakat tartalmazó felületrészt, a másodikban pedig szintén interaktívan, pontosítja. A felületrész körberajzolása felületen futó görbével, a CAD rendszer szolgáltatásait igénybe véve történik. A művelet során meghatározom a görbe és a fényvonalak metszéspontjait. A következő, pontosító lépésben a metszéspontok közötti fényvonalszakaszon adott sűrűséggel meghatározom a fényvonalak görbületét és görbületi fésűként megjelenítem. A jó és a hibás fényvonalszakaszok görbületei lényegesen eltérnek egymástól. Amíg a jó fényvonalszakaszokon a görbületi fésű szinte belesimul a fényvonal-görbébe a hibás szakaszokon jelentős ingadozást látható. A pontosító lépésben a felhasználó ezen eltérések alapján pontosítja az előzetes kijelöléssel meghatározott pontokat.

Az eljárások a fényvonalakat egyenként javítják. Ez a megközelítés helytelen, mivel a fényvonalak nem javíthatók egymástól függetlenül, a javítás a szomszédos fényvonalak alakjára is kihat, a javítás időigényes, sok tapasztalatot igényel. Az egyenkénti javítás helyett olyan módszert javasolok, amely az egymást követő hibás fényvonalak eloszlása (mintázata) és alakja alkotta fényvonal struktúráját javítja.

A javasolt eljárás a fényvonal struktúra hibáinak feltárására az egymást követő hibás fényvonalakon pontsorozatokat definiál, majd a pontsorozatokon távolság és szögfüggvényeket hoz létre. A következő, javító lépésben először kisimítja a függvényeket, majd a simított függvényértékekből a javított fényvonalakhoz új pontokat számol. Ezt követően a hibás szakaszok végpontjait és az új pontokat felhasználva létrehozza a javított fényvonalszakaszokat.

A javított fényvonalaknak megfelelő felület előállítására problematikus, mivel a felület paraméterei és a fényvonal közötti összefüggés sokváltozós, és nemlineáris. A probléma megoldásához a jelenlegi eljárások a nemlinearitást sok egyszerűsítést alkalmazva kezelik, ami korlátozza az alkalmazhatóságukat.

A felület előállítására kidolgozott eljárásban a nemlinearitás kezelésére genetikus algoritmust (GA) alkalmazok. A GA paramétereit többlépéses és több szempontot figyelembe vevő vizsgálatok és kísérletek alapján határoztam meg. A kifejlesztés során először a felület génreprezentációját, és a megállási feltételt határoztam meg. A kódolás módját a génreprezentáció alapján választottam ki. Ezt követően változatokat dolgoztam ki a fitness számítására, és a szakirodalom alapján kijelöltem a felhasználható operátorokat (skálázás, kiválasztás, keresztezés és mutáció). Összeállítottam egy elsődleges, működő algoritmust majd az algoritmust finomhangoltam. Ebben a műveletben a paramétereket és az operátorokat szisztematikusan változtattam és az így előálló GA-k teljesítményét (sebességét és robusztusságát) értékeltem. Az értékeléshez saját, a feladathoz illeszkedő mérőszámokat alkalmaztam. A GA-t a kifejlesztése során különböző nagyságú és kiterjedésű hibát tartalmazó felületeken teszteltem.

A fényvonalakat számító, javító és felületet javított fényvonalakhoz igazító eljárásokból új felületjavítási eljárást dolgoztam ki. Az eljárás a kis és a nagyobb kiterjedésű lokális hibákat is jó hatékonysággal kezeli, a felületi hiba nagyságára mérsékelten érzékeny. Robosztus, a felületek CAD reprezentációjától független, nagymértékben automatizált a tervezői igényekhez igazítható. A módszert felülettervező rendszerbe illesztettem. Alkalmazhatóságát egymástól jelentősen eltérő geometriai bonyolultságú és funkciójú formatervezett ipari felületeken vizsgáltam és igazoltam.

1. Tézis

Új számítási eljárást dolgoztam ki a fényvonalak előállítására. Az eljárásban a fényvonal-pontok és a fényvonalakat reprezentáló görbék pontosságát a felhasználó közvetlenül írhatja elő.

Az eljárásban először létrehozom a fényforrásokat reprezentáló egyeneseket, majd a felületet a paramétervonalak mentén felosztom és a felosztásnak megfelelően felületi pontokat és a pontokhoz tartozó felületi normálisokat számolok. Ezt követően a felületi normálisok és a fényforrás-egyenesek közti merőleges távolságokból távolságfüggvények diszkrét pontjait számolom. A fényvonal-pontokat a távolságfüggvények zérushelyei határozzák meg, ugyanakkor a zérushelyek csak közelítőleg határozhatók meg, mivel a távolságfüggvények csak véges számú helyen ismertek. A pontosításhoz behatárolom a zérushelyek szomszédos diszkrét függvényértékeit, majd a felhasználó által előírt értékkel pontosítom a zéruspont helyét. A következő lépésben a zérushelyekből meghatározom a felületen a fényvonal-pontokat, majd a fényvonalak folytonos reprezentációjához a pontokra B-Spline görbét illesztetek. A görbe a felhasználó által előírt pontossággal közelíti (approximálja) a fényvonal-pontokat.

Az eljárás az eddig használt módszerekhez képest több előnyös tulajdonsággal rendelkezik. Egyrészt a pontosság növeléséhez nincs szükség a felület sűrűbb újrafelosztására és a teljes fényvonal-pont számítási procedúra megismétlésére, mivel a pontosítás nem globális, hanem lokális

jellegű. Mindegyik fényvonal-pont ugyanolyan pontossággal számítható, és a felhasználó a saját igényeinek megfelelően, közvetlenül előírhatja a fényvonal-pontok és a fényvonal-pontokra illesztett, a fényvonalat reprezentáló, folytonos görbe pontosságát.

A tézist részletesen az 5. fejezetben tárgyalom, a tézishoz kapcsolódó publikációk: [19],[21],[22],[23],[24].

2. Tézis

Új eljárást dolgoztam ki a fényvonalak javítására. Az eljárás az egymást követő hibás fényvonal-szakaszok alakjából és mintázatából álló fényvonal struktúrát javítja.

Az eljárásban először kijelölöm a hibás fényvonal-szakaszok végpontjait, és a végpontokban meghatározom a fényvonalak érintőit. A végpontok meghatározása két lépésben történik. Először interaktívan, közelítőleg jelölöm ki a hibás fényvonal-szakaszok végpontjait, majd az így kijelölt végpontok között megjelenítem a fényvonal görbületi fésűjét, amellyel a felhasználó, a görbületi eltéréseket figyelembe véve, pontosítja a végpontok helyét.

A következő lépésben, egy erre a célra kifejlesztett algoritmust alkalmazva, az egymást követő hibás fényvonal-szakaszok meghatározott pontjaiból pontsorozatokat hozok létre, amelyeket felhasználva a fényvonal struktúra hibáját reprezentáló távolság és szögfüggvények (hibafüggvények) diszkrét pontjait határozom meg.

Ezt követően, a diszkrét pontokat felhasználva a hibafüggvényeket kisimítom. A függvény simítás kényszerei: a végpontok, és az érintők iránya a végpontokban. A simítás feltétele: a diszkrét pontok (hibás függvényértékek) és a simított függvény közti távolságok négyzetösszegének minimalizálása.

A simított (B-Spline) függvény fokszámát és a kontroll pontok maximális értékét a felhasználó alapértelmezett értékekből kiindulva adhatja meg. Az alapértelmezett értékeket kísérletekkel határoztam meg, egymástól geometriailag és funkcióban is jelentősen eltérő szabadformájú ipari felületeken, különböző nagyságú, kiterjedésű és komplexitású fényvonal struktúrák javításával.

A hibafüggvény simítását követően az új függvényértékekből új fényvonal pontokat számolok. A hibás fényvonal-szakaszokat a végpontjaik között kimetszem, majd a végpontokat, a végpontok érintőit és az új pontokat felhasználva helyükre javított fényvonal-szakaszokat illesztetek. A javított fényvonal-szakaszokat B-Spline görbékkel reprezentálom, amelyeket approximációt alkalmazva hozok létre. A görbeillesztés kényszere: végpontok, az érintők iránya a végpontokban. Az illesztés feltétele: a felhasználó által előírt illesztési pontosság.

