BARICENTRIKUS KOORDINÁTÁKON ALAPULÓ DEFORMÁCIÓS TECHNIKÁK HATÉKONYABB HASZNÁLATÁT SEGÍTŐ MÓDSZEREK FEJLESZTÉSE

Egyetemi doktori (PhD) értekezés

A szerző neve: Tóth Ákos

A témavezető neve: Dr. Kunkli Roland Imre

Debreceni Egyetem
Természettudományi és Informatikai Doktori Tanács
Informatikai Tudományok Doktori Iskola
Debrecen, 2021.

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

Debrecen, 2021. . . . .

. . . .
Tóth Ákos
jelölt

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

Debrecen, 2021. . . . .

. . . .
Dr. Kunkli Roland Imre
témavezető

BARICENTRIKUS KOORDINÁTÁKON ALAPULÓ DEFORMÁCIÓS TECHNIKÁK HATÉKONYABB HASZNÁLATÁT SEGÍTŐ MÓDSZEREK FEJLESZTÉSE

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

Írta: Tóth Ákos okleveles programtervező informatikus

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

Témavezető: Dr. Kunkli Roland Imre

Az értekezés bírálói:

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

A bírálóbizottság:

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

Az értekezés védésének időpontja: 20. . . ..

Tartalomjegyzék

1. Bevezetés

A koordináta-rendszerek olyan eszközök, melyek lehetővé teszik számunkra azt, hogy egy geometriai alapelem helyzetét egyértelműen meghatározzuk a térben. Miközben a Descartes-féle derékszögű koordináta-rendszer talán a leggyakrabban használt és tárgyalt rendszer, léteznek olyan problémák is, melyek vizsgálata és megoldása egy alternatív koordináta-rendszerben sokkal egyszerűbb. Ezen dolgozatban tárgyalt megoldások és eredmények alapjául egy olyan koordináta-rendszer szolgál, melyben egy geometriai objektum pozícióját egy politóp helyzetéhez viszonyítva tudjuk definiálni. A szóban forgó rendszer koordinátáit baricentrikus koordinátáknak nevezzük, melyek fogalmát először Möbius definiálta 1827-ben [1]. Wachspress 1975-ben publikált úttörő munkájától [2] kezdve a baricentrikus koordináták általánosításra kerültek tetszőleges sokszögekre a síkban, és általános politópokra magasabb dimenziókban. Ennek köszönhetően ma már gyakran használatosak különböző adatok lineáris interpolációjára, háromszöghálók paraméterezésére, kép- és modelldeformációra, valamint végeselem módszerek esetén.

Az általánosított baricentrikus koordináta-módszerek néhány tulajdonságukban eltérnek egymástól és különböző típusú sokszögek esetén eltérően viselkednek, ezért összehasonlításuk gyakran felmerül a szakirodalomban [3]. A doktori iskolában való kutatásom kezdetén, a jelenleg ismert vizsgálatokhoz képest, egy részletesebb szempontrendszer alapján elvégeztem a koordináta-módszerek összehasonlítását.

A baricentrikus koordináta-módszerek gyakran használatosak alakzat- és képdeformációra, mely alkalmazások használatakor elengedhetetlen a deformálandó képet körülvevő sokszög háromszögelése. Mivel a háromszögelés meghatározása nem triviális, nagymértékben befolyásolja a deformációs végeredményt, így ezen irányú kutatásainkban megvizsgáltuk a különböző háromszögelési technikákat a baricentrikus koordinátákon alapuló deformációs módszerek vonatkozásában. Továbbá kidolgoztunk egy olyan megoldást, mely az eddigi módszereknél jobb eredményeket biztosít bizonyos feltételek teljesülése esetén.

A képdeformációs alkalmazások analógiájára a baricentrikus koordináta-módszerek a háromdimenziós térben is alkalmazhatóak deformációra. Ebben az esetben a manipulációt a modellt burkoló ketrecmodell biztosítja. Ezen ketrecmodell meghatározása nagyon sok esetben manuálisan történik, azonban az elmúlt években néhány módszer automatizálta ezt a folyamatot [4, 5, 6]. Kutatásunk egyik céljaként fogalmaztuk meg egy olyan módszer létrehozását, mely az eddigi módszereknél gyorsabban és hatékonyabban képes egy tetszőleges háromdimenziós modellhez ketrecmodellt létrehozni.

A virtuális valóság eszközök rohamos fejlődésével egyre nagyobb igény jelentkezik azokra a megoldásokra, melyek segítségével egy valódi személy alteregója, virtuális mása könnyen elkészíthető. Az elmúlt években számos olyan, az MPEG-4 szabványt támogató, deformációalapú módszer [7, 8] jelent meg, melyek az előbb említett problémára nyújtanak megoldást. Azonban kutatásink során megállapítottuk, hogy a létező rendszerek, a használt deformációs módszereik korlátai miatt, bizonyos esetekben nem tudnak kielégítő deformációs eredményeket biztosítani. Ezért ezen területhez kapcsolódó kutatásunkban egy olyan módszer létrehozásával foglalkoztunk, mely egy fejmodell MPEG-4 szabványosítási folyamatát végzi el és az valósághűbb eredményeket produkál a létező rendszereknél, köszönhetően a baricentrikus koordinátákon alapuló deformációnak.

A dolgozat első fejezetében bemutatásra kerülnek a használt matematikai eszközök, ezt követően a baricentrikus koordináta-módszerek összehasonlítása és elemzése során elért eredményeket ismertetem. A dolgozat 3. fejezetében kerülnek közlésre a baricentrikus koordinátákon alapuló deformációs alkalmazások területén elvégzett kutatásaink eredményei, melyeket függelékként elhelyezett szemléltető ábrák és mérési eredmények egészítenek ki.

2. Matematikai háttér

Ebben a fejezetben ismertetésre kerülnek azok az alapvető matematikai fogalmak, melyek a dolgozatban bemutatott és ismertetett eredmények alapjaiul szolgáltak. A fejezetben közölt definíciók és elméletek nem képezik részét a szerző új eredményeinek, azonban kutatásunk pontos céljának és eredményeinek megfogalmazása érdekében a Hormann és Sukumar által szerkesztett könyv [9] alapján összefoglalásra kerültek.

2.1. Baricentrikus koordináták

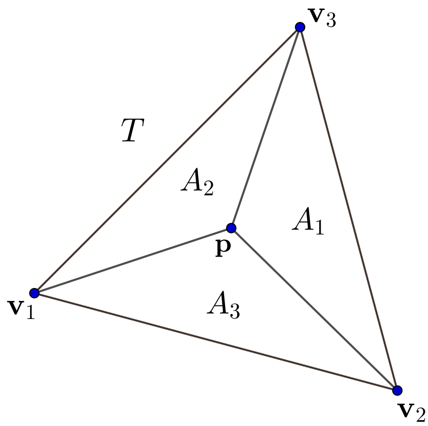
A baricentrum a csillagászatban egy olyan pontot jelöl két vagy több égitest között, melyben gravitációs erejük kiegyensúlyozza egymást. Ez a pont nem más mint az objektumok által alkotott rendszer tömegközéppontja, amely körül az égitestek keringenek. Geometriai vonatkozásban, a koordináták egy új rendszerét, a baricentrum és baricentrikus koordináták fogalmát A. F. Möbius német matematikus határozta meg 1827-ben [1].

A probléma jól szemléltethetősége érdekében tekintsünk először egy a $\mathbf{v}_1$, $\mathbf{v}_2$ és $\mathbf{v}_3$ nem kollineáris pontokkal adott T háromszöget az euklideszi síkon, ekkor bármely $\mathbf{p} \in T$ pont a háromszöget három részháromszögre osztja. A $\mathbf{p}$ pont baricentrikus koordinátái a részháromszögek A_i területeinek és az eredeti háromszög A területének az arányai alapján meghatározhatóak a következő összefüggés segítségével (lásd 1. ábra):

$$\mathbf{p} = b_1(\mathbf{p})\mathbf{v}_1 + b_2(\mathbf{p})\mathbf{v}_2 + b_3(\mathbf{p})\mathbf{v}_3, \tag{2.1}$$

ahol

$$b_i(\mathbf{p}) = \frac{A_i}{A}, \quad i = 1, 2, 3. \tag{2.2}$$



1. ábra. Egy háromszög $\mathbf{p}$ belső pontjának a baricentrikus koordinátáinak a meghatározásához használt jelölések

Hasonló módon, egy T tetraéder bármely $\mathbf{p} \in T$ pontja a tetraédert négy résztetraéderre osztja, a $\mathbf{p}$ pont baricentrikus koordinátái pedig a térfogatok arányaival kiszámíthatóak. Ezen elv alapján a baricentrikus koordináták meghatározása általánosítható a $\mathbf{v}_1, \ldots, \mathbf{v}_{d+1}$ csúcsokkal adott d dimenziós szimplexekre az alábbi módon:

$$\mathbf{p} = \sum_{i=1}^{d+1} b_i(\mathbf{p})\mathbf{v}_i, \tag{2.3}$$

ahol

$$b_i(\mathbf{p}) = \frac{V_i(\mathbf{p})}{V}, \quad i = 1, \ldots, d+1, \tag{2.4}$$

továbbá $V_i(\mathbf{p})$ a megfelelő d-szimplex térfogata, míg a $V = V_1(\mathbf{p}) + \cdots + V_{d+1}(\mathbf{p})$ az eredeti d-szimplex térfoga, amely nem függ a $\mathbf{p}$ ponttól.

A $b_i(\mathbf{p}), (i = 1, \ldots, d+1)$ függvényeket a $\mathbf{p}$ ponthoz tartozó *baricentrikus koordinátáknak* nevezzük.

2.2. Általánosított baricentrikus koordináták

Az elmúlt években számos olyan törekvés látott napvilágot, amely általánosította a baricentrikus koordináták fogalmát tetszőleges politópokra. Kutatásunk során, mi a sokszögek és a poliéderek vonatkozásában foglalkoztunk a baricentrikus koordinátákkal, így szükségesnek tartom az általánosított baricentrikus koordináták fogalmának a bevezetését Hormann és Sukumar munkája [9] alapján az alábbi módon.

2.2.1. Definíció. Legyen P egy politóp, mely a $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^d$ csúcsokkal adott, ahol $n \geq d+1$, a politóp belső pontjainak a zárt halmaza pedig legyen $\bar{P}$. A $b = [b_1, \ldots, b_n] : \bar{P} \to \mathbb{R}^n$ függvényeket *általánosított baricentrikus koordinátáknak* nevezzük, ha az alábbi feltételeket kielégítik:

$$\sum_{i=1}^{n} b_i(\mathbf{p}) = 1, \quad \forall \mathbf{p} \in \bar{P} \tag{2.5}$$

és

$$\sum_{i=1}^{n} b_i(\mathbf{p})\mathbf{v}_i = \mathbf{p}, \quad \forall \mathbf{p} \in \bar{P}. \tag{2.6}$$

Az $n = d+1$ esetben a baricentrikus koordináta-függvények meghatározása egyértelmű, az egyetlen függvény, amely a 2.2.1. definícióban ismertetett feltételeket kielégíti, az a (2.4)-es egyenletben definiált b_i függvény. Az $n > d+1$ feltétel teljesülése esetén azonban a b_i függvények meghatározása már nem egyértelmű, ennek köszönhetően számos általánosítása jelent meg a baricentrikus koordinátáknak.

A 2.2.1. definícióban ismertetett feltételeken kívül további nem kötelező, de elvárt tulajdonságok is megfogalmazhatóak a b_i függvényekre,

- *nemnegativitás*: $b_i(\mathbf{p}) \geq 0, \quad \forall \mathbf{p} \in \bar{P},$ (2.7)

- *Lagrange-tulajdonság*: $b_i(\mathbf{v}_j) = \begin{cases} 0, & \text{ha } i \neq j \\ 1, & \text{egyébként,} \end{cases}$ (2.8)
- *linearitás*: b_i lineáris az élek és a lapok mentén, (2.9)
- *folytonosság*: $b_i \in C^\infty$. (2.10)

Megjegyezzük, hogy a (2.4)-es egyenletben megadott b_i függvény az $n = d + 1$ teljesülése esetén az előbb felsorolt (2.7)–(2.10) tulajdonságok mindegyikét kielégíti.

Az általánosított baricentrikus koordinátáknak két nagy csoportját különböztetjük meg. Egyfelől beszélhetünk zárt alakkal rendelkező módszerekről, amelyek általában olyan $w = [w_1, \ldots, w_n] : P \to \mathbb{R}^n$ *súlyfüggvényeken* alapulnak, melyek kielégítik a

$$\sum_{i=1}^{n} w_i(\mathbf{p})(\mathbf{v}_i - \mathbf{p}) = 0, \quad \forall \mathbf{p} \in P \tag{2.11}$$

egyenlőséget. Ezeket a súlyfüggvényeket *homogén koordinátáknak* is szokás nevezni, ugyanis normalizálásuk után az általánosított baricentrikus koordinátákat kapjuk:

$$b_i(\mathbf{p}) = \frac{w_i(\mathbf{p})}{W(\mathbf{p})}, \quad W(\mathbf{p}) = \sum_{j=1}^{n} w_j(\mathbf{p}), \quad i = 1, \ldots, n. \tag{2.12}$$

Másfelől megkülönböztetünk olyan koordináta-módszereket is, melyek különböző numerikus módszerek felhasználásával állíthatók elő.

2.2.1. Kétdimenziós baricentrikus koordináták

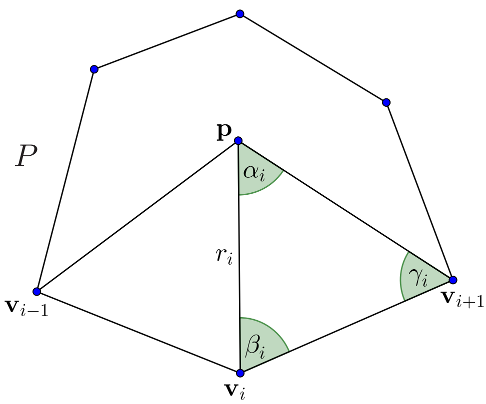
Ebben az alfejezetben a teljesség igénye nélkül szeretnék bemutatni olyan kétdimenziós általánosított baricentrikus koordináta-módszereket, melyekre a kutatásunk során kiemelt figyelmet fordítottunk néhány jellegzetes tulajdonságuk miatt. Az alfejezetben tárgyalt koordináta-módszerek mindegyike egyszerű P sokszögeken értelmezett, melyeknek $\mathbf{v}_i$ csúcsai az óramutató járásával ellentétesen adottak, továbbá feltesszük, hogy a csúcsok indexelése ciklikus, azaz $\mathbf{v}_{i+kn} = \mathbf{v}_i$, ahol $i \in \{1, \ldots, n\}$ és $k \in \mathbb{Z}$.

Wachspress-koordináták

Wachspress volt az első, aki munkájában [2] ajánlotta a baricentrikus koordináták általánosítását az $n > 3$ csúccsal rendelkező sokszögek esetén. Később Meyer és mtsai. [10] ezen *Wachspress-koordináták* egy egyszerűbb megadási módját javasolták az alábbi módon:

$$w_i = \frac{\operatorname{ctg} \gamma_{i-1} + \operatorname{ctg} \beta_i}{r_i^2}, \quad i = 1, \ldots, n, \tag{2.13}$$

ahol γ_{i-1} és β_i szögek, míg $r_i = ||\mathbf{v}_i - \mathbf{p}||$ (lásd 2. ábra).



2. ábra. Az általánosított baricentrikus koordináta-módszereknél használt jelölések a P sokszög esetén

Szigorúan konvex sokszögek esetén, minden w_i súlyfüggvényre teljesül, hogy $w_i(\mathbf{p}) > 0$, bármely $\mathbf{p} \in P$ esetén, továbbá ezen Wachspress-koordináták kielégítik a (2.7)–(2.10) tulajdonságok mindegyikét. Fontos jellemzőjük még, hogy affin invariánsak, azonban nem konvex sokszögek esetén a módszer nem jól definiált, azaz a koordináták akár negatív értékeket is felvehetnek.

Diszkrét harmonikus koordináták

*A diszkrét harmonikus koordinátáka*t [11, 12] az alábbi módon választott súlyfüggvényekkel tudjuk definiálni:

$$w_i = \operatorname{ctg} \beta_{i-1} + \operatorname{ctg} \gamma_i, \quad i = 1, \ldots, n, \tag{2.14}$$

ahol β_{i-1} és γ_i szögek (lásd 2. ábra).

Érdekes tulajdonság, hogy a diszkrét harmonikus és a Wachspress-koordináták húrsokszögek esetén megegyeznek, egyéb szigorúan konvex sokszögek tekintetében a diszkrét harmonikus koordináták a (2.8)–(2.10) tulajdonságokat teljesítik. Nem konvex sokszögek esetén nem jól definiáltak, a koordináták akár negatívak is lehetnek.

Középérték koordináták

Az általánosított baricentrikus koordináták talán egyik legnépszerűbb típusa a *középérték koordináták* [13], melyet az alábbi súlyfüggvényekkel definiálhatunk:

$$w_i = \frac{\operatorname{tg}(\alpha_{i-1}/2) + \operatorname{tg}(\alpha_i/2)}{r_i}, \quad i = 1, \ldots, n, \tag{2.15}$$

ahol α_{i-1} és α_i szögek, míg $r_i = ||\mathbf{v}_i - \mathbf{p}||$ (lásd 2. ábra).

A középérték koordináták jól definiáltak egyszerű sokszögeken, egymásba ágyazott egyszerű sokszögeken és mindenhol a síkban, továbbá kielégítik a (2.8)-as és a (2.9)-es tulajdonságot. A koordináták pozitívak bármely négyszög belsejében és folytonosak is, kivéve a sokszög csúcsaiban.

2.2.2. Háromdimenziós baricentrikus koordináták

A háromdimenziós általánosított baricentrikus koordináta-módszerek nagyon sok esetben a hozzájuk tartozó kétdimenziós koordináták egyértelmű kiterjesztései, így a háromdimenziós módszerek a kétdimenziós módszerek tulajdonságait is öröklik.

A leggyakrabban említett módszerek három dimenzióra történő általánosításai, mint például a Wachspress-, a diszkrét harmonikus, a középérték és a harmonikus koordináta-módszer részletesen megtalálható a szakirodalomban [14, 15, 16, 17]. Az előbbiekben említett módszerek közül talán a legnépszerűbb és a mi kutatásunk során is a leggyakrabban használt a középérték koordináták, így ezen módszer magasabb dimenzióra történő kiterjesztésének a módját részletezzük bővebben Hormann és Sukumar munkája [9] alapján.

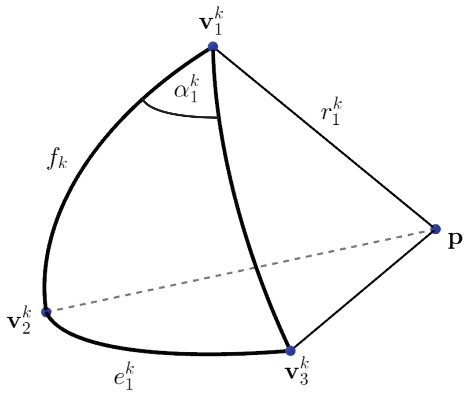
Középérték koordináták

Floater, Kós és Reimers [16], illetve Ju, Schaefer és Warren [18] egymástól függetlenül általánosították három dimenzióra a középérték koordinátákat háromszöglapokkal határolt tetszőleges poliéderekre.

Legyen P egy $\mathbf{v}_1, \ldots, \mathbf{v}_n$ csúcsokkal és az $f_1, \ldots, f_m$ háromszöglapokkal adott poliéder, továbbá legyen $f_k = \left[\mathbf{v}_1^k, \mathbf{v}_2^k, \mathbf{v}_3^k\right]$ a P egy lapja. Az f_k háromszöglap $\mathbf{p}$ középpontú egységgömbre való vetítésével egy T_k gömbháromszöget kapunk. A *háromdimenziós középérték koordináták* súlyfüggvényei az f_k háromszöglap $\mathbf{v}_j^k$ csúcsaira vonatkozóan a következő módon definiálhatóak:

$$w_j^k = \frac{e_j^k - e_{j-1}^k \cos\alpha_{j+1}^k - e_{j+1}^k \cos\alpha_{j-1}^k}{r_j^k \sin\alpha_{j+1}^k \sin\alpha_{j-1}^k}, \quad j = 1, \ldots, 3, \quad k = 1, \ldots, m, \tag{2.16}$$

ahol α_j^k szögek, e_j^k gömbi ívhosszak, míg $r_j^k = ||\mathbf{v}_j^k - \mathbf{p}||$ (lásd 3. ábra).



3. ábra. A $\mathbf{p}$ középpontú egységgömbre vetített f_k háromszöglap

2.3. Baricentrikus koordináták alkalmazásai

Az általánosított baricentrikus koordináták legfontosabb felhasználási területe az *interpoláció*. A probléma úgy fogalmazható meg, hogy ha tekintjük az $f_1, \ldots, f_n \in \mathbb{R}^d$ vektorokat a $\mathbf{v}_1, \ldots, \mathbf{v}_n$ csúcsokban, akkor keressük azt az $f : \bar{P} \to \mathbb{R}^d$ függvényt, amely az alábbi módon definiálható:

$$f(\mathbf{p}) = \sum_{i=1}^{n} b_i(\mathbf{p}) f_i. \tag{2.17}$$

Ezt az f függvényt *baricentrikus interpolációs függvény*nek nevezzük.

A (2.5)-ös és a (2.6)-os tulajdonságokból következik, hogy az f affin függvény lesz, míg a (2.8)-as tulajdonság miatt a függvény interpolálni fogja a $\mathbf{v}_i$ csúcsokban adott f_i vektorokat, továbbá, ha a

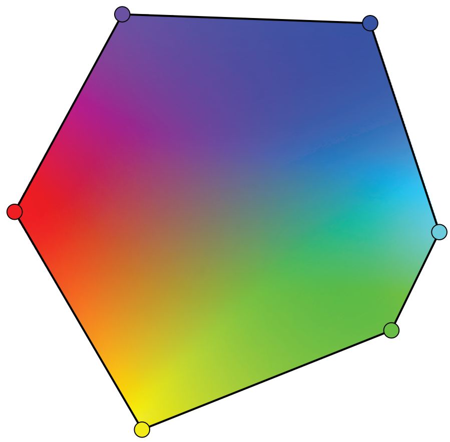
b_i koordináták teljesítik a (2.9)-es tulajdonságot, akkor az f a P élei mentén is lineáris lesz.

Az általánosított baricentrikus koordináták leggyakoribb interpolációs alkalmazásai a színinterpoláció és a képdeformáció.

2.3.1. Színinterpoláció

Legyen adott egy P sokszög, melynek $\mathbf{v}_i$ csúcsaiban tekintsük az $f_i \in [0,1]^3$ RGB színértékeket. Ezen színértékek a P sokszög belsejében interpolálhatóak a (2.17)-es egyenlet segítségével (lásd 4. ábra).

Megjegyezzük, hogy bizonyos koordináta-módszerek esetén az interpolált RGB színérték az érvényes $[0;1]$ intervallumon kívülre is eshet, ebben az esetben a közelített értéket 0-ra, illetve 1-re kerekítjük.



4. ábra. Színinterpoláció egy sokszög belsejében a középérték koordináta-módszer felhasználásával

2.3.2. Alakzat- és képdeformáció

Legyen adott egy I kép, melyet a $\mathbf{v}_1, \ldots, \mathbf{v}_n$ csúcsaival adott P sokszög határol, továbbá tekintsünk egy $\mathbf{v}'_1, \ldots, \mathbf{v}'_n$ csúcsaival adott P'

sokszöget. A (2.17)-es összefüggés segítségével az I kép deformálható, ha feltesszük, hogy $f_i = \mathbf{v}'_i$, azaz

$$f(\mathbf{p}) = \sum_{i=1}^{n} b_i(\mathbf{p})\mathbf{v}'_i. \tag{2.18}$$

Tegyük fel, hogy az eredeti I kép olyan, hogy $I : \bar{P} \to C$, ahol $\bar{P}$ a poligon belső pontjainak a zárt halmaza, míg C egy tetszőleges színtér, továbbá a deformált kép I' legyen, olyan, hogy $I' : \bar{P}' \to C$. Ebben az esetben a deformált kép a következő összefüggéssel határozható meg:

$$I'(f(\mathbf{p})) = I(\mathbf{p}). \tag{2.19}$$

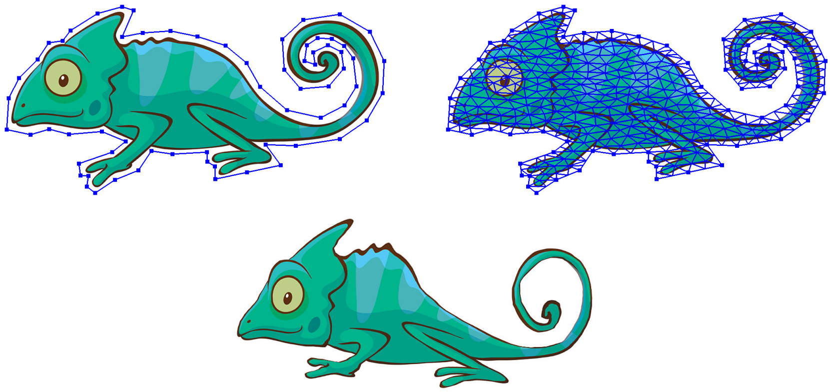
Megjegyezzük, hogy a gyakorlatban ezt a deformációt nem az eredeti kép minden egyes képpontján szokás végrehajtani, hanem egy a sokszöghöz meghatározott háromszögelés csúcsain (vertexein). Ebben az esetben a következő módon fogalmazható át a probléma.

Legyen adott egy P sokszög a $\mathbf{v}_1, \ldots, \mathbf{v}_n$ csúcsokkal és a sokszög egy T háromszögelése a $\mathbf{t}_j$ vertexekkel, továbbá tekintsük az eredeti I képet a T háromszögelés textúrájának. A P sokszög $\mathbf{v}_i$ csúcsainak a $\mathbf{v}'_i$ pozíciókba való mozgatásával az eredeti kép deformálódni fog, és a háromszögelés $\mathbf{t}_j$ vertexeinek az új $\mathbf{t}'_j$ pozíciói a (2.18)-as összefüggés átírásával határozhatóak meg:

$$\mathbf{t}'_j = \sum_{i=1}^{n} b_i(\mathbf{t}_j)\mathbf{v}'_i. \tag{2.20}$$

Mivel az I képet a T háromszögelés textúrájának tekintettük, ezért a P sokszög újrapozicionálásával az I' deformált képet kapjuk eredményül (lásd 5. ábra). Az I' kép képpontjainak a meghatározása a textúrázásból adódóan textúrafilterezési módszerekkel [19] valósítható meg.

Fontos megjegyezni, hogy a deformált kép erősen függ az eredeti P sokszögtől, az új, deformált P' sokszögtől, a háromszögeléstől és a használt koordináta-módszertől is, mivel a b_i koordináták az eredeti P sokszögre vonatkozóan kerülnek meghatározásra, melyek a későbbi deformációk alatt nem változnak.



5. ábra. Képdeformáció a középérték koordináta-módszerrel. Bal oldalon fent az eredeti I kép* a határoló P sokszöggel, jobb oldalon fent a sokszög a T háromszögeléssel, míg lent a deformált I' kép.

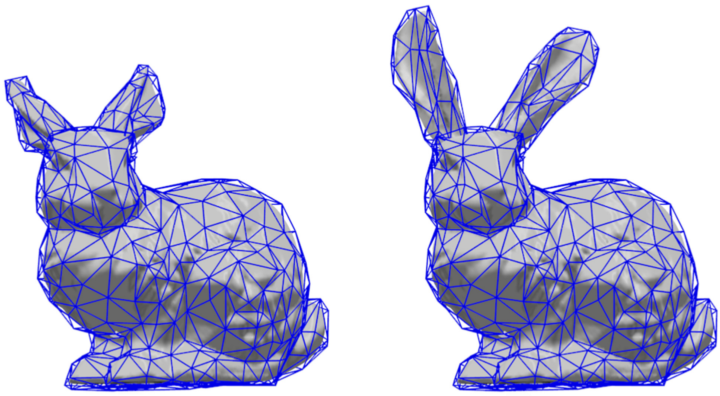
2.3.3. Modelldeformáció

A modelldeformációs módszerek az alakzat- és képdeformációs alkalmazások három dimenzióra történő kiterjesztései. Tekintsünk egy M háromdimenziós modellt, melyet egy a $\mathbf{v}_i$ csúcsaival adott $P \subset \mathbb{R}^3$ ún. ketrecmodell határol. A P ketrec $\mathbf{v}_i$ csúcsainak a $\mathbf{v}_i'$ pozíciókba való mozgatásával az eredeti M modell deformálódni fog a $P' \subset \mathbb{R}^3$ deformált ketrecmodell tekintetében (lásd 6. ábra). A (2.18)-as képlet átírásával az alábbi módon határozható meg a modelldeformáció:

*Forrás: `https://www.freepik.com/free-vector/seamless-background-design-with-green-chameleon_6609305.htm#page=1&query=lizard&position=14` Elérés időpontja: 2021. március 2.

$$\mathbf{p}' = \sum_{i=1}^{n} b_i(\mathbf{p})\mathbf{v}'_i, \qquad (2.21)$$

ahol $\mathbf{p}'$ a $\mathbf{p} \in M$ modellpont új pozíciója, míg $b_i(\mathbf{p})$ a $\mathbf{p}$ pont háromdimenziós általánosított baricentrikus koordinátája a $\mathbf{v}_i$ ketrecpontra vonatkozóan.



6. ábra. Modelldeformáció a középérték koordináta-módszerrel. Bal oldalon az eredeti M modell a határoló P ketrecmodellel, míg jobb oldalon a deformált modell a P' ketreccel.

3. Koordináta-módszerek vizsgálata

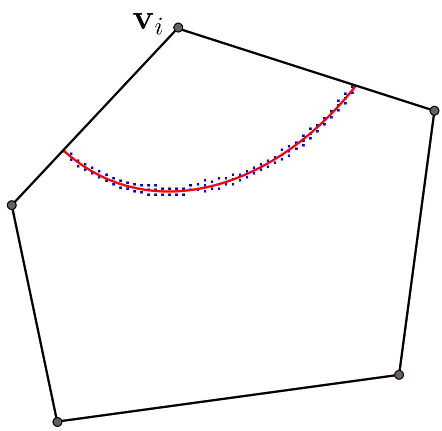
Ahogyan a bevezetésben is említésre került, az általánosított baricentrikus koordináták a 2. fejezetben bemutatott tulajdonságaik miatt napjainkban nagyon népszerűek mind a számítógépes grafika, mind a képfeldolgozás területén.

Az általánosított koordináta-módszerek kétdimenziós esetben különböző típusú sokszögek esetén eltérően viselkednek és más tulajdonságokkal rendelkeznek, így a koordináta-módszerek vizsgálata és összehasonlítása igen népszerű probléma manapság. Floater, Hormann és Kós munkájukban [3] már kitértek a koordináta-módszerek kontúrvonalakkal történő összehasonlítására, azonban mélyebb analízis nélkül tették mindezt. Kontúrvonal alatt értjük a P sokszög azon pontjait, melyeknek egy $\mathbf{v}_i$ csúcshoz tartozó baricentrikus koordináta értékei megegyeznek. Ebből kifolyólag a koordináta-módszerek az említett kontúrvonalakkal történő összehasonlítása csak egy felszínes képet ad a módszerek hasonlóságáról, illetve eltéréseiről.

Így célom volt a Wachspress-, a diszkrét harmonikus és a középérték koordináta-módszerek vizsgálata és viselkedéseik összehasonlítása n oldalú, konvex sokszögek esetén a kontúrvonalak mintázata alapján. Az összehasonlítás kiindulópontját a kontúrvonalak görbületfüggvényei jelentették, így részletesebb következtetések fogalmazhatók meg a korábban említett munkához képest. A fejezet további részében a már publikált eredményeim [20] szintézisére törekszem.

3.1. Kontúrvonalak meghatározása

Annak érdekében, hogy a különböző koordináta-módszerek kontúrvonalainak a görbületfüggvényeit összehasonlítsuk, szükséges ezen kontúrvonalak hatékony meghatározása. Mivel ezek kifejezésére nem létezik zárt alak, így egy $\mathbf{v}_i$ csúcs esetén egy $b_i \in [0; 1]$ baricentrikus koordináta-értékhez tartozó kontúrvonal alatt a P sokszög azon belső pontjait értjük, melyek b_i baricentrikus koordinátái a $\mathbf{v}_i$ csúcsra vonatkozóan megegyeznek vagy egy bizonyos, előre definiált határértéknél kisebbek. Ily módon a P sokszög belső pontjainak egy halmazát kapjuk, melyen egy görbeillesztési eljárást elvégezve meghatározhatjuk a választott b_i baricentrikus koordináta-értékhez tartozó kontúrvonalat a $\mathbf{v}_i$ csúcsra vonatkozóan (lásd 7. ábra).



7. ábra. A kék pixelek jelölik azokat a belső pontokat, melyek b_i baricentrikus koordinátái a $[0{,}3; 0{,}32]$ intervallumba esnek a $\mathbf{v}_i$ csúcsra vonatkozóan, piros színnel az illesztett görbe látható

Számos görbeillesztési módszer létezik a szakirodalomban [21], ezek közül talán az egyik legnépszerűbb eljárás a polinomillesztés, mely kutatásom alapjául is szolgált. A legjobban illeszkedő polinom meghatározásának a feladata a legkisebb négyzetek módszerével a következő módon fogalmazható meg.

Adott az $(x_i, y_i), (i = 1, \ldots, n)$ pontok halmaza, amit az

$$f(x) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0 \qquad (3.1)$$

m-edfokú polinommal közelítünk, ahol az $n \geq m + 1$ feltétel teljesül. Célunk azon legjobban közelítő $f(x)$ függvény megválasztása, amelyre a

$$\begin{aligned} &\sum_{i=1}^{n} [y_i - f(x_i)]^2 = \\ =&\sum_{i=1}^{n} \Big[y_i - \Big(a_m x_i^m + a_{m-1} x_i^{m-1} + \cdots + a_1 x_i + a_0\Big)\Big]^2 \end{aligned} \tag{3.2}$$

mennyiség minimális legyen. Az ismeretlen $a_0, a_1, a_2, \ldots, a_m$ együtthatók meghatározásával a keresett $f(x)$ függvény előállítható, a szóban forgó kontúrvonal pedig az $y = f(x)$ explicit alakban felírható.

3.2. Kontúrvonalak görbületfüggvénye

Miután meghatároztuk a különböző baricentrikus koordináta-értékekhez tartozó kontúrvonalakat, szükséges az $f(x, y) = 0$ implicit alak előállítása a fentebb említett explicit alakból, hogy a görbe egy tetszőleges x_0 pontjában a görbület könnyen meghatározható legyen az alábbi módon:

$$\kappa(x_o) = \frac{f''(x_0)}{(1 + f'(x_0)^2)^{3/2}}. \tag{3.3}$$

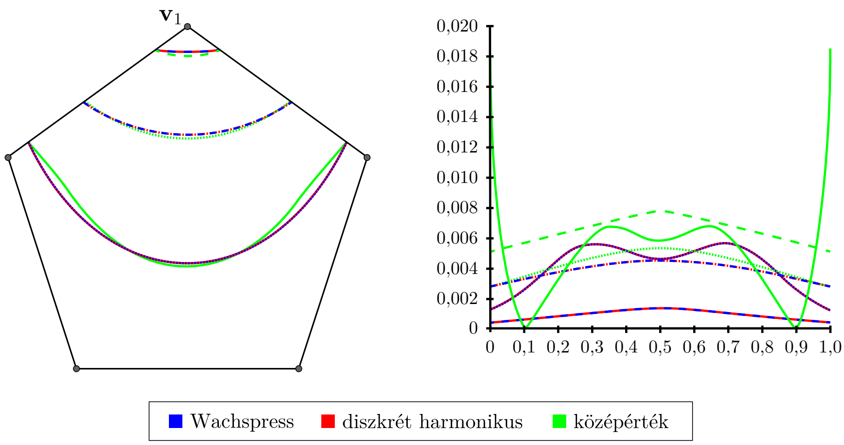
Ha a kontúrvonalak minden egyes pontjában meghatározzuk a görbületet, a kapott értékek egy görbületfüggvény grafikonjával vizualizálhatóak. Ezáltal a kontúrvonalak karakterisztikája, jellemzői könnyen kiértékelhetőek és nagyszerű alapot biztosítanak az összehasonlításhoz.

3.3. Görbületfüggvények és inflexiós pontok

Ebben az alfejezetben áttekintjük az elért eredményeket és kitérünk a vizsgált koordináta-módszerek előnyeire és hátrányaira is egyaránt.

A kutatás során a Wachspress-, a diszkrét harmonikus és a középérték koordináták viselkedése került összehasonlításra konvex, n oldalú sokszögek esetén. Minden esetben egy $\mathbf{v}_i$ csúcshoz tartozó különböző b_i baricentrikus koordináta-értékek kontúrvonalai és a hozzájuk tartozó görbületfüggvényei kerülnek kiértékelésre.

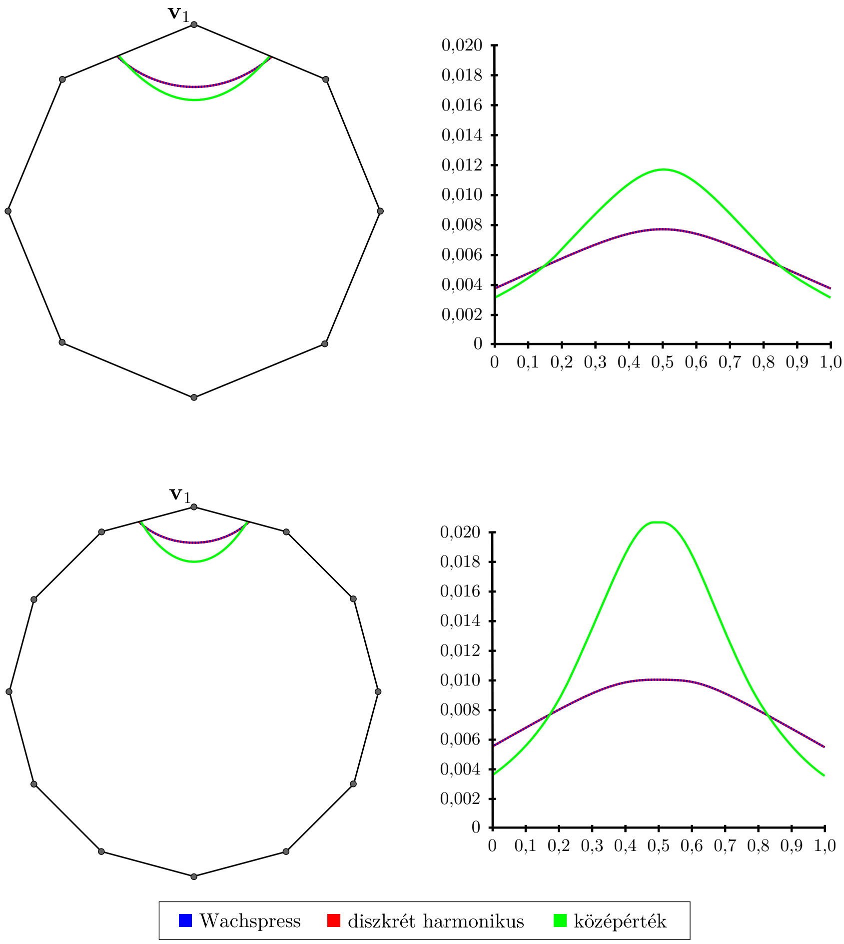
Az első példában (lásd 8. ábra) a $\mathbf{v}_1$ csúcshoz tartozó kontúrvonalakat szabályos ötszög esetén vizsgálhatjuk meg. Megfigyelhető, hogy a Wachspress- és a diszkrét harmonikus koordináták minden esetben megegyeznek, ugyanis a koordináta-módszerek húrsokszögek esetén azonosak [3].



8. ábra. Bal oldalon a baricentrikus koordináta-módszerek kontúrvonalai fentről lefelé haladva a $b_1 = 0{,}8$, a $b_1 = 0{,}4$ és a $b_1 = 0{,}1$ értéknél a $\mathbf{v}_1$ csúcs esetén, szabályos sokszögben, míg jobb oldalon a kontúrvonalak görbületfüggvényei láthatóak

Mindemellett, amikor a kontúrvonalak közel helyezkednek el a $\mathbf{v}_1$ csúcshoz, akkor a koordináta-módszerek kontúrvonalai nagyon hasonlóak egy körívhez, ezt a szinte konstans görbületfüggvény is alátámasztja. Távolabbi kontúrvonalak esetén viszont a középérték koordináták nemkívánatos inflexiós pontokat generál.

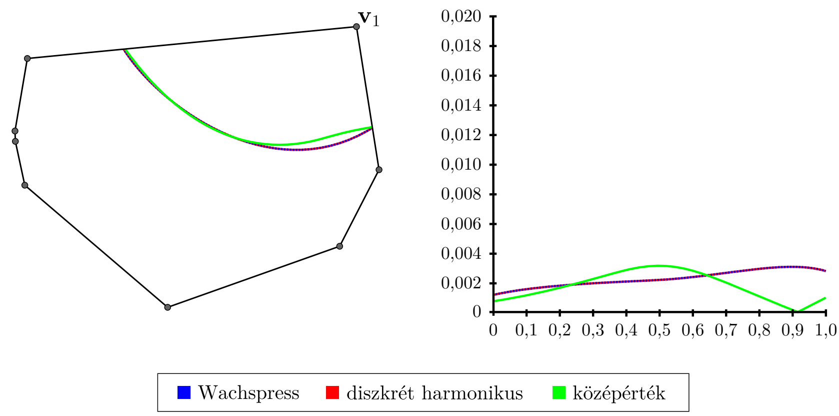
Ahogy a 9. ábrán is látszik, minél nagyobb a szabályos sokszög



9. ábra. Bal oldalon a baricentrikus koordináta-módszerek kontúrvonalai a $b_1 = 0{,}25$ értéknél a $\mathbf{v}_1$ csúcs esetén, szabályos nyolc- (fent) és tizenkétszögben (lent), míg jobb oldalon a kontúrvonalak görbületfüggvényei láthatóak

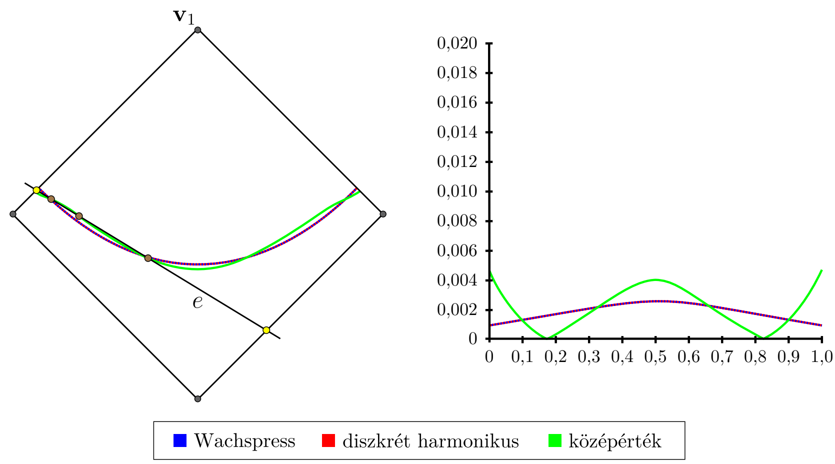
megfelelő $\mathbf{v}_1$ csúcsában a belső szög, annál nagyobb az eltérés a középérték és a másik kettő egybeeső koordináta-módszer között. Továbbá itt is elmondható az, hogy a Wachspress- és a diszkrét harmonikus koordináta-módszerek által generált kontúrvonalak sokkal közelebb állnak egy körívhez, ahogy az a görbületfüggvényen is látszik.