Az eljárással a felhasználó az eddigi manuális helyett, nagyrészt automatizált módon javíthat, és képes nagyobb kiterjedésű és összetettebb fényvonal struktúra javítására is. Ezen felül, a módszer az alapértelmezett értékek alkalmazásával számszerűsíti a tervezői tapasztalatokat, amivel elérhetővé válik a minőségi és gyors javítás lehetősége a kevésbé tapasztalt tervező számára is.

A tézist a dolgozat 6. fejezetében tárgyalom, a tézishoz kapcsolódó publikációk: [1],[3],[4],[5],[6],[7],[8].

3. Tézis

Új eljárást dolgoztam ki a javított fényvonalaknak megfelelő felület előállítására. Az eljárás a nemlineáris, sokváltozós feladatot genetikus algoritmus (GA) alkalmazásával oldja meg.

A GA megtervezése során meghatározom az alkalmazandó kódolási módszert, a génreprezentációt, a keresési teret, a populáció méretét és a kezdeti populáció előállítási módszerét, a fitnessfüggvényt, az alkalmazandó genetikus operátorokat és az operátorok paramétereit.

A GA megtervezéséhez saját munkamódszert dolgoztam ki. A munkamódszerben a GA-t egy működő alapváltozatból hoztam létre lépésenként javítva. A javításban a GA paramétereit és operátorait különböző tesztfelületeken több szempontot figyelembe véve vizsgáltam. A vizsgálatokhoz mutatókat és mérőszámokat hoztam létre, amelyekkel a genetikus keresés folyamatát, a GA sebességét és megbízhatóságát értékeltem. A vizsgálatokban alkalmazott tesztfelületek egymástól geometriailag és funkcióban is jelentősen eltérő szabadformájú ipari felületek.

A tézist a dolgozat 7, 8, 9 fejezeteiben tárgyalom, a tézishoz kapcsolódó publikációk: [2],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[20],[25],[26].

4. Tézis

A fényvonal számító, a fényvonal javító és a felületet a javított fényvonalakhoz igazító eljárásokat felhasználva – egymásra építve –, egy új felületjavítási módszert alkottam meg, amely CAD rendszerbe eszközként integrálható.

A módszer alkalmazásával a tervező magas minőségi követelményeket kielégítő szabadformájú felületek lokális hibáit képes összetett, saját igényeknek megfelelő módon, nagyrészt automatizáltan javítani.

Az eszköz alkalmazhatóságát egymástól jelentősen eltérő hibajellemzőkkel (nagyság, kiterjedés, komplexitás) rendelkező, különböző geometriai bonyolultságú és funkciójú formatervezett ipari felületeken vizsgáltam és igazoltam.

A vizsgálatokhoz erre a célra kifejlesztett vizsgálati módszereket alkalmaztam. A módszer a kis és a nagyobb kiterjedésű lokális hibákat is jó hatékonysággal javítja, a felületi hiba nagyságára mérsékelten érzékeny. A módszert a Rhino 5[©] felülettervező rendszerbe illesztettem, alkalmazhatósága a felület CAD reprezentációjától független.

A tézist a dolgozat 10. fejezetében ismertetem, a tézishoz kapcsolódó publikációk: [2],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[20],[25],[26].

13. Az eredmények hasznosítási lehetősége

A kidolgozott módszer, széles hibajellemző határok között alkalmazható, nagyrészt automatizált, de lehetőséget ad egyéni paraméterek beállítására is, amelyekkel jelentősen megkönnyíti a lokális hibák problémájának kezelését. A tervezési folyamatba jól beilleszthető, mivel ésszerű idő alatt végrehajtható a javítás, az automatizáltság következtében kevesebb tapasztalatra van szükség a kiváló eredmény eléréséhez.

A módszer kifejlesztése során kapcsolatba kerültem a *Ford Motor Company* formatervezési részlegének vezetésével, akik részben tanácsaikkal láttak el, részben pedig érdeklődést mutattak a kidolgozott módszer saját felületrekonstrukciós tervezési eljárásaik közé történő beiktatására.

A módszer iránt ezen felül a *Solid Modeling Solution* szoftverfejlesztő cég is érdeklődik, akik a rendelkezésemre bocsájtott approximációs rutinjaikkal szintén segítették a módszer megalkotását.

14. A dolgozat tartalmának angol nyelvű ismertetése

The correction of local irregularities of high quality surfaces is not completely solved. The main reason to this is that the current methods have been developed for the correction of simple and elementarily small errors. The size, extent and complexity of the errors (error characteristics) on present day surfaces however vary largely. As the result, the application of current methods is laborious and difficult, the Designer many times corrects manually.

To solve this problem, I developed a surface correction method capable for correcting local surface errors with error characteristics varying on large scale.

For representing the surface geometry the method uses parametric representation of the free form surfaces in NURBS form. The method manipulates the surface geometry only by the modification of surface's control points.

For evaluation of local surface errors the method uses highlight lines. The highlight lines integrate the advantages of several methods, their shape is very similar to physical reflection lines, and their handling in computer environment is simple. For efficient and sensitive discovery of surface errors, method for calculation of highlight lines with high precision was developed.

In the process of correction the method first corrects highlight lines and next the control points of the surface are adjusted accordingly. To correct the highlight lines first, sequences of evaluation points are defined to measure the error in terms of distance and angle functions. Next, these functions are smoothed and based on the new function values, new highlight line points are calculated. The new highlight line curves are constructed using these points.

Next, the creation of the surface producing the corrected highlight lines takes place. The main problem in this step is the nonlinear relationship between the corrected highlight lines and the control points of the surface. To deal with nonlinearity current methods use a number of simplifications to linearize the relationship, as consequence their applicability is limited. To handle the nonlinearity I developed a genetic algorithm (GA).

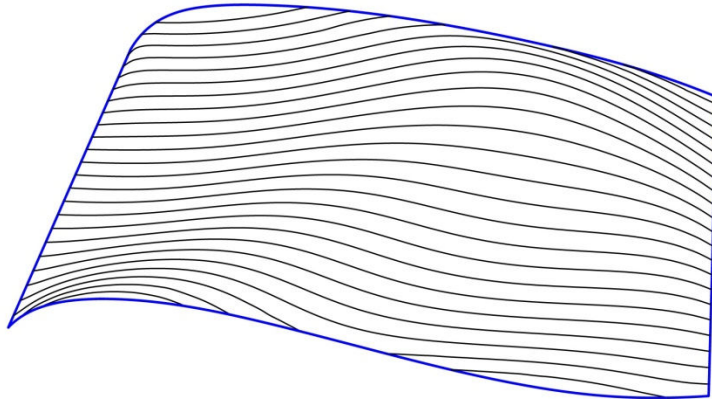
The parameters of the GA are determined multi-objective inspections evaluating the fastness and the reliability of the genetic search. In this process first the genetic representation of the surface and the stop conditions of the search were determined. Next, variants for fitness (objective) function were developed and the applicable genetic operators (scaling, selection, crossover and mutation) were selected. In next step I assembled a basic genetic algorithm, which was then fine-tuned. In fine-tuning process the parameters and the operators were systematically modified, and the GA produced was evaluated. For the evaluation I developed assessment numbers and indexes, suited for the task. In this process the GA was tested on several surfaces containing errors of different size, extent and complexity.

Using the methods for highlight line calculation, highlight line correction and adjustment of surface to corrected highlight lines, a new method for surface correction is created.

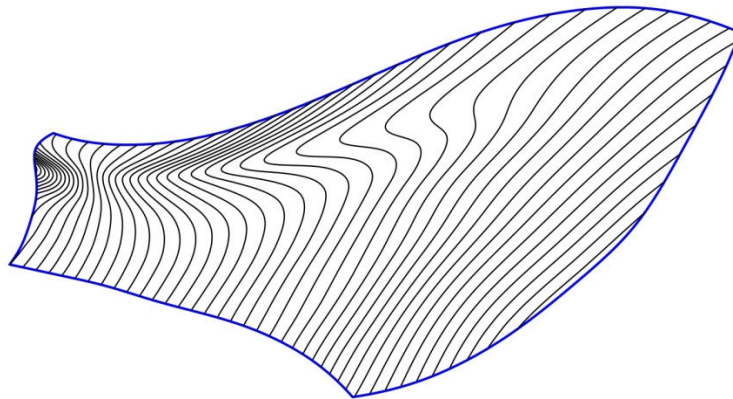
The method is capable to correct local errors of largely different extent with the same efficiency; it is moderately sensitive to error size and complexity, varying on a large scale. The method is robust, automated to a great extent, suitable to meet specific intents of the Designer. It was integrated into CAD environment, its application is independent of surface's CAD representation. The method's applicability was evaluated and proved with correction of industrial surfaces of various geometric complexity and functionality.

15. Mellékletek

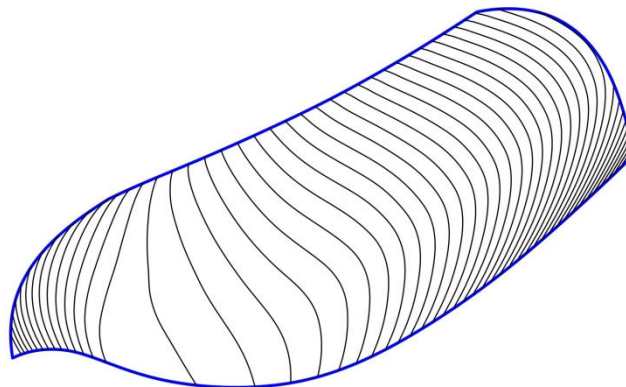
15.1. Példák fényvonalak számítására



68. ábra. Példa fényvonalak számítására

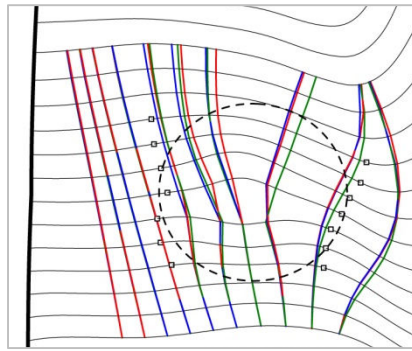


69. ábra. Példa fényvonalak számítására

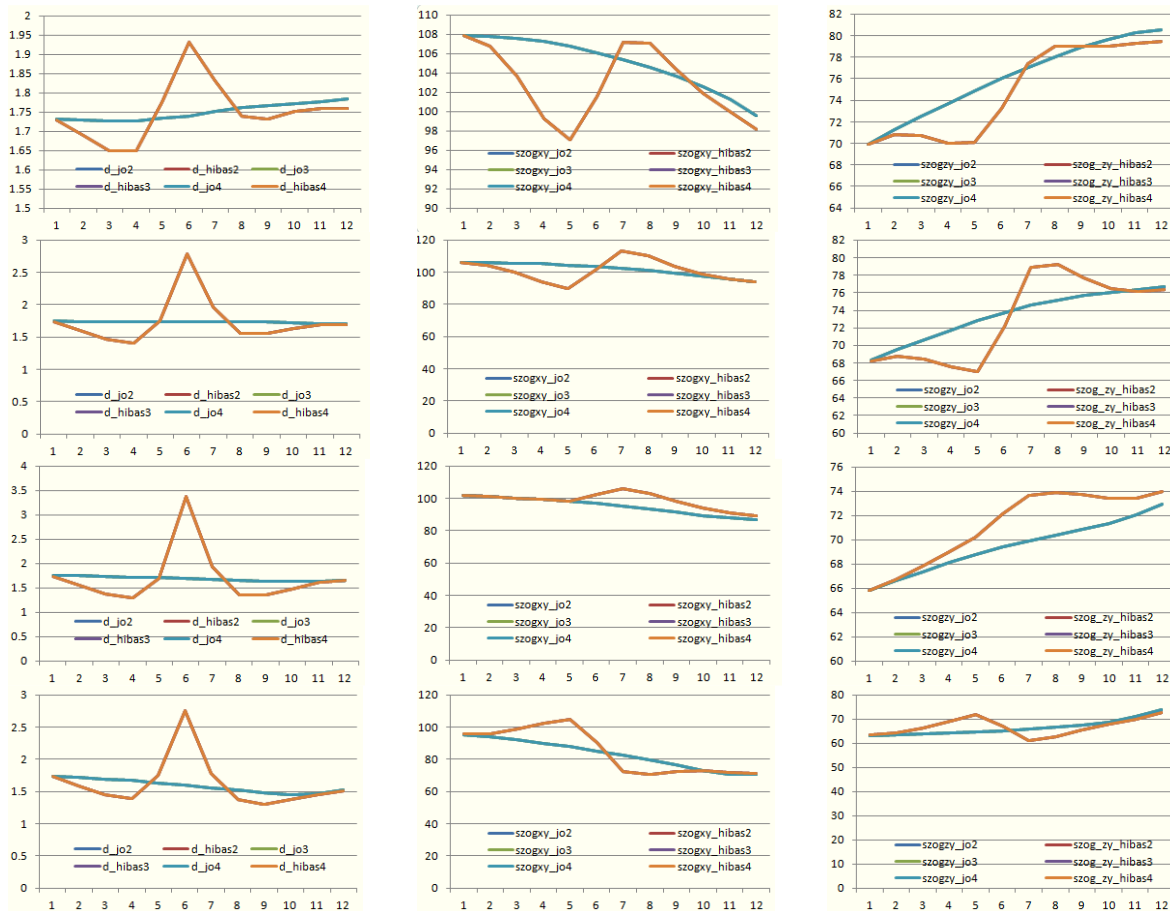


70. ábra. Példa fényvonalak számítására

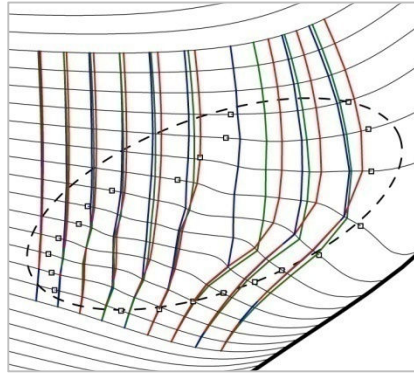
15.2. Fényvonal struktúra javítása



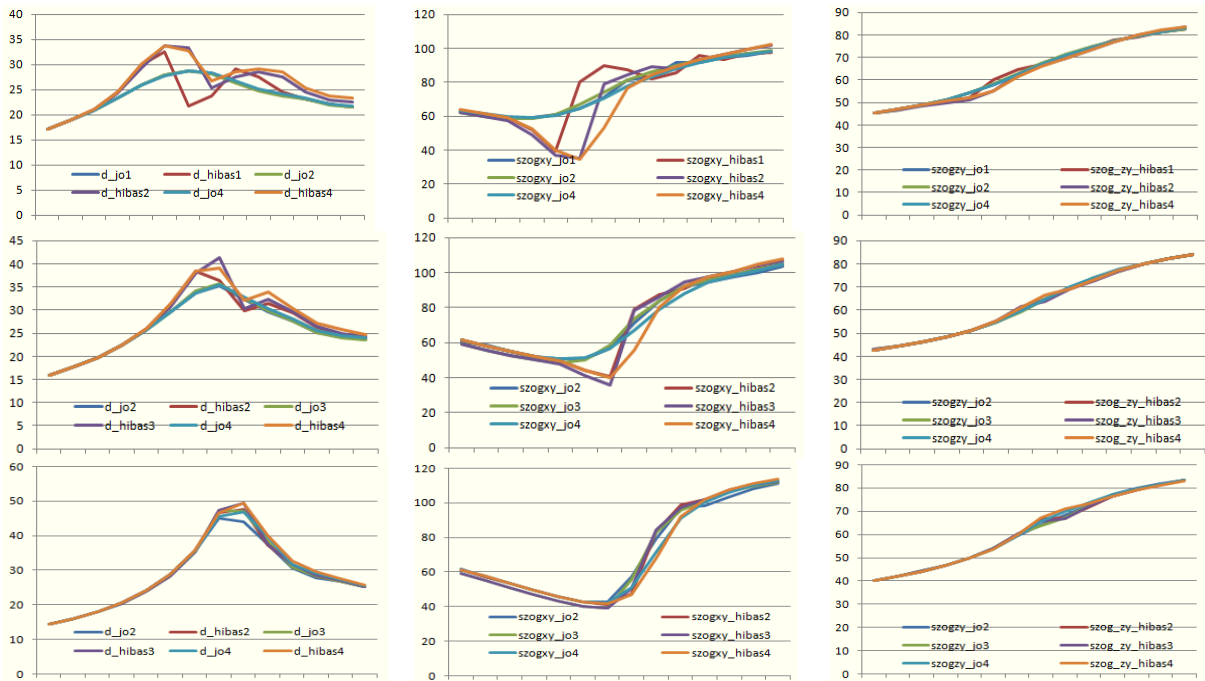
71. ábra. Az S2 felület pontsorozat vonalai (1-3 módszer)