Megállapítható, hogy míg a középérték koordináták inflexiós pontokat generál szabályos és húrsokszögeken, addig a Wachspress- és a diszkrét harmonikus koordináta-módszer nem (lásd 10. és 11. ábra). Így feltételezhető, hogy a Wachspress- és a diszkrét harmonikus koordináta-módszer *hullámzáscsökkentő* tulajdonsággal rendelkezik húrsokszögek esetén, míg a középérték koordináták nem. A hullámzáscsökkentő tulajdonság a szabadformájú görbék jellemzője, mely azt mondja ki, hogy a görbét bármely hipersík legfeljebb annyi pontban metszi, mint a kontrollpoligonját. A mi esetünkben a sokszög veszi át a kontrollpoligon szerepét (lásd 11. ábra).



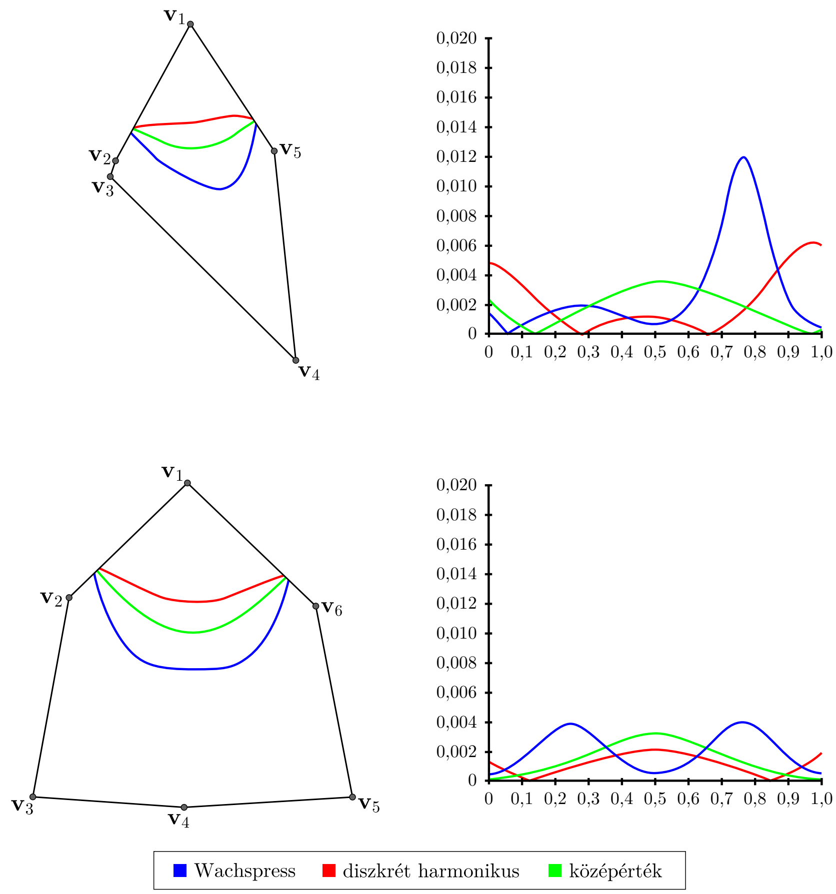
10. ábra. Bal oldalon a baricentrikus koordináta-módszerek kontúrvonalai a $b_1 = 0{,}25$ értéknél a $\mathbf{v}_1$ csúcs esetén, húrsokszögben, míg jobb oldalon a kontúrvonalak görbületfüggvényei láthatóak

Szabálytalan konvex sokszögek esetén a koordináta-módszerek szignifikánsan eltérő kontúrvonalakat produkálnak. A Wachspress-koordináta-módszer kontúrvonalai azokat a $\mathbf{v}_j$ csúcsokat közelítik, ahol a hozzátartozó C_j háromszög területe – amit az adott $\mathbf{v}_j$ és a szomszédos $\mathbf{v}_{j-1}$, $\mathbf{v}_{j+1}$ csúcsok definiálnak – nagyobb. Érdemes megvizsgálni



11. ábra. Bal oldalon a baricentrikus koordináta-módszerek kontúrvonalai a $b_1 = 0{,}15$ (lent) értéknél a $\mathbf{v}_1$ csúcs esetén, húrsokszögben, míg jobb oldalon a kontúrvonalak görbületfüggvényei láthatóak. Az e egyenes segítségével a hullámzáscsökkentő tulajdonság figyelhető meg. A sárga és barna pontok jelölik az egyenes és a sokszög, illetve az egyenes és a középérték módszer kontúrvonalának a metszéspontjait.

a kontúrvonalakhoz tartozó görbületfüggvényeket is, ugyanis a görbék viselkedését remekül leírják. A 12. ábra felső sokszögénél a $\mathbf{v}_4$ csúcshoz tartozó C_4 háromszög területe nyilvánvalóan nagyobb, mint a többi háromszög területe, így a kontúrvonal főként ezt a csúcsot közelíti, míg a görbület a $[0{,}5; 0{,}8]$ intervallumon minden kétséget kizáróan növekszik. Hasonló módon a 12. ábra alsó sokszögénél a C_3 és a C_5 háromszög területe közel megegyező, így a kontúrvonal azonos mértékben közelíti a $\mathbf{v}_3$ és a $\mathbf{v}_5$ csúcsot, továbbá mivel a C_4 háromszög területe minimális, a kontúrvonal középső része ellaposodik és egyenesként kezd el viselkedni. A görbületfüggvény szintén jellemzi a kontúrvonal ezen viselkedését, mivel a $[0{,}1; 0{,}4]$ és a $[0{,}6; 0{,}9]$ intervallumon a görbe hasonlóan viselkedik, míg az $x = 0{,}5$ értéknél a görbület nagysága közel nulla.



12. ábra. Bal oldalon a baricentrikus koordináta-módszerek kontúrvonalai a $b_1 = 0{,}25$ értéknél a $\mathbf{v}_1$ csúcs esetén, szabálytalan konvex sokszögekben, míg jobb oldalon a kontúrvonalak görbületfüggvényei láthatóak

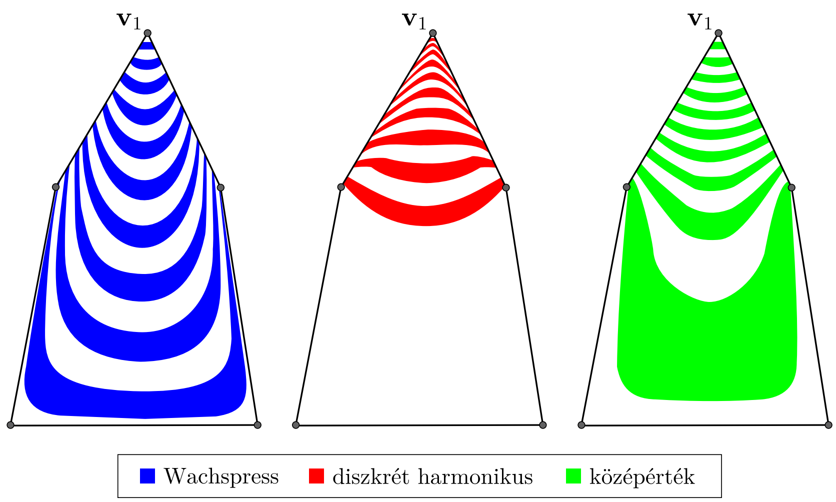
A diszkrét harmonikus koordináták kontúrvonalai közelítik a $\mathbf{v}_i$ csúcsot, ha a β_{i-1} és a γ_i szög tompaszög (lásd 2. ábra). Így például a 12. ábra felső sokszögénél ellentétesen viselkedik, mint a Wachspress-koordináta-módszer, mivel a $\mathbf{v}_2$ és a $\mathbf{v}_5$ csúcsnál a sokszög belső szögei tompaszögek. Továbbá a harmonikus koordináta-módszer gyakran generál inflexiós pontokat szabálytalan sokszögeken.

A középérték koordináta-módszer sokkal robusztusabb, kevésbé érzékeny a szabálytalan sokszögekre, mint a többi módszer (lásd 12. ábra). Ezt a megfigyelést alátámasztják a középérték koordináta-módszer hasonló kontúrvonalai és görbületfüggvényei is.

3.4. Koordináta-módszerek mintázata

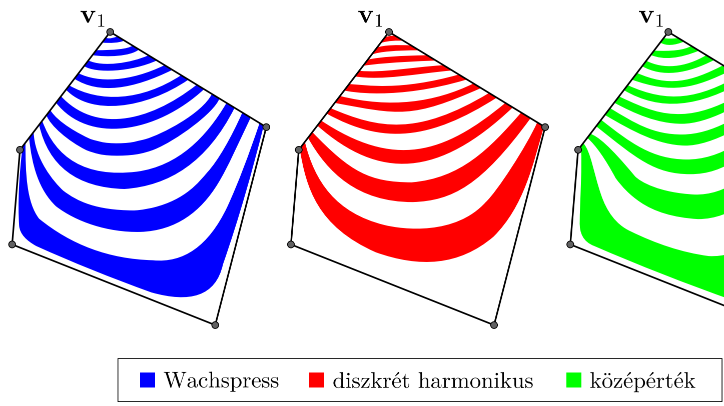
A következő példák a három különböző koordináta-módszer kontúrvonalainak mintázatát mutatják be. A kontúrvonalakat jelölő sávokhoz a $[0;1]$ intervallumból tartoznak baricentrikus koordináta-értékek 0,05 lépésközzel a $\mathbf{v}_i$ csúcsra vonatkozóan.

A 13. ábra esetében a három különböző módszer szignifikánsan eltérő mintázatot produkál. A Wachspress-koordináta-módszer generálja a legegyenletesebb eloszlását a sávoknak, ez az, amit naívan elvárunk egy ilyen típusú alakzat esetén.



13. ábra. Baricentrikus koordináta-módszerek kontúrvonalainak mintázata egy szabálytalan sokszögben a $\mathbf{v}_1$ csúcsra vonatkozóan

Észrevehető az is, hogy a középérték koordináta-módszerhez tartozó mintázatok utolsó sávjai egyértelműen vastagabbak, mint a többi módszeré, miközben a diszkrét harmonikus koordináta-módszer eltérően viselkedik, ugyanis a felső sávok a kijelölt $\mathbf{v}_i$ csúcsot közelítik (lásd 13. és 14. ábra). Ez a viselkedés annak köszönhető, hogy a $\mathbf{v}_i$ csúccsal szomszédos csúcsokhoz tartozó belső szögek tompaszögek, ugyanis, ha $\beta_{i-1} + \gamma_i > \pi$ (lásd 2. ábra), akkor a sokszögnek lehetnek olyan belső pontjai, melyeknek baricentrikus koordinátái negatívak a $\mathbf{v}_i$ csúcsra vonatkozóan [3].



14. ábra. Baricentrikus koordináta-módszerek kontúrvonalainak mintázata egy közel szabályos sokszögben a $\mathbf{v}_1$ csúcsra vonatkozóan

Továbbá, ha egy közel szabályos sokszöget tekintünk (lásd 14. ábra), akkor megfigyelhető, hogy míg a Wachspress-koordináta-módszer közel hasonlóan viselkedik, mint a 13. ábrán, addig a másik két módszer eltérően és szinte egyenletes mintázatot generálnak ebben az esetben.

3.5. Koordináta-módszerek affin kombinációja

Ahogy az előző alfejezetekben kitértünk rá, minden általánosított baricentrikus koordináta-módszernek vannak előnyei és hátrányai egyaránt. Például a diszkrét harmonikus koordináta-módszer a Dirichlet-energia [22] minimalizálásán alapszik, mégis szokatlan és szabálytalan mintázatát produkálja a kontúrvonalaknak. Minden alkalmazásban a felhasználónak kell eldöntenie, hogy melyik módszer illeszkedik a legjobban az adott problémára. Erre megoldást kínálva, definiáljuk a baricentrikus koordináták affin kombinációját a következő módon.

3.5.1. Definíció. Tekintsünk egy konvex P sokszöget a síkban, amely a $\mathbf{v}_1, \ldots, \mathbf{v}_n$ csúcsokkal adott, továbbá legyenek b^W és b^H a P sokszöghöz tartozó Wachspress-, illetve diszkrét harmonikus koordináta-függvények. Ezen koordináta-függvények $b^A = \left[b_i^A, \ldots, b_n^A\right] : \bar{P} \to \mathbb{R}^n$ affin kombinációja a

$$b_i^A = (1-\lambda)\, b_i^W + \lambda b_i^H, \tag{3.4}$$

ahol $\lambda \in [0; 1]$ egy szabad paraméter.

Az új koordináták pedig bármely $\mathbf{p} \in \bar{P}$ pont esetén kielégítik a 2.2.1. definícióban említett (2.5)-ös

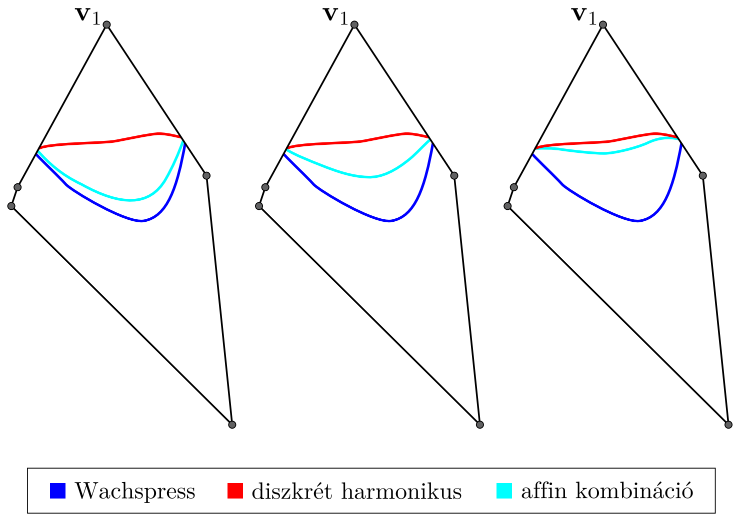
$$\begin{aligned}\sum_{i=1}^{n} b_i^A(\mathbf{p}) &= \sum_{i=1}^{n} \Big((1-\lambda) b_i^W(\mathbf{p}) + \lambda b_i^H(\mathbf{p})\Big) = \\ &= (1-\lambda) \sum_{i=1}^{n} b_i^W(\mathbf{p}) + \lambda \sum_{i=1}^{n} b_i^H(\mathbf{p}) = \\ &= (1-\lambda) + \lambda = 1\end{aligned} \tag{3.5}$$

és (2.6)-os

$$\begin{aligned}\sum_{i=1}^{n} b_i^A(\mathbf{p})\mathbf{v}_i &= \sum_{i=1}^{n} \Big((1-\lambda) b_i^W(\mathbf{p})\mathbf{v}_i + \lambda b_i^H(\mathbf{p})\mathbf{v}_i\Big) = \\ &= (1-\lambda) \sum_{i=1}^{n} b_i^W(\mathbf{p})\mathbf{v}_i + \lambda \sum_{i=1}^{n} b_i^H(\mathbf{p})\mathbf{v}_i = \\ &= \mathbf{p} - \lambda\mathbf{p} + \lambda\mathbf{p} = \mathbf{p}\end{aligned} \tag{3.6}$$

tulajdonságokat.

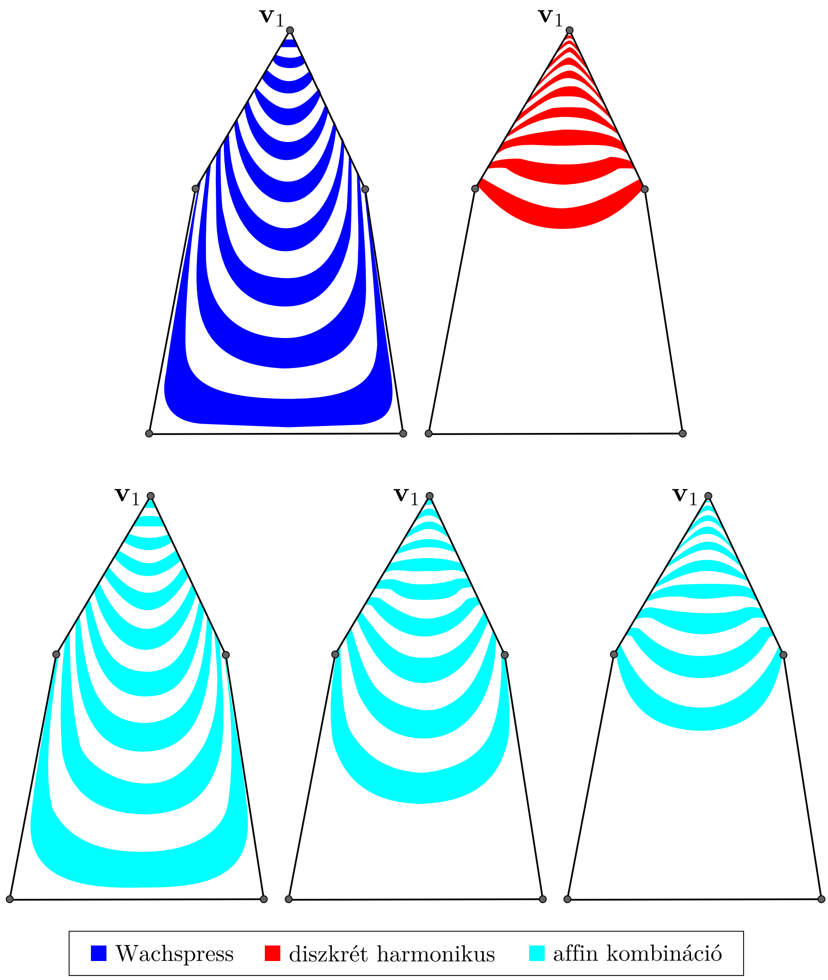
Az affin kombináció egy eszköz arra, hogy a különböző koordináta-módszerek előnyeit kombináljuk. A λ szabad paraméter megválasztása flexibilitást ad számunkra a két módszer súlyozásában, ami különböző típusú affin kombinációk létrehozását eredményezi. Továbbá egyfajta kompromisszumot biztosít az egyenletes mintázat és az energiaminimalizáció között, ahogy a 15. és a 16. ábrán is látszik.



15. ábra. A Wachspress- és a diszkrét harmonikus koordináta-módszer affin kombinációjának kontúrvonalai a $\lambda = 0{,}22$, a $\lambda = 0{,}5$ és a $\lambda = 0{,}83$ paraméterértéknél egy szabálytalan sokszögben, a $\mathbf{v}_1$ csúcsra vonatkozóan

3.6. Összegzés

A fejezetben tárgyalt általánosított baricentrikus koordináta-módszerekről elmondható, hogy szabályos és húrsokszögek esetén a Wachspress- és a diszkrét harmonikus koordináták megegyeznek és rendelkezhetnek a hullámzáscsökkentő tulajdonsággal, miközben a középérték koordináták nem kívánatos inflexiós pontokat generálhat. A



16. ábra. A Wachspress- és a diszkrét harmonikus koordinátamódszer (fent) affin kombinációjának mintázata (lent) a $\lambda = 0{,}12$, a $\lambda = 0{,}5$ és a $\lambda = 0{,}81$ paraméterértéknél egy szabálytalan sokszögben, a $\mathbf{v}_1$ csúcsra vonatkozóan

példák segítségével belátható, hogy a sokszög belső szögei nagymértékben befolyásolják a módszerek közötti eltérést. Minél nagyobb egy adott csúcshoz tartozó belső szög, annál nagyobb az eltérés a középérték és a többi koordináta-módszer kontúrvonalai között az adott csúcsra vonatkozóan.

Szabálytalan konvex sokszögek esetén a három különböző koordináta-módszer a kontúrvonalak eltérő mintázatát produkálja. Általánosan elmondható, hogy a Wachspress-koordináták közelíti a sokszög oldalait a legnagyobb mértékben, ugyanakkor egyenletes eloszlását biztosítja a kontúrvonalakhoz tartozó sávoknak, míg a diszkrét harmonikus koordináták sok esetben nem az elvárt módon viselkedik, mivel negatív koordinátákat generál. Megállapítható, hogy a középérték koordináták a legrobusztusabb a szabálytalan sokszögekben.

Annak érdekében, hogy a különböző baricentrikus koordináták hátrányait leküzdjük, definiáltuk a módszerek affin kombinációját, melyben a λ szabad paraméter további rugalmasságot ad számunkra az előnyök és a hátrányok súlyozásában.