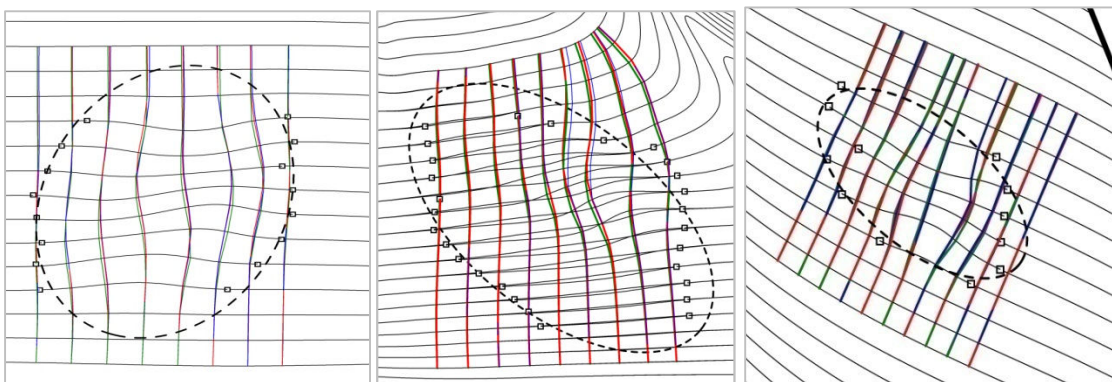
72. ábra. Az S2 felület hibás és javított értékelési függvényei (5-8 görbe).



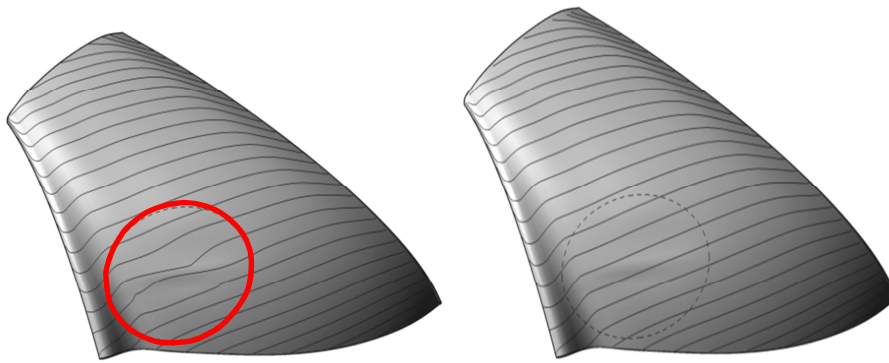
73. ábra. Az S7 felület pontsorozat vonalai (1-3 módszer).



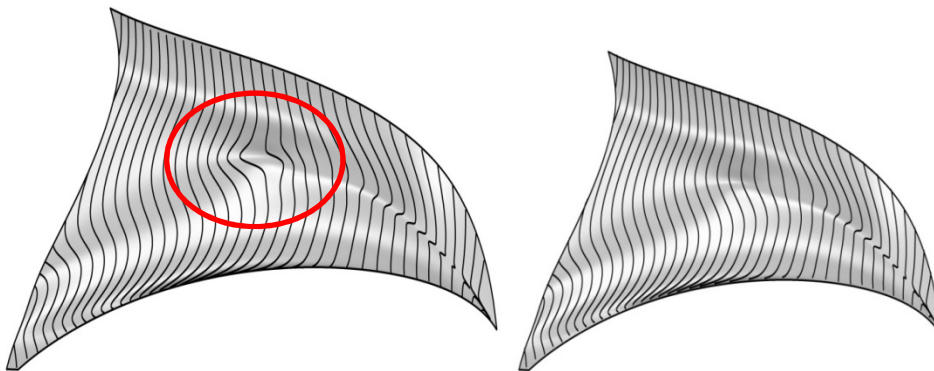
74. ábra. Az S7 felület hibás és javított értékelési függvényei (7-9 görbe).



75. ábra. Az S3 felület pontsorozat vonalai (1-3 módszer).

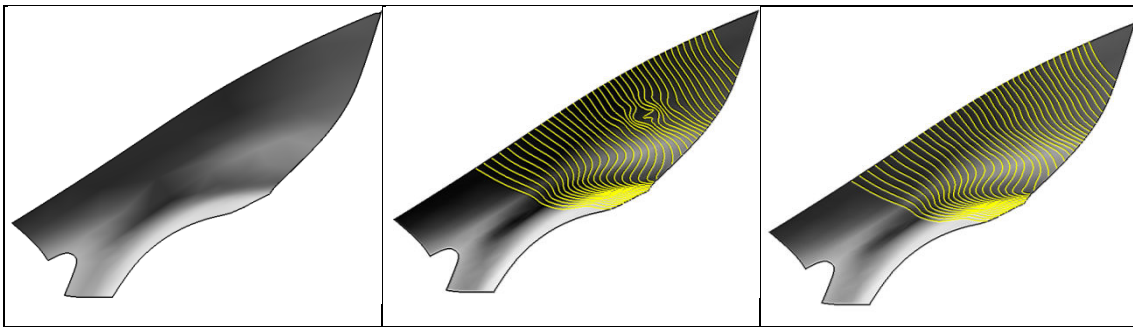


76. ábra. Példa fényvonal struktúra javításra (autókarosszéria-elem)

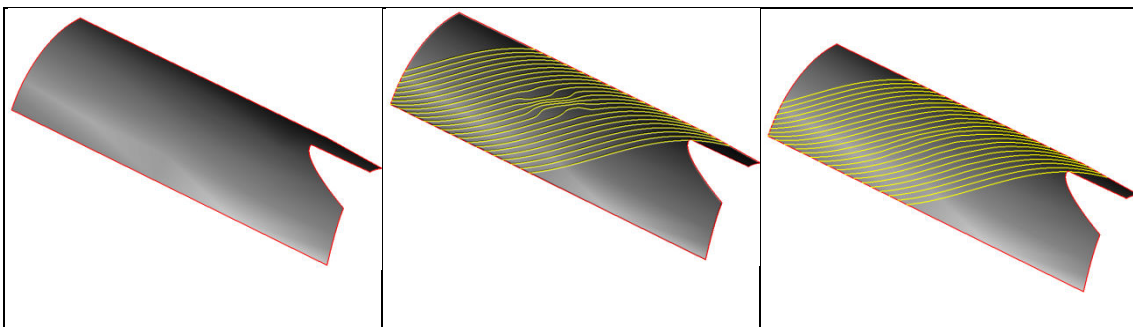


77. ábra. Példa fényvonal struktúra javításra (autókarosszéria-elem)

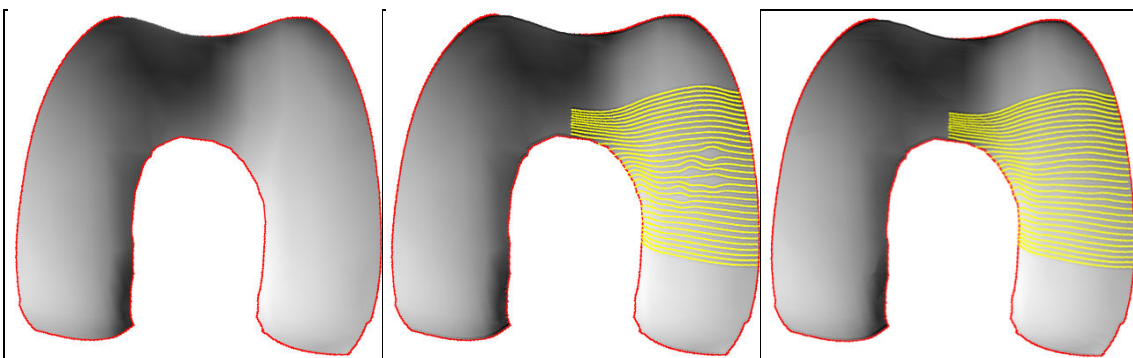
15.3. Példák felületmodellek lokális hibáinak javítására



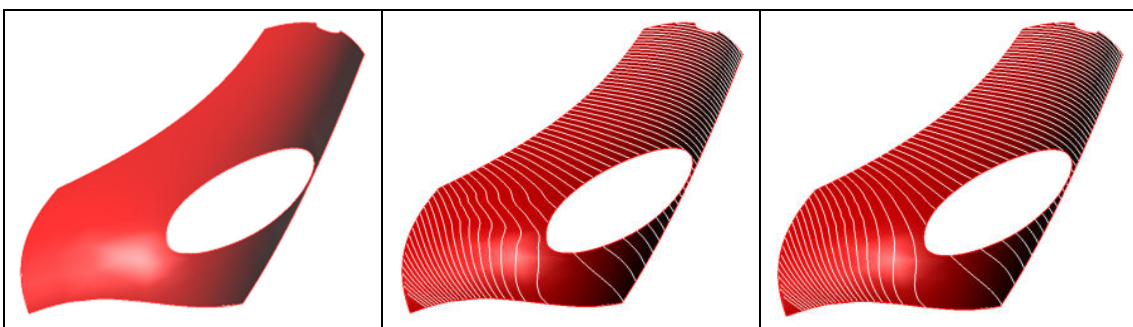
78. ábra. S2 Sárvédő elem javítás előtt (a., b.) és után (c.)



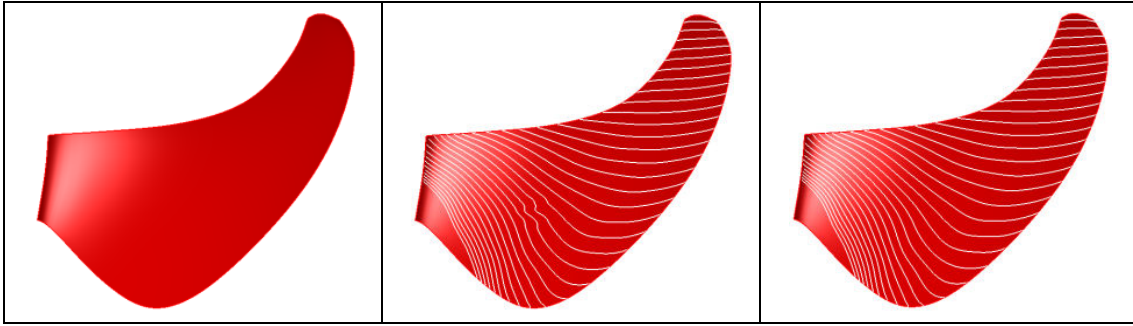
79. ábra. S8 Turbinalapát javítás előtt (a., b.) és után (c.)



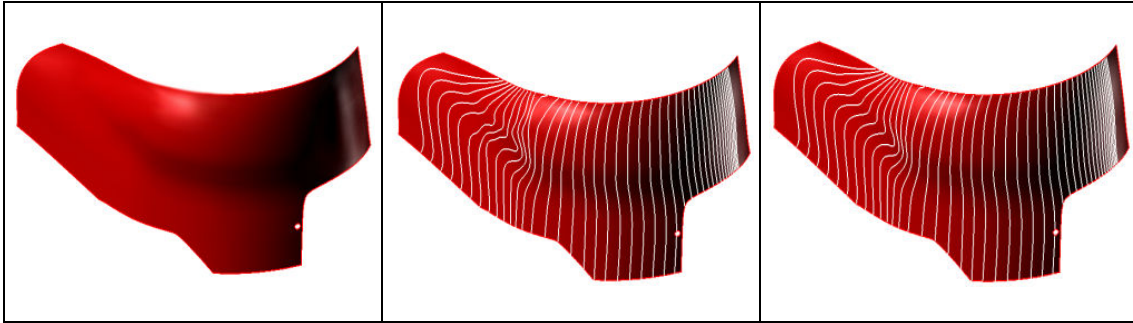
80. ábra. S6 Térdprotézis elem (femur) javítás előtt (a., b.) és után (c.)



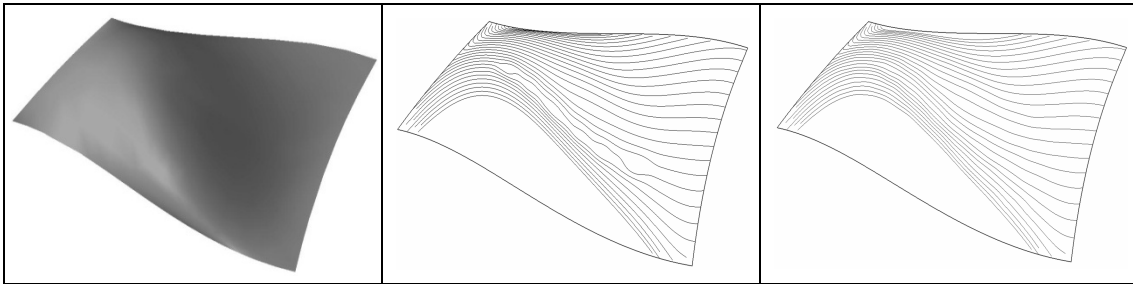
81. ábra. S7 Karosszéria elem javítás előtt (a., b.) és után (c.)



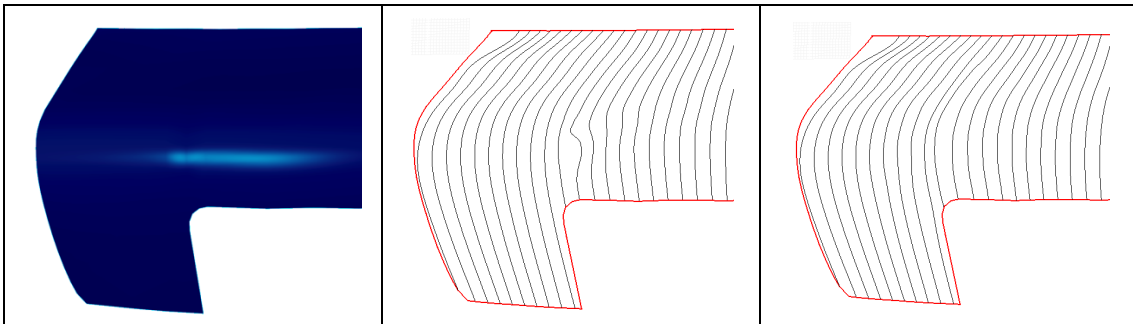
82. ábra. S9 Propeller lapát javítás előtt (a., b.) és után (c.)



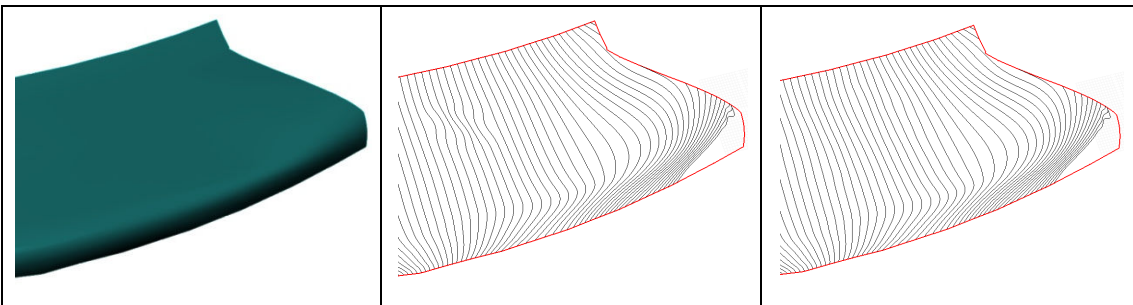
83. ábra. S10 karosszéria elem javítás előtt (a., b.) és után (c.)



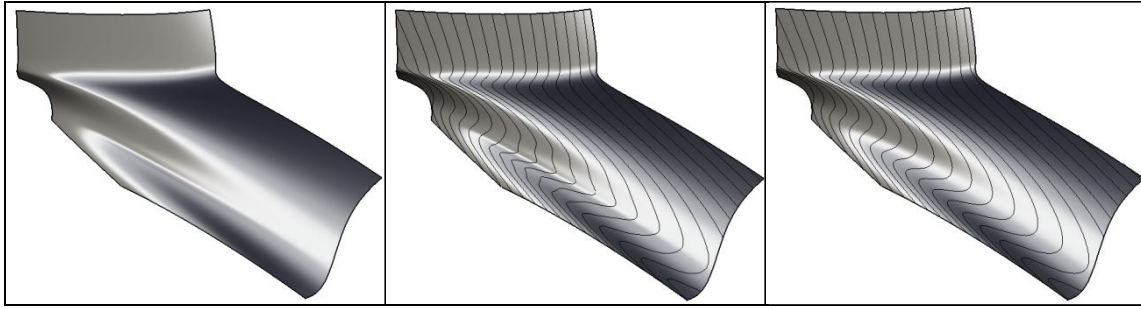
84. ábra. S4 szabadformájú felületelem javítás előtt (a., b.) és után (c.)