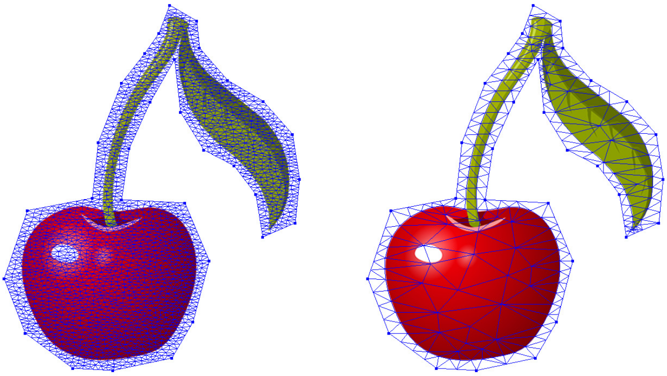
4. Deformációs módszereket támogató fejlesztések

Az általánosított baricentrikus koordináták egyik legfontosabb alkalmazási területe az interpoláció, mely segítségével különböző kép- és modelldeformációs módszereket definiálhatunk (lásd 2.3. alfejezet). Kutatásunk során az előbb említett felhasználási területeken folytattunk vizsgálódásokat és fejlesztettünk azokat támogató algoritmusokat. Ebben a fejezetben a témában elért eredményeink kerülnek bemutatásra.

4.1. Képdeformációhoz használt háromszögelési technikák

Ahogyan már említettük az általánosított baricentrikus koordináták az interpolációs tulajdonságaik miatt a színinterpoláció mellett a képdeformációs alkalmazásokban is előszeretettel használatosak. Mindamellett, hogy egyszerűen és hatékonyan számolhatóak, sima deformációs eredményeket biztosítanak.

A baricentrikus koordinátákon alapuló képdeformáció esetén a deformációt a képet körülvevő sokszög háromszögelése biztosítja. Azonban ezen háromszögelés meghatározása nem egyértelmű, mivel befolyásolja a deformációs eredményeket. Manapság az ilyen típusú képdeformációs algoritmusok a grafikus processzorra (GPU-ra) implementáltak és egyenletes háromszögelést (pl. Delaunay-háromszögelés) használnak [23]. Ha a használt egyenletes háromszögelés több ezer belső pontot tartalmaz, akkor a legtöbb esetben sima deformációt kapunk, azonban nem egyenletes háromszögelés használatával csökkenthetjük a belső pontok számát, ezzel pedig a számítási időt és költségeket is redukálhatjuk (lásd 17. ábra). Továbbá fontos megjegyezni azt, hogy a deformációs eredmény függ a kezdeti és a deformált sokszögtől, az alkalmazott koordináta-módszertől és a használt háromszögeléstől is.



17. ábra. Bemeneti képek egyenletes (balra) és nem egyenletes (jobbra) háromszögelési technikával felosztott határoló sokszögekkel

Az említett háromszögelési technikák (egyenletes és nem egyenletes) eltérő eredményeket szolgáltatnak és a részletes összehasonlításuk még nem történt meg a szakirodalomban ezen a területen. Ezért ezen kutatásunk kiemelt célja volt az, hogy meghatározzunk egy olyan nem egyenletes háromszögelési módszert, mely hasonlóan sima és pontos deformációkat eredményez, viszont a háromszögek számának a redukálásával számítási időt és költséget tud csökkenteni. Továbbá elvégeztük alapos vizsgálatukat és összevetésüket az általánosított baricentrikus koordinátákon alapuló képdeformációs módszerek tükrében. Összehasonlításuk során figyelembe vettük a bemeneti képet, az azt határoló sokszöget – amit sok esetben a felhasználó határoz meg –, illetve az alkalmazott koordináta-módszert. Mindezek mellett megvizsgáltuk azt is, hogy a különböző textúrafilterezési algoritmusok hogyan tudják növelni a deformációs eredmények minőségét. A következőekben tárgyalt eredmények a Tóth és Kunkli 2020-ban publikált cikke [24] alapján kerülnek bemutatásra.

4.1.1. Háromszögháló létrehozása

Amint a 2.3.2. alfejezetben részletesen bemutatásra került, a baricentrikus koordinátákon alapuló képdeformációs alkalmazásokban egy kép deformációja az azt határoló sokszög háromszögelésével kerül a leggyakrabban definiálásra. A sokszög csúcsainak a manipulálásával a háromszögelés pontjainak az új pozíciója a (2.20)-as összefüggéssel meghatározható, és mivel a bemeneti képet a háromszögelés textúrájaként tekintjük, a kép deformálódni fog.

Ebből kifolyólag a baricentrikus koordinátákon alapuló képdeformációs alkalmazások elengedhetetlen feladata egy kétdimenziós háromszögháló meghatározása. E háromszögháló pontosan a kezdeti sokszög által határolt területet fedi le, illetve bizonyos, a felhasználó által definiált feltételeket teljesít. A kutatásunkban vizsgált egyenletes és nem egyenletes háromszögelések esetében két feltételt fogalmaztunk meg, melyek a háromszögek alakjára és méretére vonatkoztak. A háromszögek alakjára vonatkozó feltétel egy $\boldsymbol{B}$ felső korlát, mely a háromszög köré írható kör sugara és a legrövidebb él aránya. Míg a háromszögek méretére vonatkozó feltétel egy $\boldsymbol{S}$ felső korlát, mely a háromszögek leghosszabb élének a hossza. A kezdeti sokszög által határolt terület háromszögelésére Shewchuk algoritmusát [25] használtuk, mely egy Delaunay-háromszögelés finomításán alapszik. Az algoritmus egy kényszerfeltételekkel definiált Delaunay-háromszögelésbe addig szúr be új pontokat, míg a meghatározott feltételek nem teljesülnek.

4.1.1. Definíció. Egy T háromszögelés *Delaunay-háromszögelés*, ha minden T-beli e élhez létezik egy C kör az alábbi tulajdonságokkal:

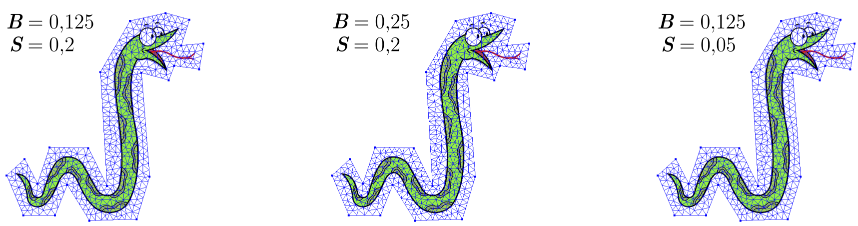
- az e él végpontjai a C kör körvonalán helyezkednek el, és
- a C kör belsejében más T-beli pont nem helyezkedik el.

4.1.2. Definíció. Legyen G egy egysíkú egyenes vonalú vonal gráf (Planar Straight Line Graph, röviden: PSLG) [26]. A G egy T háromszögelését *kényszerfeltételekkel definiált háromszögelés*nek (*constrained triangulation*) nevezzük, ha az tartalmazza a G összes élét a háromszögelés részeként. Ezen éleket *rögzített él*nek (*constrained edges*) nevezzük.

4.1.3. Definíció. A G egy *kényszerfeltételekkel definiált Delaunay-háromszögelés*e (*constrained Delaunay triangulation*, röviden: CDT) [27] alatt egy olyan *kényszerfeltételekkel definiált háromszögelés*ét értjük a G pontjainak, mely annyira *Delaunay-háromszögelés*, amennyire csak lehetséges.

Ahogy a fenti definíciókból is látszik, ha a deformációhoz használt kezdeti sokszög éleivel definiáljuk G-t, akkor Shewchuk algoritmusával egy egyenletes háromszögeléssel létrehozott kétdimenziós háromszöghálót tudunk meghatározni. Míg a nem egyenletesen háromszögelt kétdimenziós háromszögháló létrehozásához további rögzített éleket, illetve pontokat kell megadnunk a kezdeti sokszög által határolt területről és megfelelő feltételeket kell definiálnunk.

A háromszögek alakjára és méretére vonatkozó feltételek módosításával a háromszögháló létrehozásakor szabályozható a háló pontjainak a száma. A $\boldsymbol{B} = \sqrt{2}$ kezdőértékkel garantálható, hogy Shewchuk algoritmusa végrehajtható, de a $\boldsymbol{B}$ értékének a növelésével a háromszögek száma változtatható. Hasonló módon az $\boldsymbol{S}$ felső korlát módosítása hatással lesz a konstruált háromszöghálóra (lásd 18. ábra).



18. ábra. Egy kezdeti sokszöghöz létrehozott háromszöghálók különböző feltételekkel

4.1.2. Saját, nem egyenletes háromszögelési technikánk

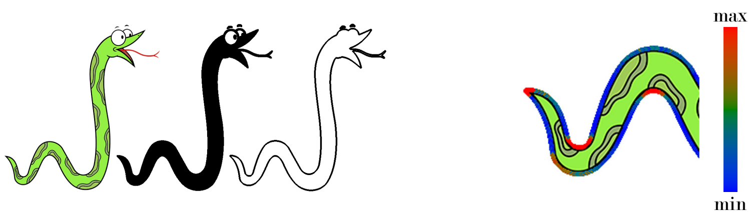
Az általánosított baricentrikus koordinátákon alapuló képdeformációs módszerekhez általunk létrehozott adaptív, nem egyenletes háromszögelés alapötlete az, hogy a hálót alkotó háromszögeket a bemeneti kép és a használt koordináta-módszer figyelembevételével helyezzük el. Így a deformáció után is megőrizhető a bemeneti kép kontúrjának a simasága, illetve csökkenthető a háromszögek száma a háromszöghálóban, ezzel pedig redukálható a számítási idő. Továbbá a bemeneti képet határoló kezdeti sokszög pozíciója sem lesz hatással a deformációs eredmények minőségére.

Az algoritmusunk bemenete egy tetszőleges alakzatot tartalmazó I bemeneti kép és az azt határoló P kezdeti sokszög, amit általában a felhasználó határoz meg egy grafikus felhasználói felület segítségével.

Kontúr meghatározása

A kezdeti sokszög meghatározása után az algoritmusunk Suzuki és Abe módszerével [28] kontúrkeresést hajt végre a bemeneti képen. Ahhoz, hogy pontos eredményeket kapjunk, célszerű a bemenetet bináris képpé konvertálni egy küszöbölő eljárással vagy egy Canny élkereső módszerrel.

A kontúrkeresés eredményeképpen egy $C = \{(x_t, y_t)\}_{t=1}^{N}$ ponthalmazt kapunk, ahol N a kontúrpontok számát jelöli (lásd 19. ábra).



19. ábra. Bal oldalon a bemeneti kép, a bináris kép és a megtalált kontúr. Jobb oldalon a kontúron vizualizált görbület. Pirossal jelöltük azokat a részeket, ahol a görbület értéke nagyobb.

Görbületszámítás

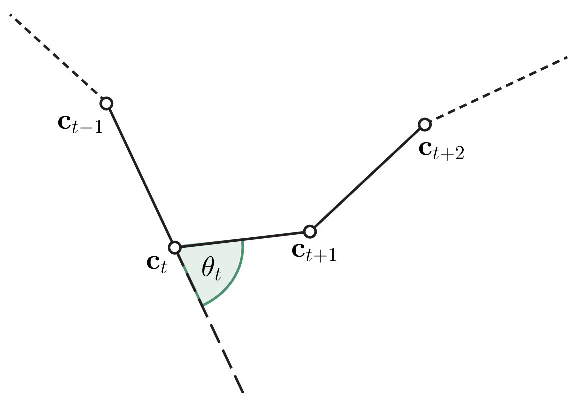
Ezt követően az algoritmusunk a bemeneti alakzat kontúrjának egy pontbeli κ_t görbületét határozza meg. A görbület a görbe egy adott pontbeli, egységhosszú érintővektorának az irányváltozását, illetve az irányváltozás nagyságát jellemzi, mely az alábbi módon számolható:

$$\kappa(t) = \frac{||\boldsymbol{r}'(t) \times \boldsymbol{r}''(t)||}{||\boldsymbol{r}'(t)^3||}. \tag{4.1}$$

Azonban a diszkrét világban, csakúgy mint a mi esetünkben, szakaszoknak egy sorozataként reprezentált az alakzat kontúrja, ezért a görbületet nem paraméterhez, hanem a C ponthalmazunk pontjaihoz rendeljük. Így a kontúr $\mathbf{c}_t \in C$ pontbeli $\widehat{\kappa}_t$ görbülete a $\mathbf{c}_t$ pontban találkozó szakaszok egymáshoz viszonyított változása az érintővektor irányában:

$$\widehat{\kappa}_t = \angle \left(\left(\mathbf{c}_t - \left(\mathbf{c}_{t-1} - \mathbf{c}_t \right) \right), \mathbf{c}_t, \mathbf{c}_{t+1} \right) = \theta_t, \tag{4.2}$$

ahol t egy pozíció a kontúron (lásd 20. ábra).



20. ábra. A diszkrét görbület meghatározásához használt jelölések

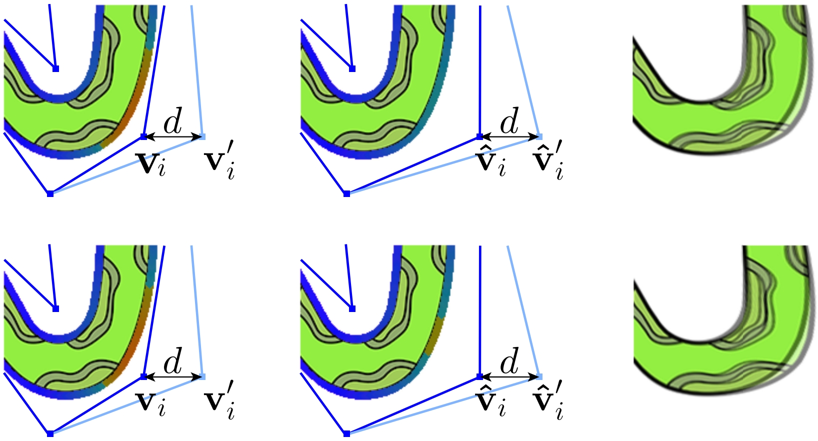
A görbület értéke nagyobb, ha az egymással szomszédos szakaszok közötti változás mértéke nagy, míg konstans értéket vesz fel, ha a kontúr egy egyenest közelít. A görbületértékek felhasználásával a kontúr mintavételezése szabályozható. Kisebb értéknél kevesebb, míg nagyobb görbületértéknél több pont mintavételezésével biztosítható, hogy a háromszögháló élei megfelelően kövessék a bemenetként megadott alakzatot. Megjegyezzük, hogy az algoritmusunk lehetővé teszi, hogy a diszkrét görbület számításakor különböző nagyságú lépésközöket használjunk, így a kontúr zaja csökkenthető.

Deformációs hatás számítása

Ahogyan korábban is említettük, a deformációs eredmények minősége nem csupán a kezdeti és a manipulált határoló sokszögtől függ, hanem a használt koordináta-módszertől is (lásd 21. ábra). Ezért az algoritmusunk meghatározza a P kezdeti sokszög legközelebbi csúcsának a hatását a $\mathbf{c}_t \in C$ pontban:

$$F(\mathbf{c}_t) = \frac{w(\mathbf{c}_t)}{\sum_{k=1}^{n} w_k(\mathbf{c}_t)}. \tag{4.3}$$

Ezen értékekkel eldönthető, hogy a bemeneti kép, mely részei fognak jobban deformálódni a kezdeti sokszög egy csúcsának a manipulálásakor. Ahogy a 21. ábra különböző soraiban látszik, ugyanolyan kezdeti sokszögek, de eltérő koordináta-módszerek mellett a deformációs hatás változik, ezt mutatja az alakzat különböző színű kontúrja. Érdemes azt is megfigyelni a 21. ábra első és második oszlopának az összevetésével, hogy ugyanazon koordináta-módszer, de különböző kezdeti sokszögek esetén a deformációs hatás szintén eltérő. Továbbá végeztünk egy deformációs tesztet mindkét koordináta-módszer, mindkét kezdeti sokszögével, a $\mathbf{v}_i$, illetve $\hat{\mathbf{v}}_i$ csúcsait a sokszögeknek eltoltuk az azonos d távolságra lévő $\mathbf{v}'_i$, illetve $\hat{\mathbf{v}}'_i$ pozíciókban. Ezen deformációk eredményeinek az összehasonlítása a 21. ábra harmadik oszlopában látható. Ez azt igazolja számunkra, hogy a deformációs eredmény erősen függ a kezdeti sokszög pozíciójától.



21. ábra. A középérték (felül) és a maximum entrópia (alul) koordináta-módszer deformációs hatása különböző kezdeti sokszögek (1. és 2. oszlop) esetén. Minél nagyobb a hatása a deformációnak, annál pirosabb a kontúr. A különböző kezdeti sokszöggel végzett deformációk eredményeinek az összehasonlítása a 3. oszlopban látható. Mindkét esetben az 1. oszlopban látható kezdeti sokszög produkálta az erősebb deformációt.

Kontúr mintavételezése

Algoritmusunk egyik legfontosabb része, hogy meghatározzuk a kontúr azon pontjait, melyek a nem egyenletes háromszögeléssel létrehozott háromszögháló alapját adják. Az algoritmus a korábbiakban kalkulált $\widehat{\kappa_t}$ és $F(\mathbf{c}_t)$ érték figyelembevételével mintavételezi a C kontúrt az alábbi módon:

$$r = \begin{cases} r_1, & \text{ha } \widehat{\kappa}_t > \Delta\widehat{\kappa} \text{ és } F(\mathbf{c}_t) > \Delta F \\ r_2, & \text{egyébként,} \end{cases} \tag{4.4}$$

ahol r_1 és r_2 mintavételezési gyakoriság változó, továbbá

$$C' = \{(x_t, y_t) \in C \mid t \pmod{r} = 0\}, \tag{4.5}$$

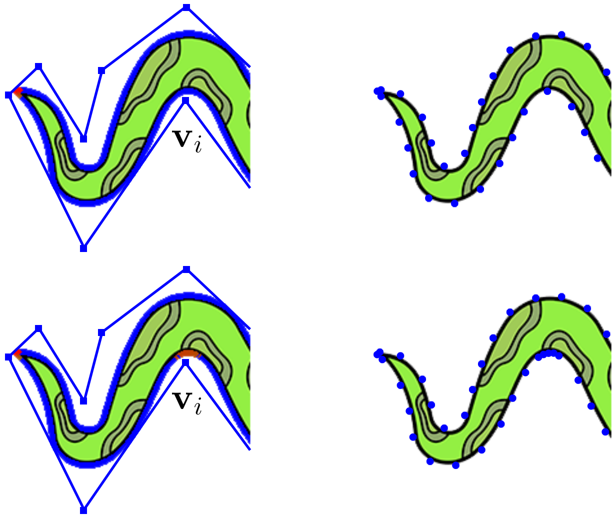
ahol r a mintavételezési arány, $\Delta\widehat{\kappa}$ és ΔF alsó korlát.

A szabadformájú görbék mintavételezésére számos módszer létezik [29]. Általános működési elvük az, hogy meghatározzák a görbe azon pontjait, melyek bizonyos feltételeket kielégítenek (pl. görbület, ívhossz, paraméterezés vagy komplexitás). Ezen módszerek közül a legnépszerűbbek az egyenletes mintavételezésen alapuló algoritmusok, melyek azonban nagyon sok esetben nem elég pontosak. Ezenkívül léteznek adaptív megoldások (ún. nem egyenletes mintavételezés) is, melyek a mintavételezés gyakoriságát megemelik a görbe azon részein, ahol pl. a görbület nagy. Ezen módszerek sokkal hatékonyabb közelítéseit adják egy-egy görbének.

Algoritmusunk nem egyenletes módon mintavételezi a bemeneti kép kontúrját a korábban meghatározott értékek figyelembevételével (lásd 22. ábra). Ezzel a célunk az, hogy a mintavételezés gyakoriságát növeljük azokon a részeken, ahol a görbület értéke nagy vagy a kezdeti sokszög közel helyezkedik el a bemeneti kép kontúrjához, miközben az egyéb részeken csökkenthető a mintavételezett pontok száma. Természetesen egyéb feltételek is megfogalmazhatóak a mintavételezésre vonatkozólag az adaptív módszernek köszönhetően.

A (4.4)-es összefüggésben az r_1 változót 4-re, az r_2-t 20-ra, a ΔF-t 0,7-re, míg a $\Delta\widehat{\kappa}$-t pontosan a kontúr átlagos görbületének az értékére állítottuk. Így, a kontúr minden negyedik pontját mintavételezzük azokon a részeken, ahol a görbület nagyobb, mint az átlag, illetve a deformációs hatás nagyobb, mint 0,7. Viszont, ha a feltételek nem teljesülnek, akkor minden 20. pontja kerül kiválasztásra a kontúrnak. Megjegyezzük, hogy ezen értékek erősen függenek a bemeneti kép felbontásától. Kutatásunkban Full HD képekkel dolgoztunk, azonban, ha az algoritmust nagyobb méretű képekkel szeretnénk használni, akkor az r_1 és r_2 értéket arányosan növelnünk kell.

Fontosnak tartjuk megemlíteni, hogy a (4.4)-es összefüggésben használt r_1 és r_2 változó, illetve $\Delta\widehat{\kappa}$ és ΔF alsó korlát értéke szabadon módosítható, így a kontúr közelítésének a finomsága egyszerűen csökkenthető vagy növelhető.