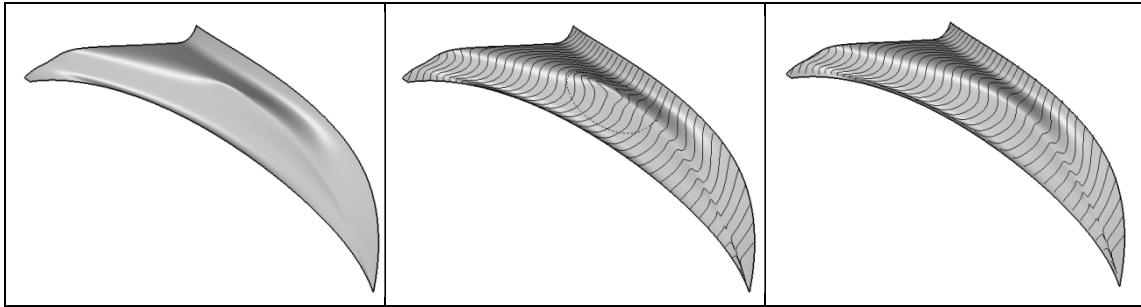
85. ábra S3 csomagtartó fedél javítás előtt (a., b.) és után (c.)



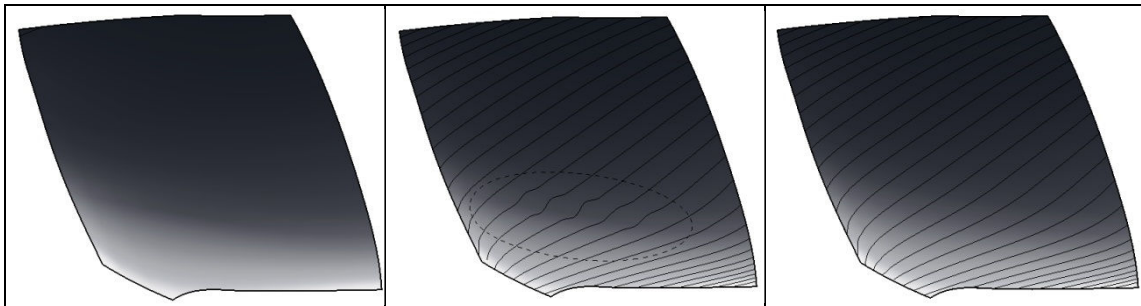
86. ábra. S1 Csomagtartó fedél javítás előtt (a., b.) és után (c.)



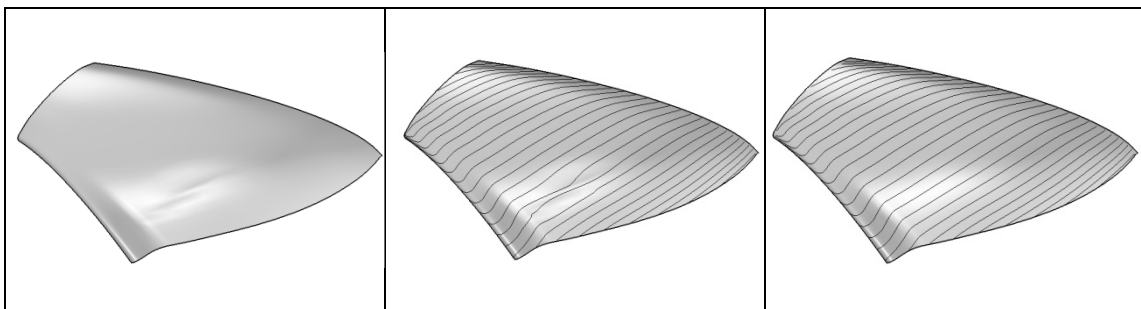
87. ábra. Karosszéria elem javítás előtt (a., b.) és után (c.)



88. ábra. Karosszéria elem javítás előtt (a., b.) és után (c.)



89. ábra. Karosszéria elem javítás előtt (a., b.) és után (c.)



90. ábra. S5 Karosszéria elem javítás előtt (a., b.) és után (c.)

16. Irodalomjegyzék

- [Andersson, 1996] Andersson, R., Surface design based on brightness intensity or isophotes-theory and practice, Advanced Course on FAIRSHAPE, pp. 131– 143. B. G. Teubner, 1996
- [Andersson, 2005] Andersson, R., Surface Design through Modification of Shadow Lines, Holistic Product Development, Werner Dankwort (Ed.), Shaker Verlag, Aachen, 2005, Vol.5 pp. 239-257
- [Álmos et al., 2003] Genetikus algoritmusok, Typotex Kiadó 2003, ISBN 978-963-9326-45-3
- [Beier & Chen, 1994] Beier, K.-P., Chen, Y., Highlight-line algorithm for real-time surface-quality assessment, *Computer-Aided Design* 26, pp. 268-278
- [Bosinco et al., 1998] Bosinco, P., Durand, G., Goussard, J., Lieutier, A., Massabo, A., Complex shape modifications, *Proceedings of the IDMME '98 Conference*, Compiègne
- [Bonneau et Hagen, 1994] Bonneau, G.-P., Hagen, H., Variational design of rational Bézier curves and surfaces, *Curves and Surfaces in Geometric Design*, pp. 5 I-58
- [Case et al., 2002] Case, K., Graham, I.J., Wood, R.L., Abdul Karim, M.S., CAD Genetic Algorithms for Evolutionary Form and Function Design, Proceedings of the 18th National Conference on Manufacturing Research Leeds, pp. 103-107
- [Chen, 1993] Chen, Y., Highlight lines for surface quality control and shape manipulation, Thesis (Ph.D.) University of Michigan
- [Chen et al., 1997] Chen, Y., Beier, K.-P., Papageorgiou, D., Direct highlight line modification on NURBS surfaces. *Computer Aided Geometric Design*, 14 (1997), pp. 583-601
- [Creusen & Schoormans, 2005], M.,E., H., Creusen, J., P.,L., Schoormans, The Different Roles of Product Appearance in Consumer Choice, *Journal of Product Innovation Management*, Vol. 22(1), pp. 63–81
- [Creusen et al., 2010] Creusen, M., Veryzer, R., Schoormans, J., Product value importance and consumer preference for visual complexity and symmetry, *EUROPEAN JOURNAL OF MARKETING* Volume: 44 Issue: 9-10 pp.: 1437-1452
- [Covarrubias, & Bordegoni, 2014] Covarrubias, M., Bordegoni, M., Interaction with virtual aesthetic shapes through a desktop mechatronic system: A desktop system that supports designers in the creation, evaluation and modification of aesthetic virtual shapes, 2014, *Virtual and Physical Prototyping*, 9(1), pp.27 -43
- [Dankwort & Podehl, 1998.] Dankwort, C.W., Podehl, G., A new aesthetic design workflow – Results from the European project FIORES. Dagstuhl Seminar 98461: CAD Tools and Algorithms for Product Design. International Conference and Research Centre for Computer Science, Schloss Dagstuhl, Warden, Nov. 1998
- [Dankwort et al., 2001] Dankwort, W., Podehl, G., A New Aesthetic Design Workflow - Results from the European Project FIORES
- [Deb et al., 2002] Deb K., Anand, A., Joshi, D., A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization, *Evolutionary Computation* , Vol. 10, No. 4, pp. 371-395
- [Deb, 2009] Optimization for Engineering Design: Algorithms and Examples, PHI Learning Pvt. Ltd., India E
- [Dumitrescu et al., 2000] Dumitrescu D., Lazzarini B., Jain L. C., Dumitrescu A., Evolutionary Computation, CRC Press, International Series on Computational Intelligence, 2000
- [Fan, et al. 2008] Fan, S., Shi, M., Ke, Y., Li, J., Local Beautification Technique for Freeform Surfaces
- [Farin & Sapidis, 1989] Farin, G., Sapidis, N., Curvature and the fairness of curves and surfaces. *IEEE Comp. Graph. Applic.*, Vol.3 pp. 52-57
- [Farin et al., 2002] Farin, G., Hoschek, J., Kim, M.-S., Handbook of Computer Aided Geometric Design, ISBN: 978-0-444-51104-1, Elsevier B.V., Amsterdam, Holland
- [Farin & Hansford, 2002] Farin, G., Hansford, D., Reflection Lines: Shape in Automotive Design, *The Magazine of Design & Technology Education*, New Jersey , March 2002
- [Goldberg, 1989] Goldberg E., D., Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, 1989