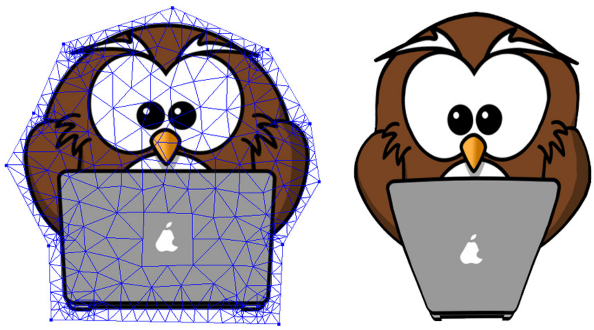


22. ábra. A bemeneti kép kontúrjának a mintavételezése középérték (felül) és metrikus (alul) koordináta-módszer használata esetén, megegyező kezdeti sokszögek mellett. Bal oldalon a bemeneti kép kontúrja a görbület és a deformációs hatás értékeinek alapján vizualizált, míg jobb oldalon a kontúr mintavételezett pontjai vannak kékkel jelölve. Az alsó képpár esetében a mintavételezés gyakoriságát a $\mathbf{v}_i$ csúcs közelében növeltük, mivel a metrikus koordináta-módszer deformációs hatása nagyobb, mint a középérték koordináta-módszeré.

Miután a bemeneti kép kontúrjának a mintavételezése megtörtént, a mintavételezett pontokat a kezdeti sokszög által határolt területhez generálandó háromszögháló rögzített pontjainak tekintjük. Így a deformáció alapjául szolgáló nem egyenletes háromszögeléssel létrehozott háromszögháló a 4.1.1. alfejezetben tárgyalt módon legenerálható. A $\boldsymbol{B}$ és $\boldsymbol{S}$ felső korlát megfelelő meghatározásával pedig a generált háromszögháló tetszőlegesen finomítható.

Lyukak a háromszöghálóban

A nem egyenletes háromszögelés másik előnye az egyenletessel szemben, hogy képes lyukakat kezelni a háromszöghálóban (lásd 23. ábra). Így a felhasználónak lehetősége van arra, hogy lyukakat – amelyek egyszerű H_i sokszögek – definiáljon a kezdeti sokszög belsejében. Ebben az esetben az algoritmusunk a H_i sokszögek éleit rögzített élekként tekinti a kényszerfeltételekkel definiált Delaunay-háromszögelésben.



23. ábra. Bal oldalon a bemeneti kép a hozzá generált háromszöghálóval, mely lyukat tartalmaz, míg jobb oldalon a deformáció eredménye

Ez a megoldás nagyon hasznos lehet akkor, ha olyan képeket vagy alakzatokat szeretnénk deformálni, melyek logót, szöveget vagy olyan területeket tartalmaznak, melyek nem sérülhetnek a deformáció következtében.

Nem egyenletes háromszögelési technikánk lépései

Algoritmusunk működése az alábbi lépésekkel foglalható össze:

1. Az I bemeneti kép, a P kezdeti sokszög és a H_i lyukak meghatározása.

2. A bemenet $C = \{(x_t, y_t)\}_{t=1}^{N}$ kontúrjának a meghatározása.
3. A kontúr $\widehat{\kappa}_t$ görbület értékeinek a meghatározása.
4. A kontúr $F(\mathbf{c}_t)$ deformációs hatás értékeinek a meghatározása.
5. A kontúr mintavételezése a $\widehat{\kappa}_t$ és az $F(\mathbf{c}_t)$ érték tekintetében.
6. A kétdimenziós háromszögháló legenerálása kényszerfeltételekkel definiált Delaunay-háromszögeléssel a C' és a H_i segítségével.

Algoritmusunk utolsó lépése után egy olyan nem egyenletes háromszögeléssel létrehozott kétdimenziós háromszöghálót kapunk, melyben a háromszögek élei követik a bemeneti kép kontúrját. Továbbá a háromszögháló azon részein, ahol a bemeneti kép kontúrjának a görbülete és a kezdeti sokszög deformációs hatása nagyobb, az algoritmusunk több háromszöget helyez el, míg az egyéb részeken minimalizálja a háromszögek számát.

4.1.3. A képdeformáció minőségének javítása

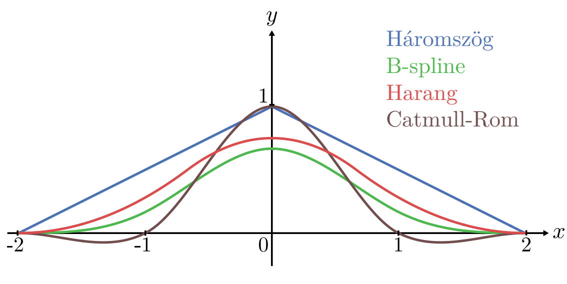
Az általánosított baricentrikus koordinátákon alapuló képdeformációs alkalmazások a legtöbb esetben az OpenGL [30] függvénykönyvtárat használják a megjelenítésre. Korábban említettük, hogy a megjelenítés alapja az, hogy a bemeneti képet a kezdeti sokszög által meghatározott háromszögháló textúrájának tekintjük. Annak érdekében, hogy az OpenGL ki tudja rajzolni a háromszöghálót a hozzárendelt textúrával, egy mintavételezési eljárást hajt végre, mely során a háromszögháló pontjait a textúrához rendeli a textúrakoordinátaértékek alapján. A mintavételezési eljárás eredményei texelek, melyek a textúraként definiált kép színinformációkat is tartalmazó pixelei, segítségükkel egy pixel végleges színe megállapítható. Azonban gyakran előfordul, hogy egy számított textúrakoordináta tört értéket vesz fel, így az OpenGL-nek meg kell határoznia, hogy az adott textúrakoordinátához melyik texel tartozzon. Ezt az eljárást textúrafilterezésnek nevezzük.

Egy kép deformálása közben általában rengeteg transzformációt hajtunk végre, így a textúra filterezési algoritmust megfelelően kell megválasztanunk. A leggyakrabban használt módszerek a `GL_NEAREST` és a `GL_LINEAR`, de ezek nem tudnak jó képminőséget biztosítani minden esetben, például, ha a deformálandó képen skálázás transzformációt hajtunk végre. A `GL_NEAREST` interpoláció annak a pixelnek a színinformációját használja, melynek a középpontja a legközelebb van a textúrakoordináta értékéhez, így az eredménykép mozaikos (ún. pixeles) lesz. A `GL_LINEAR` (vagy más néven bilineáris) interpoláció a legközelebbi négy pixel színét közelíti, így ez a módszer sokkal simább, elmosottabb képet eredményez, ahol az egyes pixelek nehezen kivehetők.

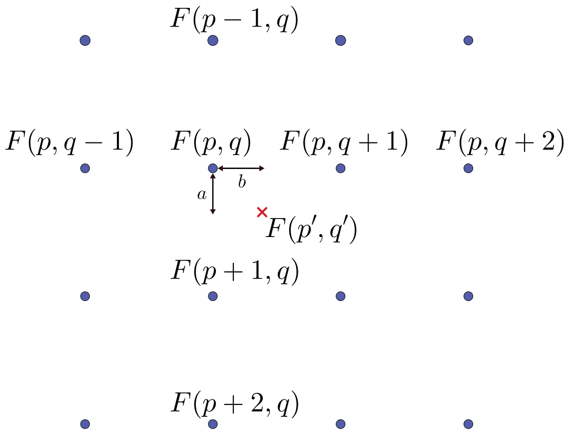
A GLSL (OpenGL Shading Language) programozási nyelv segítségével ún. shadereket írhatunk, melyekben saját filterező módszereket tudunk definiálni. Annak érdekében, hogy növeljük a deformációs eredmények minőségét, egy általánosított bikubikus interpolációs módszert [31] implementáltunk az alábbi egyenlettel:

$$F(p', q') = \sum_{m=-1}^{2} \sum_{n=-1}^{2} F(p+m, q+n) R_C\{(m-a)\} R_C\{-(n-b)\}, \quad (4.6)$$

ahol az $F(p, q)$ az interpolálandó $F(p', q')$ textúrakoordinátához legközelebb elhelyezkedő pixel, az a és b koordinátatávolságokat jelöl, míg $R_C(x)$ bikubikus interpolációs függvény (lásd 24. és 25. ábra).



24. ábra. Vizualizált interpolációs függvények



25. ábra. A bikubikus interpolációhoz használt jelölések

Miközben a `GL_LINEAR` interpoláció a legközelebbi négy pixelt használja egy textúrakoordinátához tartozó pixel meghatározásához, addig a bikubikus interpoláció a legközelebbi tizenhat pixelt. Így egy olyan textúrafilterező algoritmus használatával, melyben bikubikus interpolációt használunk, tovább növelhető a deformációs eredmények minősége (lásd 26. ábra).

(a) `GL_NEAR`

(b) `GL_LINEAR`

(c) háromszög

(d) B-spline

(e) harang

(f) Catmull-Rom

26. ábra. Különböző interpolációs függvényekkel definiált textúrafilterező algoritmusok közötti különbség

4.1.4. Elért eredmények

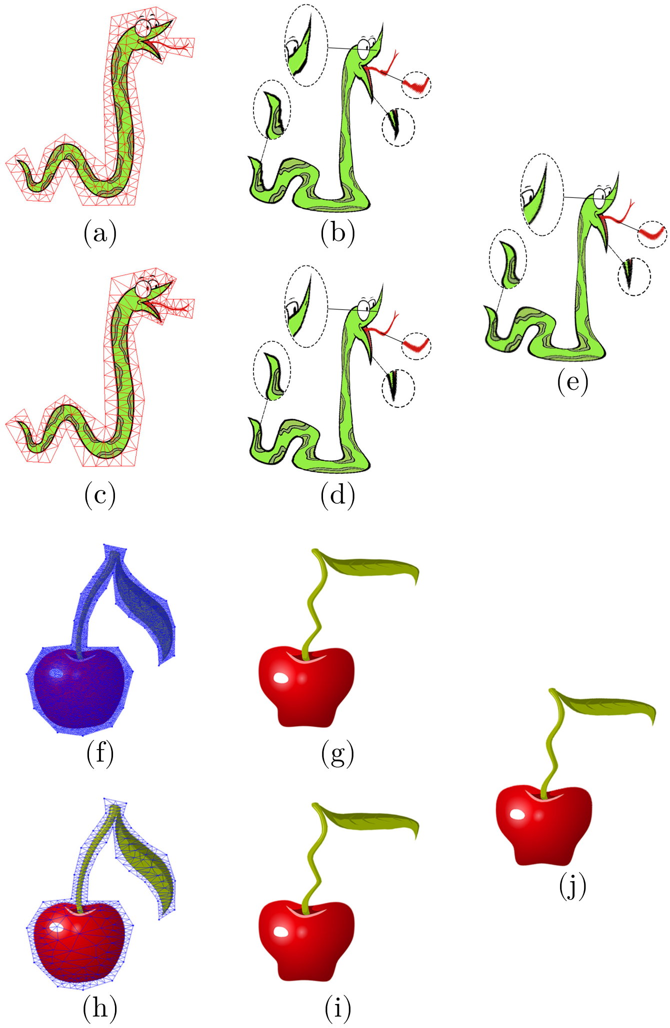
Implementáció

Az egyenletes és a nem egyenletes háromszögelési technikák prototípusát C++ programozási nyelven implementáltuk egy Intel Core i7 4,4 GHz-es processzorral és 16 GB memóriával rendelkező személyi számítógépen. A nem egyenletes háromszögelés esetében a bemeneti kép kontúrjának a meghatározásához Suzuki és Abe algoritmusát [28] használtuk, mely az OpenCV függvénykönyvtárban [32] is elérhető. A kezdeti sokszögekhez meghatározott kétdimenziós háromszöghálók generálásához egy kényszerfeltételekkel definiált Delaunay-háromszögelést [27] alkalmaztunk, mind az egyenletes, mind a nem egyenletes háromszögelés esetén. Ezen módszer egy implementációja megtalálható a CGAL függvénykönyvtárban [33].

Validáció és összehasonlítás

Az implementált algoritmusokat számos bemeneti képen kipróbáltuk, az egyenletes, illetve nem egyenletes háromszögeléssel kapott deformációs eredményeket pedig összehasonlítottuk egymással (lásd 27. ábra). A középérték [13], a maximum entrópia [34], a Poisson [35] és a *blended* [36] általánosított baricentrikus koordináta-módszereket használtuk az összehasonlításhoz. Megállapítottuk, hogy az előbbiekben említett koordináta-módszerek esetén, mind az egyenletes, mind a nem egyenletes háromszögelési technika megfelelően működik.

A bemutatott nem egyenletes háromszögelési technika a bemeneti kép külső kontúrját veszi csak figyelembe. Azonban amikor egy kevésbé részletes rajzfilmfigurát vagy egy olyan képet szeretnénk deformálni, amely nagy összefüggő területeket tartalmaz, akkor a módszer megfelelő deformációs eredményeket produkál, amellett, hogy megőrzi a bemeneti kép kontúrjának a simaságát. Továbbá, ha az algoritmust futtató platform vagy eszköz grafikus erőforrása limitált vagy csak számítási kapacitást szeretnénk spórolni a háromszögek számának a csökkentésével, akkor a nem egyenletes háromszögelési technika



27. ábra. Bemeneti képek egyenletes (a, f), illetve nem egyenletes (c, h) háromszögelési technikával felosztott határoló sokszögekkel, illetve deformációs eredményeik (b, g és d, i) a középérték koordináták használatával. A referenciaként szolgáló deformációs eredmények (e, j) az utolsó oszlopban láthatóak.

használata szintén indokolt lehet.

Másfelől, ha a baricentrikus koordinátákon alapuló képdeformációs módszer egy GPU-s implementációja rendelkezésünkre áll, illetve az erőforrásunk lehetővé teszi a háromszögek számának a megnövelését akár több ezer darabig, akkor az egyenletes háromszögelési technika is kellően sima deformációs eredményeket produkál.

Az algoritmusok validációja érdekében a bemeneti képek egy pixel alapú deformációját is elvégeztük (lásd 27. ábra). Ez egyfajta referenciaként szolgál a deformációs eredmények összehasonlításakor, ugyanis ebben az esetben a bemeneti kép minden egyes pixelét deformáltuk. Azaz az előszámítási fázisban meghatároztuk a pixelek baricentrikus koordinátáit a kezdeti sokszög vonatkozásában, a deformációs fázisban pedig minden egyes pixelnek új pozíciót számoltunk. A bemeneti kép skálázásakor ez lyukakat produkálhat, ugyanis keletkezhetnek olyan régiók a deformált képen, melyeket nem fednek pixelek, de egy – az előző alfejezetben tárgyalt – filterező algoritmussal ezek a lyukak megfelelően kitölthetőek.

Egy-egy deformációs teszt végrehajtásakor az egyenletes és a nem egyenletes háromszöghálóval kapott eredményeket a pixel alapú deformációval meghatározott referenciaeredményekkel vetettük össze. Az eredmények összehasonlítása érdekében megmértük a háromszöghálók és a baricentrikus koordináták számítási idejét, illetve a deformációs eredmények SSIM (Structural Similarity Index) [37] értékét is a referenciaeredményhez viszonyítva. Az SSIM értéket gyakran használják két kép ($\mathbf{X}$ és $\mathbf{Y}$) közötti hasonlóság definiálására az alábbi formulával:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})]^\alpha \cdot [c(\mathbf{x},\mathbf{y})]^\beta \cdot [s(\mathbf{x},\mathbf{y})]^\gamma, \tag{4.7}$$

ahol $\mathbf{x}$ és $\mathbf{y}$ ablakok az $\mathbf{X}$ és $\mathbf{Y}$ képeken, megegyező pozícióban. Az $l(\mathbf{x},\mathbf{y})$, a $c(\mathbf{x},\mathbf{y})$ és az $s(\mathbf{x},\mathbf{y})$ a luminancia-, a kontraszt-, illetve a struktúra-összehasonlító függvény. Míg az $\alpha, \beta, \gamma > 0$ paraméter segítségével szabályozható az adott komponens fontossága. Az SSIM 0 és 1 közötti értéket vehet fel, a vizsgált két kép megegyezése esetén az értéke 1.

Az egyenletes és a nem egyenletes háromszögelési technikák összehasonlítása folyamán kapott mérési eredményeket a 4.1. táblázat foglalja össze. A *kígyó* bemeneti kép esetén a két különböző háromszögelés közel megegyező számú háromszöget tartalmaz, illetve a számítási idők is megegyeznek, azonban a nem egyenletes háromszögeléssel kapott deformációs eredmény SSIM értéke nagyobb. A *cseresznye* bemeneti kép esetén, az egyenletes háromszögelés több mint 16 ezer háromszöget tartalmaz, ennek megfelelően a háromszögelés számítási ideje jelentősen megnőtt, azonban a deformációs eredmények SSIM értéke megegyezik.

	Kígyó		Cseresznye	
	E	NE	E	NE
Háromszögek száma	340	335	16185	715
SSIM	0,956	0,959	0,984	0,984
Háromszögelés számítási ideje (s)	0,001	0,001	0,563	0,003
Koordináták számítási ideje (s)	0,001	0,001	0,012	0,003

E - egyenletes, NE - nem egyenletes

4.1. táblázat. Az egyenletes és a nem egyenletes háromszögelési technika összehasonlítása

Az egyenletes és nem egyenletes háromszögelési technika összehasonlításával kapcsolatban további ábrák és mérési eredmények találhatóak meg az A. függelékben.

4.1.5. Összegzés

A fejezetben részletesen bemutattuk és összehasonlítottuk az egyenletes és a nem egyenletes háromszögelési technikát az általánosított baricentrikus koordinátákon alapuló képdeformációs módszerek tekintetében. Saját, nem egyenletes háromszögelési technikánkról elmondható, hogy az a háromszögháló létrehozásakor figyelembe veszi a kezdeti sokszöget és a használt koordináta-módszert is, továbbá képes lyukakat kezelni. A nem egyenletes háromszögeléssel létrehozott háromszögháló élei követik a bemeneti kép kontúrját – ellentétben az egyenletes háromszögeléssel –, így a deformáció után is megőrizhető a bemenet kontúrjának a simasága. A saját, nem egyenletes háromszögelési módszerünkkel a számítási idő csökkenthető a háromszögek számának a minimalizálásával, míg az egyenletes háromszögelés használata esetén a háromszögszámot jelentősen növelni kell a hasonlóan sima eredmény elérése érdekében. Mérési eredményeink alapján kijelenthető, hogy az általunk létrehozott, nem egyenletes háromszögelési technika hatékonyabb és pontosabb deformációs eredményeket produkál, mint az egyenletes háromszögelés, azokban az esetekben, amikor kevesebb háromszöggel generálunk háromszöghálót. Ugyanakkor fontos megjegyezni azt, hogy a deformációs eredmények függnek a deformált kezdeti sokszögtől, ugyanis a baricentrikus koordináták a kezdeti sokszög vonatkozásában kerülnek meghatározásra, melyek az előszámítási fázis után már nem változnak.

4.2. Deformációs ketrec generálása

4.2.1. Irodalmi áttekintés és motiváció

Napjainkban egy realisztikus, valósághű animációs jelenet elkészítésekor alapvető elvárás, hogy részletes, minél nagyobb felbontású háromdimenziós modelleket használjunk. Azonban ezen modellek közvetlen manipulációja idő- és számításigényes feladat, ezért a különböző modelldeformációs módszerek használata elterjedt a területen.

Az előbbiekben említett alkalmazások egy csoportját a ketrecalapú deformációs technikák (lásd 2.3.3. alfejezet) alkotják, melyek az általánosított baricentrikus koordinátákon alapszanak. Ezen módszerek amellett, hogy képesek valós időben működni, egy leegyszerűsített, zárt, az eredeti modellt metszések nélkül burkoló háromszögelt modellt (ún. ketrecmodell) használnak a manipulációhoz. Általában két fő fázisból áll a működésük. Az előszámítási fázisban az eredeti modell pontjainak a baricentrikus koordinátáit szükséges meghatározni a burkoló modell tekintetében. A deformációs fázisban pedig lehetőségünk van a bemenet valós idejű manipulációjára.

A deformációhoz elengedhetetlen ketrecmodelleket gyakran a felhasználó készíti el manuálisan, ami rendkívül fárasztó és időigényes feladat. A legrosszabb esetben ez akár több órás folyamat is lehet. Továbbá, ha az eredeti modell topológiája kellően komplex, akkor a ketrec elkészítése nagyon sok esetben kivitelezhetetlen manuálisan. Napjainkban olyan módszerek jelentek meg a probléma megoldására, melyek az említett ketreceket képesek automatikusan meghatározni.

A Deng, Luo és Miao által közölt módszer [4] kétdimenziós képekhez, illetve háromdimenziós háromszögelt modellekhez képes automatikusan ketreceket generálni. A megoldás két lépésből áll: első lépésben az eredeti modell kerül egyszerűsítésre egy tizedelő algoritmus segítségével, így meghatározva egy durva ketrecmodellt, majd a második lépésben az önmetszések eltávolítása történik a ketrecből Delaunay-felbontás felhasználásával. Az algoritmus eredménye egy durva ketrecmodell, mely csúcsszáma a felhasználó által meghatározható és amely burkolja az eredeti bemeneti modellt. A módszer hátránya, hogy az eredményül kapott ketrec nem tudja minden esetben pontosan követni a bemeneti modell kiálló részeinek a csatlakozásait. Továbbá a szerzők elmondása szerint a kívánt deformáció nem mindig valósítható meg pontosan a generált ketreccel.

Xian, Lin és Gao egy továbbfejlesztett OBB (oriented bounding box) fát használtak a bemeneti modell ketrecének a meghatározásához [5]. A kezdeti OBB meghatározása főkomponens-analízis (PCA) segítségével történik, amely a bemeneti modell csúcsain kerül végrehajtásra. Ezután iteratív módon minden egyes olyan OBB darabolásra kerül, mely nem elégíti ki a megállási feltételt. Ha a befoglaló dobozok nem darabolhatóak tovább, akkor egy bináris OBB fa kerül meghatározásra, melyből Boole-algebrai műveletekkel a végleges ketrecmodell meghatározható. Az algoritmus negatívuma, hogy az eredményül kapott ketrec nagymértékben függ a bemeneti modell kezdeti voxelizációjától, melyet nehézkes megfelelően meghatározni.

A Sacht, Vouga és Jacobson által létrehozott algoritmus [6] a bemenetként kapott modellek egy hierarchiáját generálja olyan módon, hogy biztosítja az egymás után következő modellpárok közötti burkolást metszés nélkül. Az algoritmus két egymást követő modell esetén – ahol az egyiket bemeneti, míg a másikat ketrecmodellnek tekintjük – a következő módon működik. Az első lépésben egy távolságminimalizálási problémát old meg a módszer, ugyanis a bemeneti modell kerül zsugorításra mindaddig, míg az összes csúcsa a durvább, azaz a ketrecmodell belsejében nem helyezkedik el. Ez a folyamat azonban a nagy görbületű felületrészeket tartalmazó modellek esetén nem minden esetben sikeres. A módszer a következő lépésben a zsugorított modell csúcsait transzformálja vissza az eredeti pozícióikba, miközben optimalizálja a burkoló modell csúcsainak a pozícióját úgy, hogy a két modell közötti távolság minimális legyen metszések nélkül. Ez a transzformáció nem megoldható egyetlen lépésben, ugyanis ez számos modellek közötti metszést indukálna párhuzamosan. Ezért a szerzők ezt iteratívan valósítják meg egy energiafüggvény minimalizálásával. Az előbb említett lépéseket végrehajtva minden egyes egymást követő modellpáron, háromszögelt modellek sorozatát kapjuk, de ezúttal minden modell burkolja a megelőzőt metszések nélkül. A módszer hátránya a számítási időigény, ugyanis a minimalizálási probléma megoldása mindkét fázisban rengeteg iterációs lépést követel. Továbbá a metszésdetektálást – ami nagymértékben növeli a futási időt – minden lépésben végre kell hajtani. A negatívumok közé sorolható az is, hogy az algoritmus első lépése nem megoldható, ha az eredeti és a ketrecmodell között túl nagy a kezdeti távolság.

Manapság a burkolási probléma egy gyakran kutatott terület, és hatékony megoldása nagyon nehéz feladat a deformációs ketrecet generáló algoritmusok körében. Annak igazolása, hogy egy ketrecmodell minden esetben burkolja a bemeneti modellt, szinte lehetetlen feladat. Az előbbiekben említett módszerek metszéseket detektálnak és heurisztikus megoldásokat használnak, mert a problémára egzakt, elméleti megoldás nem adható.

Ahogyan a leírtakból következik, a korábban bemutatott megoldások sok esetben nem elég hatékonyak, mivel gyakran numerikus módszereket használnak, melyek a számítási idő sokszoros megnövekedéséhez vezethetnek. Továbbá az is előfordulhat, hogy nem képesek kellően részletes ketrecmodellek meghatározására bizonyos bemeneti modellek esetén. Ezért ezen irányú kutatásainkban a kiemelt célunk egy olyan, az eddigi módszereknél gyorsabban számolható algoritmus létrehozása volt, mely háromdimenziós háromszögelt modellekhez képes ketrecmodelleket meghatározni automatikusan. Az algoritmus a baricentrikus koordinátákon alapszik, továbbá lehetőséget ad a felhasználónak, hogy meghatározza a ketrecmodell csúcsszámát közelítőlegesen, illetve a bemeneti modell és a ketrec közötti távolságot. A következőekben bemutatott algoritmus Tóth és Kunkli 2018-ban megjelent publikációja [38] alapján kerül ismertetésre.

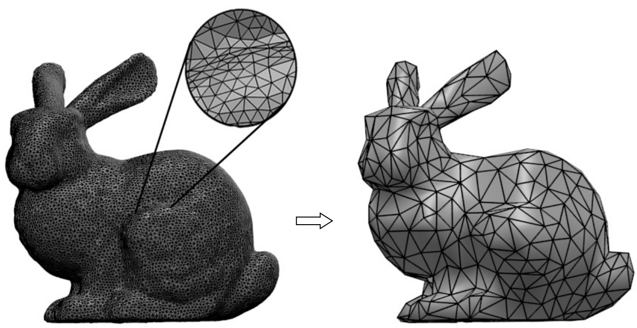
4.2.2. Saját ketrecgeneráló algoritmus

Az algoritmus bemenete egy $M = (V, F)$ általános háromdimenziós háromszögelt modell, ahol $V \in \mathbb{R}^{n\times 3}$ a csúcsok, míg $F \in \{1, \ldots, n\}^{m\times 3}$

a háromszöglapok egy halmaza. Amennyiben a bemeneti modellhez létezik a meghatározott csúcsszámú burkoló ketrec, akkor a módszer kimenete egy háromdimenziós háromszögelt ketrecmodell lesz, mely metszések nélkül burkolja a bemeneti modellt.

Egyszerűsítés

Az első opcionális lépésben a felhasználó meghatározhatja a ketrecmodell csúcsszámának a nagyságát. Ebben az esetben az algoritmus egyszerűsíti a bemeneti modellt egy tizedelőmódszerrel (lásd 28. ábra). Kutatásunkban a Garland és Heckbert által közölt modellegyszerűsítő algoritmust [39] használtuk, mely csúcspárok összehúzásán alapszik, de tetszőleges modelltizedelő algoritmus használható. Ha a felhasználó nem határozza meg a ketrecmodell kívánt csúcsszámát, akkor az algoritmus a második lépéssel folytatja a működését. A továbbiakban az eredményül kapott egyszerűsített modellre vagy az egyszerűsítés elhagyása esetén az eredeti modell egy másolatára kezdeti ketrecmodellként hivatkozunk.



28. ábra. Bal oldalon az eredeti, részletes bemeneti modell, míg az egyszerűsített kezdeti ketrecmodell a jobb oldalon látható

A deformációs ketrec meghatározása

Az algoritmus a második lépésben a kezdeti ketrecmodell csúcsaihoz új pozíciókat határoz meg. Az alapötlet a következő: ha a kezdeti ketrecmodell egy csúcsa nagy görbületű felületrészen helyezkedik el, akkor annak új pozíciója távolabb lesz a felülettől meghatározva, mint a kis görbületű felületrészeken elhelyezkedő csúcsoké. Így nagyobb esély van a későbbi lehetséges metszések elkerülésére.

Egy $\mathbf{v}_i$ kezdeti csúcs új pozíciójának a meghatározásához szükséges a csúcsot tartalmazó háromszöglapok $\mathbf{p}_{ij}$ pontjának a definiálása az alábbi módon:

$$\mathbf{p}_{ij} = \frac{\mathbf{m}_{e1} + \mathbf{m}_{e2} + \mathbf{c}_{ij} + \mathbf{v}_i}{4}, \tag{4.8}$$

ahol $\mathbf{m}_{e1}$ és $\mathbf{m}_{e2}$ a $\mathbf{v}_i$ csúcsban találkozó élek felezőpontja, míg $\mathbf{c}_{ij}$ az aktuális háromszöglap súlypontja (lásd 29. ábra).

A $\mathbf{v}_i$ csúcsot tartalmazó háromszöglapok $\mathbf{p}_{ij}$ pontjának a kiszámítása után meg kell határozni a $\mathbf{v}_{i_a}$ pontot az alábbi módon:

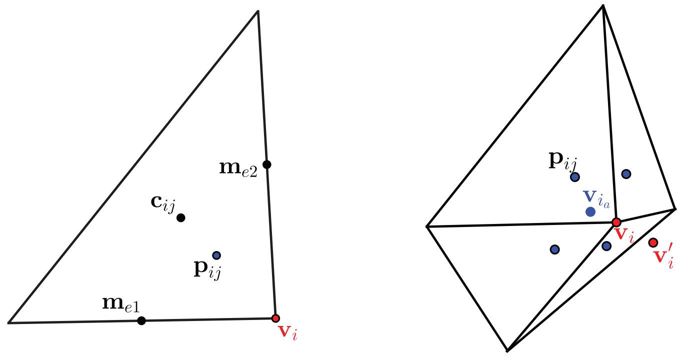
$$\mathbf{v}_{i_a} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{p}_{ij}, \tag{4.9}$$

ahol n a $\mathbf{v}_i$ csúcsot tartalmazó háromszöglapok száma.

Ezután a $\mathbf{v}_i$ kezdeti csúcs $\mathbf{v}'_i$ új pozíciója az alábbi módon határozható meg:

$$\mathbf{v}'_i = \mathbf{v}_i + \mathrm{d}(\mathbf{v}_i, \mathbf{v}_{i_a}) \cdot \mathbf{n}\left(\mathbf{v}_i\right), \tag{4.10}$$

ahol $\mathbf{n}\left(\mathbf{v}_i\right)$ olyan kifelé mutató felületi normális a $\mathbf{v}_i$ csúcsban, melyre $||\mathbf{n}\left(\mathbf{v}_i\right)|| = 1$, továbbá $\mathrm{d}(\mathbf{v}_i, \mathbf{v}_{i_a})$ a $\mathbf{v}_i$ és a $\mathbf{v}_{i_a}$ pont közötti távolság (lásd 29. ábra).



29. ábra. Bal oldalon egy a $\mathbf{v}_i$ csúcsot tartalmazó háromszöglap és a $\mathbf{p}_{ij}$ pontjának a meghatározásához szükséges jelölések, míg jobb oldalon a $\mathbf{v}_i$ csúcs $\mathbf{v}'_i$ új pozíciója

Megjegyezzük, hogy a $\mathbf{p}_{ij}$ pont tekinthető az $\mathbf{m}_{e1}$, az $\mathbf{m}_{e2}$, a $\mathbf{c}_{ij}$ és a $\mathbf{v}_i$ ponttal adott sokszög baricentrumának is. Így, ha w_1, w_2, w_3 és w_4 súlyt helyezünk el a sokszög csúcsaiban, akkor a $\mathbf{p}_{ij}$ pont pozícióját változtathatjuk a háromszöglapon. Átírva a (4.8)-as egyenletet, szabályozhatjuk a $\mathbf{v}_i$ és a $\mathbf{v}'_i$ pont közötti távolságot:

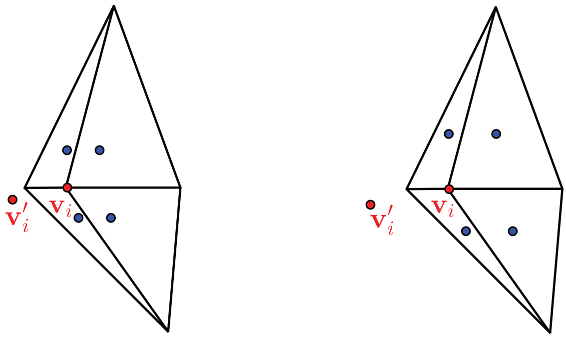
$$\mathbf{p}_{ij} = w_1\mathbf{m}_{e1} + w_2\mathbf{m}_{e2} + w_3\mathbf{c}_{ij} + w_4\mathbf{v}_i, \tag{4.11}$$

ahol $w_1 + w_2 + w_3 + w_4 = 1$.

Az algoritmus előbb említett tulajdonságából pedig következik, hogy a felhasználó szabadon meghatározhatja a bemeneti és a ketrecmodell közötti távolságot. Ez fontos szerepet játszik a modelldeformációban (lásd 30. ábra), ugyanis ezzel a távolsággal szabályozható a deformáció mértéke.

A metszések eltávolítása

Miután az algoritmus a kezdeti ketrecmodellből meghatározta az eredeti modellt burkoló ketrecet, az utolsó lépésben eltávolítja a lehetséges metszéseket. A ketrecmodell egy éle vagy egy háromszöglapja

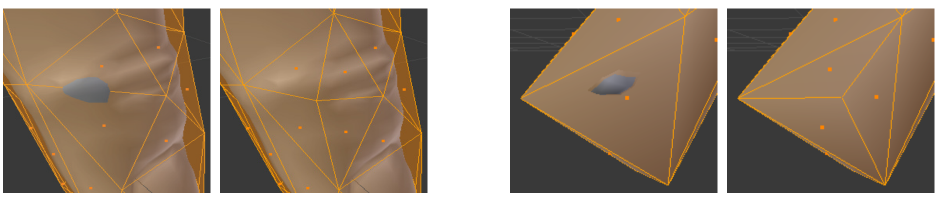


30. ábra. Különböző súlyokkal számolt $\mathbf{p}_{ij}$ pontok. Bal oldalon: $w_i = \frac{1}{4}$, $i \in \{1, 2, 3, 4\}$, míg jobb oldalon: $w_1 = \frac{2}{20}$, $w_2 = \frac{2}{20}$, $w_3 = \frac{15}{20}$ és $w_4 = \frac{1}{20}$.

metszheti a bemeneti modellt. Mindkét esetben lokálisan kerül megoldásra a metszés, de eltérő módon (lásd 31. ábra).

Az első esetben a metsző éllel szomszédos háromszöglapokat osztja fel az algoritmus négy új háromszöglappá egy új csúcs beszúrásával. Ezen új csúcsnak a pozícióját úgy határozhatjuk meg, hogy a szóban forgó háromszöglapok csúcsainak az átlagából kapott csúcsot a normálvektora mentén eltoljuk addig, míg a metszés meg nem szűnik.

A másik esetben a metsző háromszöglapot egy új csúcs beszúrásával darabolja az algoritmus. Az új csúcs a metsző háromszöglap csúcsainak az átlaga, amit addig tolunk a normálvektora mentén, amíg a metszés meg nem szűnik.



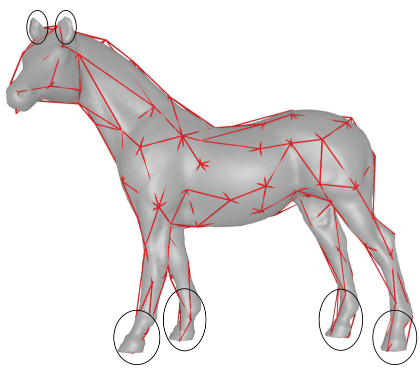
31. ábra. Bal oldalon egy metsző él és megoldása, míg jobb oldalon egy metsző háromszöglap és megoldása látható

Az algoritmus mindkét esetben iteratívan működik, így minden lépésben ellenőrizhető, hogy a metszés megszűnt-e már. A megoldás nagy előnye, hogy a metszések lokálisan kerülnek javításra egyetlen új csúcs beszúrásával további metszéseket nem indukálva. Ezenkívül a ketrecmodell csúcsainak a pozíciója nem változik, így az eredeti modell és a ketrec közötti távolság változatlan marad. A metsző élek és háromszöglapok megszüntetése után egy olyan ketrecet kapunk, mely az eredeti modellt metszések nélkül burkolja.

Korlátozások

Ahogyan korábban említettük, nem bizonyítható, hogy tetszőleges bemeneti modellhez létezik tetszőleges csúcsszámú burkoló ketrec. Jelenleg ez egy megoldatlan problémája ennek a területnek. Így az algoritmusunk bizonyos esetekben nem képes a megfelelő ketrec generálására, csakúgy, mint a már létező módszerek.

A módszerünk megáll, ha a metszések javítása közben az iterációk száma meghaladja az előre definiált korlátot. Továbbá, ha a felhasználó által kívánt ketreccsúcsszám olyan alacsony, hogy a tizedelő algoritmus csonkolja a bemeneti modell kiálló részeit (lásd 32. ábra).



32. ábra. Egy bemeneti modell és az egyszerűsítés utáni kezdetleges ketrecmodell pirossal jelölve. Az alacsony csúcsszám miatt a tizedelő algoritmus csonkolja a modell kiálló részeit.

4.2.3. Elért eredmények

Algoritmusunkat különböző topológiájú modelleken próbáltuk ki, a CAD modellektől kezdve az animációs karakterekig, és elmondható, hogy a generált ketrecmodellek alkalmasak modelldeformációs célokra. A felhasználó meghatározhatja a ketrecmodell csúcsszámának a nagyságát, illetve a bemeneti modell és a ketrec közötti távolságot. A ketrec csúcsszámának a csökkentésével a számítási idő rövidíthető, illetve megkönnyíthető az interaktív deformáció. Míg a modellek közötti távolság variálásával a deformációra lehetünk hatással. A deformáció érzékenyebb lesz, ha a távolságot minimalizáljuk, azaz egyforma súlyokat használunk a (4.11)-es egyenletben. A robusztusabb deformáció érdekében pedig a távolság növelése a cél, azaz a legkisebb súlyt a $\mathbf{v}_i$, a legnagyobbat pedig a $\mathbf{c}_{ij}$ csúcshoz kell elhelyeznünk.

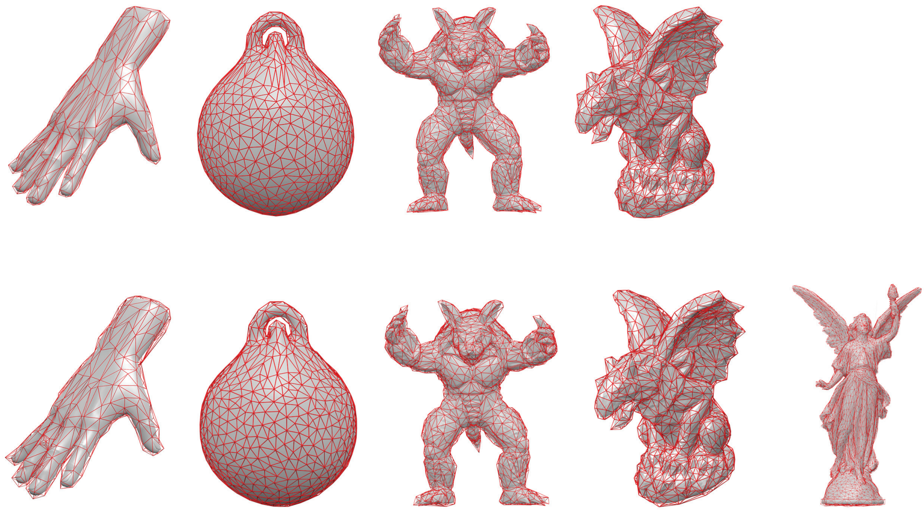
Fontosnak tartottuk algoritmusunk összevetését a szakirodalom jelenlegi meghatározó módszereivel. Így algoritmusunkat összehasonlítottuk a Sacht, Vouga és Jacobson által publikált módszerrel [6] (lásd 33. ábra). A méréseket egy Intel Core i7 4,4 GHz-es processzorral és 16 GB memóriával rendelkező személyi számítógépen végeztük, a kapott eredményeket pedig a 4.2. táblázat foglalja össze. Jól látható, hogy az általunk bemutatott algoritmus minden bemeneti modell esetén szignifikánsan gyorsabb volt, továbbá a Sacht-féle módszer az *Angel Lucy* modell esetén nem tudott ketrecet generálni.

Modell	Bemenet (L)	Ketrec (L)	Sacht-féle	Saját
Hand	8 808	600	16 s	1 s
Handles	47 104	1 500	30 s	17 s
Gargoyle	13 500	2 500	28,5 s	6 s
Armadillo	12 000	3 000	35 s	7 s
Angel Lucy	50 000	10 000	-	129 s

L - háromszöglapok száma

4.2. táblázat. Algoritmusunk összehasonlítása a Sacht-féle módszerrel. Az utolsó két oszlop a futásidőket jelöli másodpercben.

Az algoritmusunk további kimenetei és működésének megértését segítő ábrái megtalálhatóak a B. függelékben.



33. ábra. A felső sorban a Sacht, Vouga és Jacobson által közölt módszer kimenetei, míg az alsó sorban a mi algoritmusunk eredményei. A bemeneti modellek szürkével, míg a ketreceik pirossal, drótvázként megjelenítve. Az utolsó oszlopban látható az előbbiekben említett *Angel Lucy* modell, mely esetén a Sacht-féle módszer nem tudott ketrecet generálni. Az összehasonlításhoz használt modellek elérhetőek: `http://www.cs.columbia.edu/cg/nested-cages/nested-cages-supplemental-data.zip`. Elérés időpontja: 2021. március 2.

4.3. MPEG-4 szabványnak megfelelő fejmodellek

4.3.1. Irodalmi áttekintés és motiváció

Napjainkban számos kommunikációs csatorna létezik az ember és a gép között, ezeket általában billentyűzettel, egérrel vagy érintőkijelzővel vezéreljük. Azonban az elmúlt években a virtuális valóság eszközök robbanásszerű fejlődésének köszönhetően ezen eszközeink már hanggal és különböző gesztusokkal is irányíthatóak. Az említett interakciót sokkal emberközelibbnek érezhetjük, ha egyszerű karakteres üzenetek helyett egy virtuális avatárt látunk, aki beszél hozzánk a kommunikáció ideje alatt. Ebből kifolyólag a virtuális avatárok használata nagyon gyakori az ember-gép interakcióban és vélhetően a jövőben is fontos szerepet fognak majd betölteni [40].

Egy valódi vagy egy kitalált személy avatárjának az elkészítésekor a legtöbb esetben szükséges egy tetszőlegesen létrehozott háromdimenziós modell animációra való előkészítése. Erre számos megoldás létezik a komplex, manuális paraméterezéstől [41, 42] kezdve az automatikus mozgásrögzítéses módszerekig [43]. Ezen megoldások által készített avatárok legfőbb problémái az újrafelhasználhatóság és a hordozhatóság. Továbbá, mivel a felhasználásuk a legtöbb esetben különböző, a konverzió az egyes módszerek kimenetei között szinte megoldhatatlan feladat.