- [Giannini & Monti, 2002] Giannini F., Monti, M., CAD Tools Based On Aesthetic Properties, Eurographics – Italian Chapter, Genova
- [Greiner, 1994] Greiner, G., Variational design and fairing of spline surfaces, Eurographics, 1994, pp. 143-154.
- [Hagen & Santarelli, 1992] Hagen, H., Santarelli, P., Variational design of smooth B-spline surfaces. In H. Hagen, editor, Topics in Surface Modeling, pages 85-94. SIAM
- [Hagen & Schulze, 1990] Hagen, H., Schulze, G., Extremalprinzipien im Kurven- und Flachendesign, In Verfahren der graphischen Datenverarbeitung, Springer, pp. 46-60
- [Hahmann et al., 2008] Hahmann, S., Belyaev, A., Busé, L., Elber, G., Mourrain, B., Rössl C., Shape Interrogation, Shape Analysis and Structuring, Springer
- [Hermann et al., 1997] Hermann, T., Kovacs, Z., Várady, T., Special applications in surface fitting, In W. Strasser, R. Klein, and R. Rau, editors, Geometric Modeling: Theory and Practice, pp. 14-31
- [Herrera et al., 1998] Herrera F., Lozano M., Verdegay J. L., (1998), Tackling real-coded genetic algorithms, Artificial Intelligence Review, Vol. 12(4), pp. 265-319
- [Higashi et al., 1999] Higashi M., Harada H., Kuroda M., Generation of Surfaces with Smooth Highlight Lines, Conference on Curves and Surfaces (4th), St. Malo, France, 1-7 Jul 1999, Vol. 1, pp. 145-152
- [Holland, 1992] Holland, J., H., Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press, 1992, ISBN 0262581116
- [Homburg et al., 2015] Homburg, C., Schwemmler, M., Kuehnl, C., New Product Design: Concept, Measurement, and Consequences, JOURNAL OF MARKETING Volume: 79 Issue: 3 pp.: 41-56
- [Horváth & Juhász, 1996] Horváth, I., Juhász I., Számítógéppel segített gépészeti tervezés, Műszaki könyvkiadó, 1996, ISBN 963 16 1051 9
- [Jalukow & Michalewicz, 1991] Z. Jalukow, C. Z., Michalewicz, Z. An experimental Comparison of Binary and Floating Point Representation Genetic Algorithms, Proceedings of the 4th Conference on the Genetic Algorithms, pp. 31-36
- [Jun-Hai et al. 2003] Jun-Hai Yong, Fuhua (Frank) Cheng, Yifan Chen, Paul Stewart, Kenjiro T. Miura, Dynamic highlight line generation for locally deforming NURBS surfaces, Computer-Aided Design, Vol. 5, Issue 10, pp. 881-892
- [Kaufman & Klass, 1988] Kaufman, E., Klass, R., Smoothing surfaces using reflection lines for families of splines, *Computer-Aided Design* 20, pp. 312-316
- [Keith et al., 2008] Aesthetics Considerations in Evolutionary Computer Aided Design, Mechanical and Manufacturing Engineering Faculty, Loughborough University, UK
- [Klass, 1980] Klass, R., Correction of local surface irregularities using reflection lines, *Computer-Aided Design* 12, pp. 73-76
- [Lomonosov & Renner, 2008] Lomonosov, E., & Renner, G., An Evolutionary Algorithm for Surface Modification, Acta Cybernetica Hungarica
- [Loos et al., 1999] Loos J., Greiner G., Seidel H.-P., Modeling of surfaces with fair reflection line pattern, *Shape Modeling International* pp. 256-264
- [Luchs et al., 2012] Luchs, M., Brower, J., Chitturi, R., Product Choice and the Importance of Aesthetic Design Given the Emotion-laden Trade-off between Sustainability and Functional Performance, JOURNAL OF PRODUCT INNOVATION MANAGEMENT Volume: 29 Issue: 6 pp: 903-916
- [Markus et al., 1995] Márkus, A., Renner, G., Váncza, J., Genetic Algorithms in Free Form Curve Design, Vanderbilt University Press, La Verge, pp. 1-12
- [Markus et al., 1997] Markus, A., Renner, G., Vancza, J., Spline Interpolation with Genetic Algorithms, Proceedings International Conference on Shape Modeling and Applications, Aizu-Wakamatsu, Japan, pp. 47-54
- [Michalewicz, 1996] Michalewicz, Z., Genetic algorithms and data structures - evolution programs. Springer 1996, ISBN 978-3-540-60676-5, pp. I-XX, 1-387

- [Munaux, O., 2004] Munaux, O., CAD Interface and Framework for Curve Optimisation Applications Thesis, Cranfield University
- [Nishiyama et al., 2007] Y. Nishiyama, Y. Nishimura, T. Sasaki, T. Maekawa, Surface Fairing Using Circular Highlight Lines, *Computer-Aided Design and Applications*, Nr. 1–4, Vol. 4, pp. 405–414
- [Paláncz, 2011] B. Paláncz, *Numerikus módszerek*, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2011
- [Piegl & Tiller, 1997] Piegl, L., Tiller, W., *The NURBS Book*, ISBN 3-540-61545-8, Springer Verlag, Berlin, Heidelberg New York, 1997
- [Poeschl, 1984] Thomas Poeschl, Detecting surface irregularities using isophotes, *Computer Aided Geometric Design* 1 (1984) pp. 163-168
- [Rando & Roulier, 1991] Rando, T., Roulier, J.A., Designing faired parametric surfaces. *Computer Aided Design*, 23, pp. 492-497
- [Renner & Vida, 2000] Renner, G., Vida, J., Reconstruction of free-form Features with Genetic Algorithm, Proceedings of International CIRP Design Seminar, Haifa, Israel, pp. 343-354
- [Renner & Ekárt, 2003] Renner, G., Ekárt, A., Genetic algorithms in computer aided design, *COMPUTER-AIDED DESIGN* Volume: 35 Issue: 8 pp: 709-726
- [Roy et al., 2000] Roy, R., Jared, G., Tiwari, A., Munaux, O. (2000). Development of 'Tool Box' for Reallife Engineering Design Optimisation. Cranfield, UK: Technical Report Cranfield University, Flexo Report – 8
- [Stewart, 1991] Stewart, P. J., Direct Shape Control Of Free-Form Curves And Surfaces With Generalized Basis Functions, *Ph.D. dissertation*, University of Michigan
- [Talke, et al., 2009] Talke K., Salomo, S., Wieringa, J., et al., What about Design Newness? Investigating the Relevance of a Neglected Dimension of Product Innovativeness, *JOURNAL OF PRODUCT INNOVATION MANAGEMENT* Volume: 26 Issue: 6 pp.: 601-615
- [Vancsay, 2001] Vancsay Gy., Genetikus algoritmusok alkalmazása a gépészeti tervezésben, PhD dolgozat, BME Gépszerkeztani Intézet
- [Várady & Martin, 2002] Várady, T., Martin, R., Reverse Engineering, *Handbook of Computer Aided Geometric Design*, 2002, Ch. 26, Elsevier
- [Várady et al., 2007] Várady, T., Facello, M., A., Terék, Zs., Automatic extraction of surface structures in digital shape reconstruction. *Computer-Aided Design* 39, 5 (2007), pp. 379–388
- [Yi et al., 2008] Yi, P., Wei X., Zhao T., GA-based creative shape design method integrating surface fitting and generating, 7th World Congress on Intelligent Control and Automation, WCICA 2008
- [Zhang & Lai, 2001] Zhang, Z., Lai, W., An Application of Genetic Algorithms in Smoothing Automotive Body Surface, *Proceedings of the Congress on Evolutionary Computation*, Vol. 2 pp. 1296 - 1302
- [Zhang & Cheng, 1998] Zhang, C., Cheng, F., Removing local irregularities of NURBS surfaces by modifying highlight lines, *Computer-Aided Design*, 30(12), pp. 923–930
- [Zhang, et al., 2001] Zhang, C., Zhang, P., Cheng, F., Fairing spline curves and surfaces by minimizing energy, *Computer-Aided Design*, Vol. 33 (2001), pp. 913-923

A kutatási témában megjelent saját publikációk

- [1] Gyurecz & Renner, Refinement of Shape and Structure of Highlight Lines, *Proceedings of Conference of the Hungarian Association for Image Processing and Pattern Recognition*, pp 1-7, 2015.01.27. 2015.01.30. Kecskemét, NJSZT-SZTAKI-GAMF
- [2] Gyurecz et al., A Case Study of Applying Genetic Algorithm in Surface Correction Proceedings of *IESB 2014: International Engineering Symposium at Bánki*. Budapest, Magyarország, 2014.11.20. Budapest Óbudai Egyetem, 2014. pp. 89-96. (ISBN:978-615-5460-08-1)