Az említett problémák megoldása érdekében olyan ajánlások és szabványok jelentek meg, mint például a FACS (Facial Action Coding System) [44] vagy az MPEG-4 [45], mely tartalmazza az emberi fej és test egységes animálásának a kódolási alapelveit. Az utóbbi gyakran használatos az arcanimációt megvalósító alkalmazások körében, ugyanis a segítségével a beszédanimáció is kellően részletesen és valósághűen létrehozható. Az MPEG-4 szabvány használatával garantálható az is, hogy egy szabványos modellen bármilyen, a szabvány szerint kódolt animáció lejátszható, ezzel biztosítva a hordozhatóságot és az újrafelhasználhatóságot. Számos olyan megoldás létezik [46, 47, 48, 49], mely segítségével MPEG-4 szabvány alapú arcanimációt játszhatunk le vagy egy arcmodell szabvány szerinti paraméterezését végezhetjük el manuálisan.

Egy arcmodell paraméterezése erősen függ a tekintett háromdimenziós modell topológiájától. Egy viszonylag egyszerű, csupán néhány száz csúcsot tartalmazó modell esetén ez a folyamat könnyen elvégezhető manuálisan a felhasználó által. Azonban egy részletes, több ezer csúcsot tartalmazó modell esetén a manuális kalibráció rengeteg időt követel, így az nem praktikus feladat. Egy létező paraméterezés új modellre történő adaptációja pedig sajnos nem megoldható. Ezért az elmúlt években olyan módszerek jelentek meg, melyek a szabványosítás folyamatának a felgyorsítását, illetve megkönnyítését hivatottak támogatni. A Sheng és mtsai. által publikált módszer [50] a szabványosítás folyamatában nyújt segítséget parciális differenciálegyenletek felhasználásával, azonban nem támogatja az előbbiekben említett MPEG-4 szabványt.

Az utóbbi években az MPEG-4 szabványt támogató, deformációalapú módszerek is megjelentek. Az Escher, Pandzic és Thalmann által ajánlott módszer [7] egy fejmodell MPEG-4 szabványosításában nyújt segítséget egy generikus modell felhasználásával. Miután ezen a modellen a szabvány által az animációhoz szükséges tartópontok meghatározásra kerültek, az algoritmus Dirichlet-féle szabadformájú deformációs (DFFD) módszerrel deformálja a generikus modellt, hogy az közelítse a bemenetként megadott jellemzőket, ezzel személyre szabva az említett modellt. A Lavagetto és Pockaj által bemutatott módszer [8] szintén a kalibrációs folyamatot hivatott segíteni. A tartópontokat alapul véve radiális bázisfüggvényeket (RBF) használnak egy semleges arckifejezéssel bíró arcmodell deformálásához. Ugyanakkor elmondható, hogy a fentebb említett megoldások a használt deformációs módszerek (RBF és szabadformájú deformáció) korlátai miatt nem tudnak kielégítő eredményeket produkálni (lásd 34. ábra).

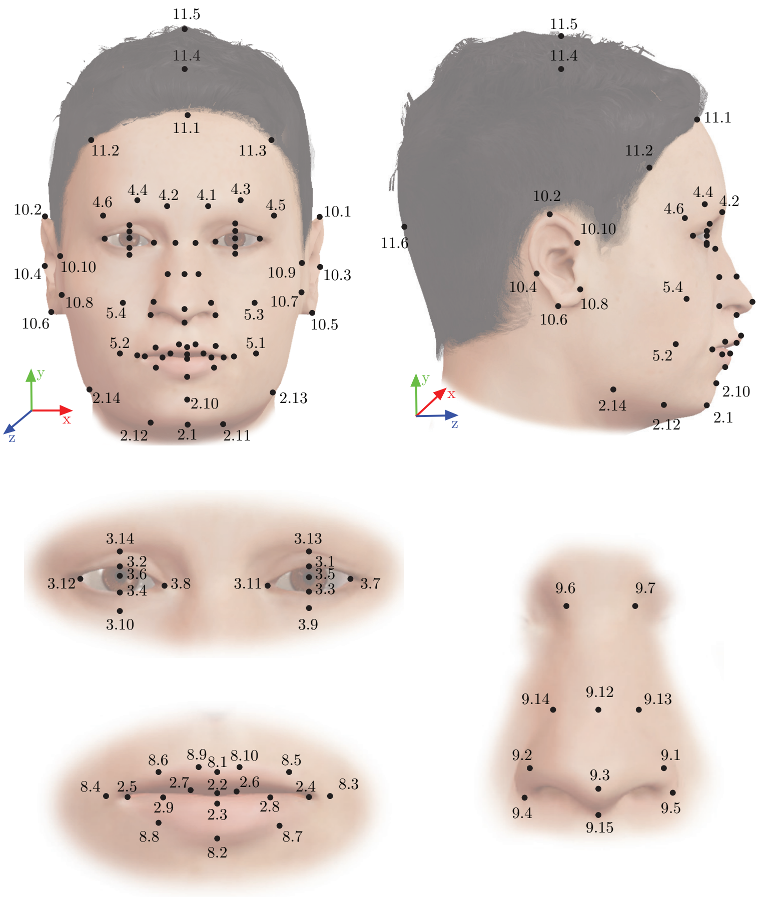
Az előzőekben említett problémák és korlátok tudatában ezen területhez kapcsolódó kutatásunk legfőbb célja volt, hogy megalkossunk egy olyan módszert, mely az MPEG-4 szabványosítási folyamatot megkönnyíti, a felhasználó által elkövethető hibák számát mi-

34. ábra. Bal oldalon a bemeneti modell, középen a szabadformájú deformációs módszerrel kapott eredmény, míg jobb oldalon az általunk is használt ketrecalapú deformációs módszer kimenete látható

nimalizálja, illetve a korábbi módszerek eredményeinél valósághűbb kimeneteket produkál. Az általunk ajánlott félautomata módszer egy általános modell deformálásával közelíti a bemenetként megadott fejmodellt. Mivel az általános modellen az MPEG-4 paraméterek már előre meghatározottak, a modell kalibrációs folyamat egy az egyben kihagyható. Az általános modell manipulációját ketrecalapú deformációs technikával végzi az algoritmusunk. Amellett, hogy ez a deformációs módszer realisztikusabb eredményeket produkál, mint a korábbi módszerek által használt deformációs megoldások, képes valós időben is működni. Algoritmusunk a 2018-ban megjelent publikációnk [51] alapján kerül bemutatásra.

4.3.2. Arcanimáció az MPEG-4 szabvánnyal

A mozgóképszakértők csoportja (Moving Picture Experts Group, továbbiakban: MPEG) 1999 márciusában mutatta be az MPEG-4 szabványt ISO szabványként, mely a hang- és a képjelek tömörítése mellett arcanimáció kódolásával is foglalkozik. Manapság ez az egyetlen széles körben elfogadott szabvány az arcanimációt megvalósító alkalmazások körében, továbbá az ipar is nagy hangsúlyt fordít rá [52]. A szabvány részletesen definiálja mindazon paramétereket, melyekkel irányítható egy emberi arcmodell animációja. Az MPEG-4 szabvány 84 tartópontot (Feature Point, továbbiakban: FP) definiál a semleges arcmodellen (lásd 35. ábra), melyek mindegyikéhez tetszőleges számú hatásköri pont megadása is szükséges. Ezen tartópontok *<csoport>.<index>* formában adottak, ahol a *csoport* az arc különböző részeit jelöli, mint például a szemek, az áll vagy a száj, míg az *index* egy egyszerű címke.



35. ábra. Az MPEG-4 szabvány által meghatározott tartópontok

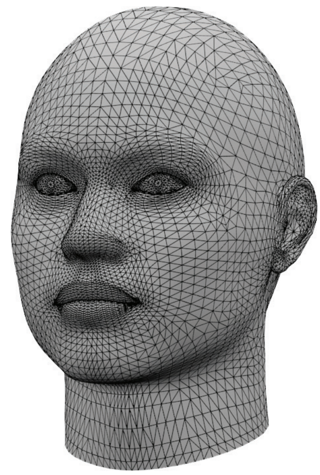
A valósághű arckifejezéseket a szabvány által megadott 68 darab arcmozgást leíró paraméter (Facial Animation Parameter, továbbiakban: FAP) határozza meg. Egy-egy FAP pontosan leírja, hogy a semleges állapotban lévő arcmodellen egy tartópontnak hogyan kell elmozdulnia a kívánt arckifejezés eléréséhez. A FAP-ok univerzális paraméterek, de a használatuk előtt kalibrálni kell őket, ami azt jelenti, hogy az általuk leírt elmozdulás nagyságát az adott emberi archoz kell igazítani. Ezen kalibráció a szabvány által meghatározott arcanimációs paraméteregységekkel (Face Animation Parameter Unit, továbbiakban: FAPU) végezhető el. A FAPU-k az emberi arcra jellemző alapvető távolságokat definiálják, mint például a szemgolyók távolsága, az orr hossza vagy a száj szélessége. Ezek alapján elmondható, hogy egy háromdimenziós arcmodell MPEG-4 szabvány szerinti animálásához elegendő csupán a modellhez tartozó tartópontok és arcanimációs paraméteregységek meghatározása. A szabvány korábban említett tulajdonságait figyelembe véve a kutatók előszeretettel alkalmazzák az MPEG-4 szabványt különböző projektekben, mint például fizikai és pszichoszociális kezelések támogatására kifejlesztett rendszerekben [53] vagy beszédszintetizáló megoldásokban [54].

4.3.3. Saját módszerünk a fejmodell-kalibráció támogatására

Ahogy korábban is említettük, egy fejmodell MPEG-4 szabvány szerinti manuális kalibrációja hosszas, fárasztó feladat, ugyanis szükséges a 84 tartópont és a hozzájuk tartozó hatásköri pontok meghatározása a modellen. Ezért egy olyan félautomata módszert terveztünk meg, mely egy háromdimenziós modell szabványosítását elvégzi. Az algoritmusunk egy előre szabványosított általános modellt manipulál egy ketrecalapú deformációs technikával úgy, hogy az közelítse a bemeneti fejmodellt. Így a modell kalibrációs folyamatot nem kell elvégeznie a felhasználónak, ezzel pedig csökkenthetjük is a szabványosításból származó esetleges hibák számát.

Általános modell

Annak érdekében, hogy valósághű beszédanimációt kapjunk, az általános modellnek szükséges teljesítenie néhány feltételt. A nagy görbületű felületrészeken kellően részletesnek kell lennie, míg az egyéb részeken tartalmazhat nagyobb poligonlapokat. Természetesen, ha a célunk egy mesebeli animációs karakter létrehozása, akkor egy kevésbé részletes általános modell praktikusabb. Mindezek mellett az alsó és felső ajaknak el kell egymástól különülniük, hogy a beszéd animálható legyen. Rendszerünkbe az általános modellt (lásd 36. ábra) az Xface nevű, nyílt forráskódú alkalmazásból [46] integráltuk. A modellt MPEG-4 szerint szabványosítottuk és módosítottuk a kinézetét, hogy egy semlegesebb arcot kapjunk, továbbá eltávolítottuk a hajat, a nyelvet és a fogakat, ezzel megkönnyítve a későbbi deformációt. Az így létrehozott általános modell a hozzá tartozó FDP (Face Definition Parameter) fájllal szabadon deformálható, az eredmény pedig felhasználható egy tetszőleges MPEG-4 szabvány alapú arcanimációs alkalmazásban.



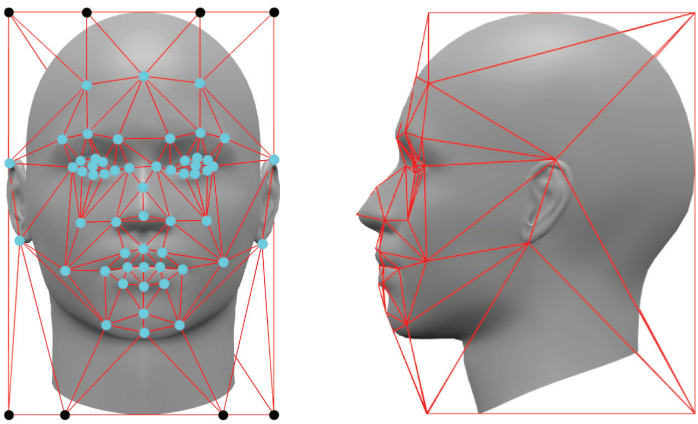
36. ábra. Az általunk használt általános modell, mely 5 570 csúcsot és 11 032 háromszöglapot tartalmaz

Bemeneti modell

A bemeneti modell bármilyen emberi fejmodell lehet a háromdimenziós rekonstrukciókból származó realisztikus modellektől kezdve egészen az animációs karakterekig. Az általános modellhez képest a bemeneti modell akár kisebb részletességű is lehet vagy akár rekonstrukciós hibákat is tartalmazhat.

Az általános modell deformációja

Az általános modell manipulációját egy ketrecalapú deformációs módszerrel, a harmonikus koordináták módszerével [17] végzi az algoritmusunk. Ahogy azt a 2.3.3. alfejezetben részletesen tárgyaltuk egy háromdimenziós modell ilyen típusú deformációjához egy topológiailag flexibilis ketrecmodellre van szükség. Ezért mind az általános, mind a bemeneti modellhez burkoló ketrecmodelleket kell meghatározni (lásd 37. ábra).



37. ábra. Az általános modellhez definiált ketrecmodell. A kék csúcsok jelölik azokat az emberi arcra jellemző pontokat, melyeket manuálisan kell meghatároznia a felhasználónak, míg a fekete csúcsokat az algoritmusunk automatikusan számolja.

Ezután, ha az általános modellt úgy manipulálja az algoritmusunk, hogy eltolja a ketrecének pontjait a hozzá tartozó (azonos indexű) bemeneti modell ketrecpontjainak a pozíciójába, akkor az általános modell közelíteni fogja a bemeneti modellt, ugyanakkor továbbra is meg fog felelni az MPEG-4 szabványnak.

A ketrecmodellek meghatározása

A ketrecmodell automatikusan generálódik, de ehhez a felhasználónak 56 darab pozíciót meg kell határoznia a fejmodellen (lásd 37. ábra). A megjelölendő pontokat az MPEG-4 szabvány tartópontjaiból származtattuk, ugyanis egy emberi fejmodell karakterisztikája a segítségükkel megfelelően leírható.

A deformációhoz használt harmonikus koordináták módszerének feltétele, hogy a deformálandó modell minden egyes pontja a ketrecen belül helyezkedjen el, ezért további 12 darab segédpontot használ az algoritmusunk a ketrecmodell definiálásához. Ezek a pontok a modell minimális és maximális koordináta-értékeiből kerülnek automatikusan meghatározásra. Ehhez fontos azt rögzíteni, hogy a modell szemeire illeszthető egyenes merőleges legyen az y és z tengelyre, továbbá az origó a modell tömegközéppontjában helyezkedjen el.

Miután megadtuk az említett pontokat, az algoritmus egy háromszögelést végez el rajtuk, ezzel létrehozva a ketrecmodellt. A háromszögelés (lásd 37. ábra) bármely modell esetén egységes, így garantálható a bemeneti és az általános modell ketrecpontjai közötti megfeleltetés. Majd egy skálázást elvégezve a ketrecen az algoritmus megszünteti a bemeneti modell és a ketrec közötti metszéseket.

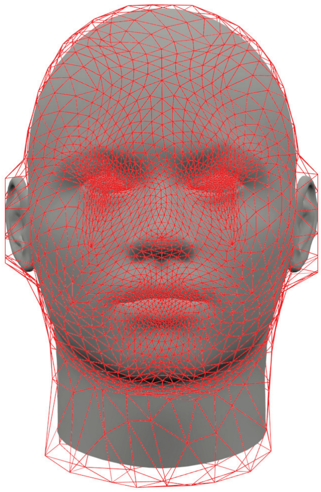
A ketrecmodellek továbbfejlesztése

Algoritmusunk szempontjából a harmonikus koordináták módszerének két fontos tulajdonsága emelhető ki. Az egyik, hogy a deformáció minősége függ a ketrecmodell csúcsainak a számától. Azaz a csúcsok számának a növelésével simább, szebb deformációt kaphatunk, ami lényeges azokon a területeken, ahol az arcanimáció központi szerepet tölt be. Így, egy területfelosztásos (subdivison) technikát alkalmazunk a ketrecen, ezzel növelve a csúcsszámot. A használt séma a következő: minden iterációs lépesben egy eredeti háromszöglap élei középpontjának az összekötésével új háromszöglapok kerülnek meghatározásra. Fontos megjegyezni, hogy ez a séma a ketrec csúcsainak az eredeti pozícióját nem változtatja.

A deformációs módszer másik fontos tulajdonsága, hogy a fejmodell és a ketrec közötti távolság csökkentésével a deformáció hatékonysága növelhető. Ezen említett távolság csökkentése céljából pedig a következőket végezzük el. Félegyeneseket definiálunk a modell origójából a ketrec csúcsain keresztül, majd az általános modell esetén, a ketrec csúcsait iterációnként egy azonos hosszúságú vektorral mozgatjuk el a hozzá tartozó félegyenesen mindaddig, amíg a modell még a ketrecen belül helyezkedik el metszések nélkül. A bemeneti modell esetén a ketrec csúcsainak az új pozíciója a hozzá tartozó félegyenes és a fejmodell metszéspontja lesz. Így a kapott ketrec metszeni fogja a bemeneti modellt, de ez ebben az esetben nem probléma, mivel a deformációt csak az általános modellen kell végrehajtani, a bemeneti modell ketrece egyfajta referenciaként szolgál a számunkra.

A bemutatott távolságcsökkentő technika nemkívánatos eredményeket produkálhat a fülek körül, legfőképp akkor, ha azok nagyon részletesek. Ez abból következik, hogy a meghatározott metszéspontok nem egyenletesen helyezkednek el ezeken a területeken, így a ketrec struktúrája változhat. A probléma megoldására a modell füleihez közel elhelyezkedő ketreccsúcsoknak a pozícióját a modell minimális, illetve maximális x koordináta-értékeiből határozzuk meg. Így a továbbiakban a sima deformáció a fülek környékén is megvalósítható.

Az említett módosítások végrehajtása után egy olyan ketrecet kapunk, mely a fejmodell kisebb felületi változásait is megfelelően leköveti, illetve minden szükséges feltételt kielégít. Az új ketrec, melyen két iterációnyi subdivision került végrehajtásra, megközelítőleg 1 500 csúcsot és 3 000 poligonlapot tartalmaz, de az alkalmazott deformációs módszer továbbra is képes valós időben működni (lásd 38. ábra).



38. ábra. Az új ketrecmodell, két iterációnyi subdivision és a távolság csökkentése után

A fejmodell kalibrációját támogató módszerünk lépései

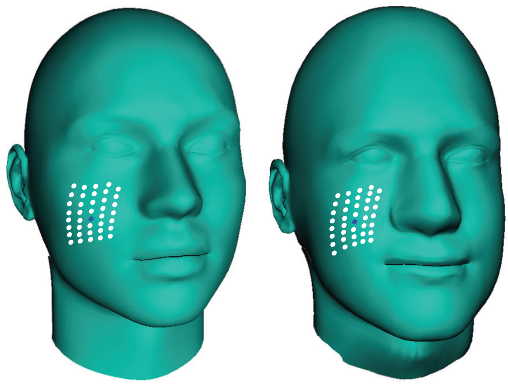
Algoritmusunk kizárólag az első két lépésben igényel felhasználói interakciót, a többiben teljesen automatikusan működik. A futtatása előtt egy általános modellel és a hozzátartozó ketreccel, illetve a megfelelő FDP fájllal kell rendelkeznünk.

1. Egy személy vagy egy fiktív karakter háromdimenziós fejmodelljének az elkészítése.
2. A szükséges pontok megjelölése a fejmodellen.
3. A ketrecmodell generálása háromszögeléssel a megjelölt és a segédpontok felhasználásával.
4. Skálázás a generált ketrecen a metszések megszüntetése végett.
5. Két iterációnyi subdivision végrehajtása a ketrecen.

6. Távolság minimalizálása a fejmodell és a ketrec között.
7. Az általános modell ketrecpontjainak eltolása a hozzájuk tartozó fejmodell ketrecpontjainak a pozíciójába.

Az algoritmusunk megértését segítő folyamatábra megtalálható a C. függelékben.

Fontos megjegyezni, hogy az általános modell deformációja erősen függ a felhasználó által az algoritmus második lépésében megjelölt pontoktól. Mindemellett a bemeneti és a deformált általános modell méretei eltérhetnek egymástól, ezt a modellek befoglaló dobozainak a méretei alapján az algoritmus javítja. A kalibrációs folyamat végére egy olyan deformált általános modellt kapunk, mely kellően hasonlít a bemenetként megadott fejmodellre, és az MPEG-4 szabvány szerinti arcanimációhoz szükség szabványpontok már definiáltak rajta (lásd 39. ábra).