- [3] Gyurecz & Renner, Evaluation and Correction of Highlight Line Structures, *Proceedings of Factory Automation 2013*, Veszprém, Magyarország, 2013.05.21-2013.05.22. Veszprém: University of Pannonia, 2013. p. 4., (ISBN:978-615-5044-80-9)
- [4] Gyurecz & Renner, Tools and Methods for Improving Highlight Lines, , *Proceedings of the 15th International Workshop on Computer Science and Information Technologies, CSIT'2013*, Vienna-Budapest-Bratislava. Vol. 2., Bécs, Ausztria, 2013.09.15-2013.09.21. Ufa: Ufa State Aviation Technical University, 2013. pp. 198-201. (ISBN:978-5-4221-0471-0)
- [5] Gyurecz & Renner, Correction of Highlight Structures, In: Kovacs G. L., Kochan D., (szerk.), *Digital Product and Process Development Systems: IFIP TC 5 International Conference, NEW PROLAMAT 2013*, Dresden, Germany, October 10-11, 2013. Proceedings., Dresden, Németország, 2013.10.10-2013.10.11. Berlin: Springer, 2013. pp. 230-238. (IFIP Advances in Information and Communication Technology; 411.) (ISBN:978-3-642-41328-5)
- [6] Gyurecz & Renner, Csibi Vencel (szerk.) Method for Correcting Highlight Line Structures, In: *Csibi Vencel (szerk.), OGÉT 2012, XX. Nemzetközi Gépészeti Találkozó*, Kolozsvár, Románia, 2012.04.19-2012.04.22. Kolozsvár: Erdélyi Magyar Műszaki Tudományos Társaság, 2012. pp. 149-153
- [7] Gyurecz & Bercsey, Surface Shape Correction by Highlight Lines, In: Ralph Stelzer, Karl-Heinrich Grote, Klaus Brökel, Frank Rieg, Jörg Feldhusen (szerk.), *Entwerfen Entwickeln Erleben: Methoden und Werkzeuge in der Produktentwicklung*, Dresden, Németország, 2012.06.15-2012.06.16. Dresden: Technische Universität Dresden, 2012. pp. 513-520, (ISBN:978-3942710800)
- [8] Gyurecz & Renner, Method for Improvement of Highlight Line Structures, In: *MANUFACTURE 2012 MTA SZTA-KI. Konferencia helye, ideje: Budapest, Magyarország, 2012.11.14-2012.11.16. Budapest: pp. 1-7*
- [9] Gyurecz & Bercsey, EVOLUTIONARY ALGORITHM IN SURFACE SHAPE CORRECTION: poszter, In: *9th International workshop on Integrated Product Development*. Magdeburg, Németország, 2012.09.05-2012.09.07. Magdeburg: p. 1
- [10] Gyurecz & Renner, Correction of Surfaces by Highlight Lines, In: Nagy Zoltán (szerk.), *Factory Automation 2011 Conference*, Győr, Magyarország, 2011.05.24-2011.05.26. Győr: Széchenyi István Egyetem, 2011. pp. 241-247, (ISBN:978-963-7175-3)
- [11] Gyurecz & Renner, Új felületminőség javító módszer kifejlesztése, In: Dr Csibi Venczel-J (szerk.), *OGÉT 2011-XIX. Nemzetközi Gépészeti Találkozó*. Csíksomlyó, Románia, 2011.04.28-2011.05.01. Kolozsvár: Erdélyi Magyar Műszaki Tudományos Társaság, pp. 152-155
- [12] Gyurecz & Renner, Correcting Fine Structure of Surfaces by Genetic Algorithm, *ACTA POLYTECHNICA HUNGARICA* 8:(6) pp. 181-190. (2011)
- [13] Gyurecz et al., Design and Optimization of Fine Structure of High Quality Surfaces *MECHEDU 1st Regional Conference-Mechatronics in Practice and Education*. Subotica, Szerbia, 2011.12.08-2011.12.10. Subotica Szabdkai Műszaki Szakfőiskola, 2011. pp. 186-191.(ISBN:978-86-85409-67-7)
- [14] Gyurecz & Renner, Evaluating and Correcting the Reflection Characteristics of Surfaces, In: Szirmay-Kalos L, Renner G (szerk.) V. Magyar Számítógépes Grafika és Geometria Konferencia. 226 p. Budapest, Magyarország, 2010.01.26-2010.01.27. Budapest: NJSZT, 2010. pp. 73-80. (ISBN:978-963-421-591-2)
- [15] Gyurecz & Renner, Improving Quality of Freeform Surfaces Using Genetic Algorithm and Highlight Lines In: Horvath I, Mandorli F, Rusák Z (szerk.), *TMCE 2010. Tools and Methods of Competitive Engineering. Proceedings of the Eight International Symposium*. Vol. 1., Ancona, Olaszország Delft: Delft Univ. of Technology, 2010. pp. 317-329., (ISBN:97890-5155-06-3)
- [16] Gyurecz & Renner, Surface Shape Optimization With Genetic Algorithm and Highlight Lines, In: Stépán G, T. Szalay, Á. Antal, I. Gyurika (szerk.), *Gépészet 2010: Proceedings of the Seventh Conference on Mechanical Engineering*. 926 p., Budapest, Magyarország, 2010.05.25-2010.05.26. Budapest: Budapest University of Technology and Economics, 2010. pp. 490-495, (ISBN:978-963-313-007-0)
- [17] Gyurecz & Renner, Application of Highlight Lines in the Design of Technical Surfaces, In: *MANUFACTURE-HU Nemzeti Technológiai Platform "GTENTP08" Szakmai Tanácsadó Testülete (szerk.), Gyártás 2010, Manufacturing 2010: A GTE XX. nemzetközi gyártás konferenciája*. Budapest, Magyarország, 2010.10.20-2010.10.21. Budapest: Gépipari Tudományos Egyesület, 2010. pp. 1-5. (ISBN:978-963-9058-31-6)

- [18] Gyurecz & Renner, Genetic Algorithm for Correcting Fine Structure of Surfaces by Highlight Lines, *ÓBUDA UNIVERSITY E-BULLETIN* 1:(1) pp. 79-85. (2010)
- [19] Gyurecz & Renner, Computation Method for Evaluation of Surface Fine Structure by Highlight Lines, *ÓBUDA UNIVERSITY E-BULLETIN* 1:(1) pp. 71-77. (2010)
- [20] Gyurecz & Renner, Improvement of Shape Properties of Technical Surfaces, *SYMMETRY: ART AND SCIENCE* 1:(1-4), Polish Academy of Sciences pp. 262-265. (2010)
- [21] Gyurecz & Renner, New Method for the Computation of Highlight Lines, *MŰSZAKI SZEMLE (EMT)* 12:(különszám) pp. 146-149. (2009)
- [22] Gyurecz & Renner, Robust Computation of Reflection Lines, *JOURNAL OF MACHINE MANUFACTURING* 1: pp. 1-4. (2009), Ed: Andrew Kaldos (ISBN 0016-8580)
- [23] Gyurecz & Renner, Evaluation of Aesthetic Quality of Surfaces by Reflection Lines, *SYMMETRY: ART AND SCIENCE* 1:(1-4), Polish Academy of Sciences pp. 208-211. (2009)
- [24] Gyurecz & Renner, Computation of Reflection Lines for the Design of High Quality Surfaces, In: Haidegger Géza (szerk.), *Biannual 19th International Conference on Manufacturing 2008*. Proceedings, Budapest, Magyarország, 2008.11.06-2008.11.07. Budapest: Gépipari Tudományos Egyesület, 2008. pp. 107-112. (ISBN:978-963-9058-25-5)
- [25] Gyurecz, Genetic Algorithm in Removing Local NURBS Surface Irregularities Using Highlight Lines, In: *ACM, GECCO 2008*, Atlanta, Amerikai Egyesült Államok, 2008.07.12-2008.07.16. New York: ACM - IEEE Service Center, 2008. pp. 1703-1704, (ISBN:978-1-60558-130-9)
- [26] Gyurecz, Removing Local Surface Irregularities by Modifying Highlight Lines, In: Ungvári László LINDI 2007. Wildau, Germany, 2007.09.13-2007.09.15. (IEEE), Wildau Fachhochschule: Óbuda University, 2007. pp. 150-153.