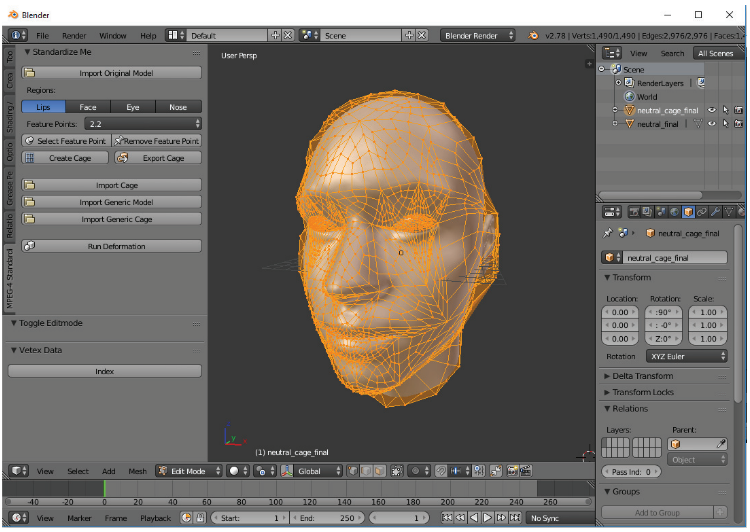


39. ábra. Bal oldalon az általános, míg jobb oldalon a deformált általános modell látható. A kék pont jelöli az 5.4-es MPEG-4 tartópontot, míg a fehér pontok annak hatásköri pontjait jelölik.

4.3.4. Elért eredmények

Implementáció

Algoritmusunkat Python szkriptként implementáltuk, mely Blender [55] kiegészítőként telepíthető. A szabadon felhasználható, *Standardize Me* [56] (lásd 40. ábra) nevet kapó szkript a teljes MPEG-4 kalibrációs folyamatot támogatja. Azaz az említett első két manuális lépésben a felhasználó betölthet egy tetszőleges modellt és jelölheti rajta a ketrecgeneráláshoz szükséges pontokat, majd a kalibrációs folyamat további lépéseit a szkript teljesen automatikusan végrehajtja.



40. ábra. A *Standardize Me* kiegészítőnk felhasználói felülete

Az algoritmusunk tesztelésére bemenetként használt háromdimenziós modellek három különböző forrásból származtak. Készítettünk saját háromdimenziós fejmodell-rekonstrukciókat egy Microsoft Kinect szenzorral, generáltunk fejmodelleket a MakeHuman [57] nyílt forráskódú modellező szoftverrel, illetve a fotóalapú rekonstrukciók készítésére alkalmas Autodesk® 123D Catch® [58] alkalmazást is használtuk. Algoritmusunkat megközelítőleg húsz különböző modellen próbáltuk ki, melyek az előzőleg említett forrásokból származtak. Összevetve ezeket a rendszereket, elmondható, hogy a MakeHuman szoftverben generált modellek adták a legjobb eredményeket, ugyanis kellően részletesek voltak, így a ketrecek generálásához megjelölendő pontok meghatározása hiba nélkül, könnyen elvégezhető volt. A MakeHuman alkalmazás azt is lehetővé tette a számunkra, hogy különböző rasszokba tartozó fejmodelleket generáljunk, így a tesztelés folyamán erre is hangsúlyt fektettünk. A másik két rekonstrukciós megoldással készített bemeneti modellek sokszor vagy nem voltak elég részletesek, vagy rekonstrukciós hibákat tartalmaztak. Így a felhasználásuk előtt ezen hibákat javítanunk kellett, a részletességük növelése érdekében pedig subdivision módszereket is alkalmaztunk.

Validáció

A bemeneti és a deformáció után kapott modellek összehasonlítása fontos szerepet játszott az algoritmusunk fejlesztésében. Az említett modellek közötti különbségek mérésére és vizualizálására a Meshlab [59] alkalmazást használtuk, melyben az alábbi formulákkal definiálható „*egyoldalú*” és „*kétoldalú*” Hausdorff-távolságokat (one-sided and two-sided Hausdorff distances) mértünk:

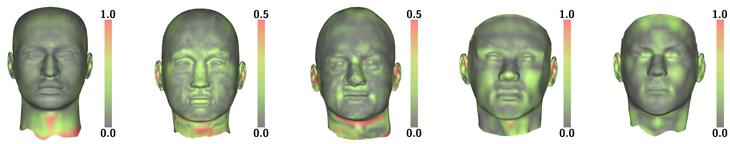
$$h(\mathcal{A},\mathcal{B}) \equiv \max_{a\in\mathcal{A}}\left(\min_{b\in\mathcal{B}} d(\mathbf{a},\mathbf{b})\right), \tag{4.12}$$

$$H(\mathcal{A},\mathcal{B}) \equiv \max(h(\mathcal{A},\mathcal{B}), h(\mathcal{B},\mathcal{A})), \tag{4.13}$$

ahol $\mathcal{A}$ az eredmény modell, $\mathcal{B}$ a bemeneti modell, míg $d(\mathbf{a},\mathbf{b})$ az $\mathbf{a}$ és $\mathbf{b}$ pont közötti euklideszi távolságot jelöli. Mérési eredményeinket a 4.3. táblázat foglalja össze.

Eredményeink validálása érdekében a BU-3DFE (Binghamton University 3D Facial Expression) adatbázist [60] is felhasználtuk, melyben valós személyek különböző arckifejezésekkel rekonstruáltak. Egy-egy személy arckifejezése alapján FAP fájlokat készítettünk, majd a

Standardize Me kiegészítőnkkel elvégeztük az MPEG-4 kalibrációs folyamatot ugyanezen személy semleges fejmodelljén. Így az eredményül kapott deformált általános modellen a létrehozott FAP fájlokat lejátszottuk és összehasonlítottuk a rekonstruált arckifejezéssel (lásd 41. ábra).



Min.	$1{,}1\times10^{-5}$	$4{,}5\times10^{-5}$	$0{,}3\times10^{-5}$	$6{,}6\times10^{-5}$	$0{,}4\times10^{-5}$
Max.	3,4282	1,2055	1,5712	1,2969	2,0151
Átlag	0,3783	0,2621	0,3083	0,2894	0,2989
Szórás	0,3175	0,1283	0,1762	0,1710	0,1959
Hausdorff	3,4282	1,6747	3,4446	1,9193	3,6731

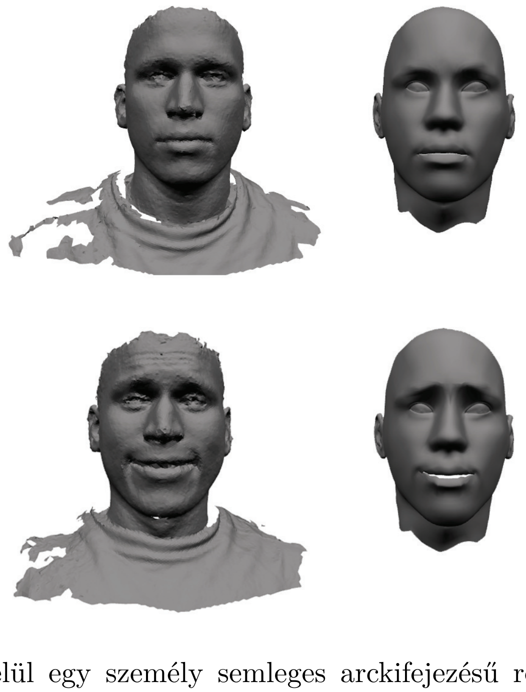
4.3. táblázat. Az eredmény és a közelített bemeneti modellek közötti összehasonlítás. A táblázat utolsó sora a „*kétoldalú*" Hausdorff-távolságokat mutatja, míg az egyéb sorok értékeit az „*egyoldalú*" Hausdorff-távolságokból határoztuk meg. Az utolsó két rekonstrukció esetén a modellek hátsó részeit levágtuk a pontos mérések végett. A jobb összehasonlíthatóság érdekében a modellek magasságai, balról jobbra, a következőek: 24,5233, 23,1186, 24,3174, 25,3531 és 23,5528.

4.3.5. Összegzés

A fejezetben részletesen bemutatott MPEG-4 kalibrációt támogató félautomata algoritmusunkról elmondható, hogy csupán a ketrecgeneráláshoz szükséges pontok megjelölésekor követel felhasználói interakciót. Ez a lépés egy meglehetősen részletes fejmodell (≈3500 csúcs, ≈7000 poligonlap) esetén körülbelül 8 percet jelent. A többi lépés teljesen automatikusan kerül végrehajtásra.

Algoritmusunk a harmonikus koordináták módszerét használja, hogy az általános modell deformálásával közelítse a bemeneti fejmodellt. Lényeges különbség az előző módszerekhez képest, hogy nem kell a felhasználónak az MPEG-4 szabványpontokat definiálnia a fejmodellen, ezért esetleges kalibrációs hibák nem befolyásolják a beszédanimáció minőségét. Úgy gondoljuk, hogy algoritmusunk más beszédanimációhoz használt paraméterezési technikával (pl. FACS) is működőképes lenne.

Validációs méréseink alapján kijelenthető, hogy a kapott eredmények kellően hasonlítanak a bemeneti modellekhez, kisebb eltérések csak a szemek, az orr és a nyak területén fordulhatnak elő.



41. ábra. Felül egy személy semleges arckifejezésű rekonstrukciója és az ezen rekonstrukció alapján készített MPEG-4 szabványos fejmodellünk. Alul ugyanezen személy boldog arckifejezésű rekonstrukciója és a fejmodellünk az ezen arckifejezés alapján generált FAP fájllal.

Algoritmusunk további összehasonlító ábrái és szemléletes kimenetei megtalálhatóak a C. függelékben.

5. Összefoglalás

A dolgozatban részletesen bemutatásra kerülnek azon kutatási eredményeink és módszereink, melyek az általánosított baricentrikus koordinátákon alapuló deformációs technikák vizsgálata közben felmerült problémákra adott válaszként jöttek létre.

Az elmúlt években a baricentrikus koordináták elméleti szakirodalma nagymértékben kibővült, illetve számos komputergrafikai alkalmazás építőköveként is rendszeresen megjelennek a kedvező tulajdonságaik miatt. Jól látható, hogy a különböző baricentrikus koordináta-módszerek kutatása az utóbbi időben központi szerepet tölt be a számítógépes grafika területén.

A 2017-ben megjelent publikációban [20] a Wachspress-, a diszkrét harmonikus és a középérték koordináta-módszerek összehasonlítását végeztem el n oldalú konvex sokszögek esetén, a kontúrvonalak mintázata alapján. Mivel az összehasonlítás során vizsgáltam a kontúrvonalak görbületfüggvényeit is, így az eddig ismert eredményeknél [3] részletesebb következtetéseket tudtam megfogalmazni. Továbbá a baricentrikus koordináta-módszerek hátrányainak kiküszöbölése érdekében definiáltam a módszerek affin kombinációját, mely segítségével két különböző módszer tulajdonságai a felhasználó által tetszőlegesen súlyozhatóak.

A baricentrikus koordináták gyakran képezik kép- és modelldeformációs alkalmazások alapját. A képdeformációs alkalmazások esetében az eredeti képet körülvevő kezdeti sokszög háromszögelése biztosítja a deformációt. A 2020-ban publikált cikkünkben [24] megvizsgáltuk a kezdeti sokszög háromszögelésének lehetőségeit. Mérési eredményeink alapján elmondhatjuk, hogy a kezdeti sokszög egyenletes, kellően sok háromszöget tartalmazó háromszögelésével sima deformációs eredményeket kapunk. Ugyanakkor, ezen kutatásunkban létrehozott nem egyenletes háromszögelési technikánk használatával erőforrást és számítási időt tudunk megtakarítani, miközben nagyon sok esetben hasonlóan sima és valósághű deformációs eredményeket kapunk.

Az általánosított baricentrikus koordinátákon alapuló modelldeformációs módszerek esetében, az ún. ketrecmodell felelős a manipulációért. Ezen ketrecmodell hatékony meghatározása komoly számításokat igénylő folyamat, a létező rendszerek [4, 5, 6] a legtöbb esetben különböző numerikus módszereket használnak. A 2018-ban publikált munkánkban [38] egy olyan algoritmust fejlesztettünk ki, mely egy tetszőleges háromdimenziós modellhez képes automatikusan meghatározni az említett ketrecmodellt. Mérési eredményeink alapján kijelenthető, hogy az általunk bemutatott módszer szignifikánsan gyorsabb futási időket produkál a létező rendszerekkel szemben.

Manapság már egyre gyakoribb és természetesebb, hogy az ember-gép kommunikációban virtuális avatárokat is használunk. Egy-egy ilyen avatár elkészítésekor elkerülhetetlen lépés a modell felkészítése az animációra. A feladat megoldására már léteznek az MPEG-4 szabványt támogató, deformációalapú rendszerek [7, 8], azonban az általuk használt deformációs technikák hátrányai és korlátai miatt sok esetben nem tudnak elegendően valósághű modelleket generálni. A 2018-ban megjelent publikációnkban [51] egy, az MPEG-4 szabványt támogató félig automatikus algoritmusunkat mutattuk be, mely egy tetszőleges modell animációra felkészítő kalibrációját képes elvégezni. Módszerünk a harmonikus koordinátákat használja a modell deformációjára. Validációs méréseink alapján pedig elmondható, hogy algoritmusunk valósághűbb eredményeket produkál, mint a létező rendszerek.

6 Summary

In this dissertation, I present our results and algorithms, which deal with some actual problems of deformation techniques based on the generalized barycentric coordinates.

In the last years, the theoretical basis of the barycentric coordinates has been well-developed, and these coordinates are frequently used as building blocks of different applications. It is clearly visible that the research of the barycentric coordinate methods is a hot topic in computer graphics.

My research of the barycentric coordinates began with comparing the Wachspress, the discrete harmonic, and the mean value coordinate methods for convex, n-sided polygons using the patterns of the contour lines [20]. Because the contour lines' curvature functions were investigated, our comparison is much more detailed than findings in the previous works [3]. Moreover, to overcome the drawbacks of these methods, the affine combination of two different barycentric coordinate functions was introduced, giving extra flexibility to the user.

The generalized barycentric coordinates are frequently used for image and model deformation. In cage-based image deformation applications, the source polygon's triangulation—which wraps the input image—is necessary. However, the determination of that triangulation is not always trivial because it can influence the deformation. Therefore, we investigated the effects of the different triangulation methods in the aspect of the cage-based image deformation techniques [24]. According to our results, we can say that in those cases, when a GPU implementation of the cage-based image deformation methods is available, and we can use a huge number of triangles, the uniform triangulation method is recommended. However, in those cases when the graphical resources of the used device or platform are limited, or the computation cost is important for us, our introduced non-uniform triangulation method can produce better results.

The 3D model deformation techniques based on the barycentric coordinates usually work with a coarser, simplified mesh—called cage—to manipulate a model. Nowadays, some methods [4, 5, 6] can automatically generate cages, but these approaches usually operate with numeric methods, increasing the computation time. Therefore, we created a simply computable and freely customizable algorithm which can define cages automatically for 3D triangulated meshes [38]. Based on our comparison results, we can say that our algorithm can produce significantly faster computation time than the existing algorithms.

Nowadays, the usage of virtual avatars is very common in human-computer interaction, and their role will be important in the future. Those approaches which create avatars are usually based on two phases. In the first one, they create the 3D model of the avatar, while the model calibration for facial animation is executed in the second phase. For calibrating a model, there are some deformation based approaches [7, 8] which support the MPEG-4 standard, but they cannot produce realistic enough results because of the limitations of their used deformation methods. Therefore, we introduced a semi-automatic method that supports the MPEG-4 standard, and it can calibrate a model for facial animation [51]. Our algorithm uses the harmonic coordinates method for the deformation of the model. Based on our measurements, we can say that our solution produces more realistic deformation results than the existing methods.

Köszönetnyilvánítás

Köszönetet mondok azon személyeknek és intézményeknek, akik és amelyek segítették értekezésem létrejöttét.*

Külön köszönettel tartozom témavezetőmnek, dr. Kunkli Rolandnak, aki az egyetemi tanulmányaim megkezdése óta folyamatosan támogatta és segítette szakmai fejlődésemet és előrehaladásomat. A közös munka során rengeteget tanulhattam tőle, és a szakma iránt tanúsított alázata és precizitása mindig követendő példa lesz számomra.

Hálával tartozom dr. Hoffmann Miklósnak, aki szakmai tanácsaival számos alkalommal segítette kutatásom előrehaladását.

Továbbá szeretnék köszönetet mondani családomnak és páromnak, hogy mindvégig bátorítottak és támogatásukkal hozzájárultak doktori kutatásom megvalósításához és a disszertációm elkészítéséhez.

*A kutatást az „Integrált kutatói utánpótlás-képzési program az informatika és számítástudomány diszciplináris területein” (EFOP-3.6.3-VEKOP-16-2017-00002) című projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

Irodalomjegyzék

[1] A. F. Möbius. *Der barycentrische Calcul.* Verlag von Johann Ambrosius Barth, 1827.

[2] E. L. Wachspress. *A Rational Finite Element Basis.* New York: Academic Press, 1975.

[3] M. S. Floater, K. Hormann és G. Kós. "A General Construction of Barycentric Coordinates over Convex Polygons". *Advances in Computational Mathematics* 24.1–4 (2006), 311–331. old.

[4] Z.-J. Deng, X.-N. Luo és X.-P. Miao. "Automatic Cage Building with Quadric Error Metrics". *Journal of Computer Science and Technology* 26.3 (2011), 538–547. old.

[5] C. Xian, H. Lin és S. Gao. "Automatic cage generation by improved OBBs for mesh deformation". *The Visual Computer* 28.1 (2012), 21–33. old.

[6] L. Sacht, E. Vouga és A. Jacobson. "Nested Cages". *ACM Transactions on Graphics (TOG)* 34.6 (2015), 1–14. old.

[7] M. Escher, I. Pandzic és N. M. Thalmann. "Facial Deformations for MPEG-4". *Proceedings of the Computer Animation.* CA '98. USA: IEEE Computer Society, 1998, 56–62. old.

[8] F. Lavagetto és R. Pockaj. "The Facial Animation Engine: Toward a High-Level Interface for the Design of MPEG-4 Compliant Animated Faces". *IEEE Transactions on Circuits and Systems for Video Technology* 9.2 (1999), 277–289. old.

[9] K. Hormann és N. Sukumar, szerk. *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics.* CRC Press, 2017.

[10] M. Meyer, A. Barr, H. Lee és M. Desbrun. "Generalized Barycentric Coordinates on Irregular Polygons". *Journal of Graphics Tools* 7.1 (2002), 13–22. old.

[11] U. Pinkall és K. Polthier. "Computing Discrete Minimal Surfaces and Their Conjugates". *Experimental Mathematics* 2.1 (1993), 15–36. old.

[12] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery és W. Stuetzle. "Multiresolution Analysis of Arbitrary Meshes". *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1995).* 1995, 173–182. old.

[13] K. Hormann és M. S. Floater. "Mean Value Coordinates for Arbitrary Planar Polygons". *ACM Transactions on Graphics (TOG)* 25.4 (2006), 1424–1441. old.

[14] T. Ju, S. Schaefer, J. Warren és M. Desbrun. "A Geometric Construction of Coordinates for Convex Polyhedra using Polar Duals". *Proceedings of the 3rd Eurographics Symposium on Geometry Processing (SGP 2005).* 2005, 181–186. old.

[15] T. Ju, P. Liepa és J. Warren. "A general geometric construction of coordinates in a convex simplicial polytope". *Computer Aided Geometric Design* 24.3 (2007), 161–178. old.

[16] M. S. Floater, G. Kós és M. Reimers. "Mean value coordinates in 3D". *Computer Aided Geometric Design* 22.7 (2005), 623–631. old.

[17] P. Joshi, M. Meyer, T. DeRose, B. Green és T. Sanocki. "Harmonic Coordinates for Character Articulation". *ACM Transactions on Graphics (TOG)* 26.3 (2007), 71:1–71:9.

[18] T. Ju, S. Schaefer és J. Warren. "Mean Value Coordinates for Closed Triangular Meshes". *ACM Transactions on Graphics (TOG)* 24.3 (2005), 561–566. old.

[19] C. Tomasi és R. Manduchi. "Bilateral Filtering for Gray and Color Images". *Sixth International Conference on Computer Vision*. 1998, 839–846. old.

[20] Á. Tóth. "Comparison and affine combination of generalized barycentric coordinates for convex polygons". *Annales Mathematicae et Informaticae* 47 (2017), 185–200. old.

[21] P. G. Guest. *Numerical Methods of Curve Fitting*. Cambridge University Press, 2012.

[22] R. Courant. *Dirichlet's Principle, Conformal Mapping, and Minimal Surfaces*. Interscience Publishers, 1950.

[23] O. Weber, M. Ben-Chen és C. Gotsman. "Complex Barycentric Coordinates with Applications to Planar Shape Deformation". *Computer Graphics Forum* 28.2 (2009), 587–597. old.

[24] Á. Tóth és R. Kunkli. "The Effects of Different Triangulation Techniques for Cage Based Image Deformation Using Generalized Barycentric Coordinates". *Journal of WSCG* 28.1–2 (2020), 155–162. old.

[25] J. R. Shewchuk. "Delaunay Refinement Algorithms for Triangular Mesh Generation". *Computational Geometry* 22.1–3 (2002), 21–74. old.

[26] Iványi, P. and Radó, J. *Előfeldolgozás párhuzamos számításokhoz*. URL: `https://regi.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0063_07_elofeldolgozas_parhuzamos/ar01s04.html`. Elérés időpontja: 2021. március 2.

[27] L. P. Chew. "Constrained Delaunay Triangulations". *Proceedings of the Third Annual Symposium on Computational Geometry*. 1987, 215–222. old.

[28] S. Suzuki és K. Abe. "Topological Structural Analysis of Digitized Binary Images by Border Following". *Computer Vision, Graphics, and Image Processing* 30.1 (1985), 32–46. old.

[29] L. Pagani és P. J. Scott. “Curvature based sampling of curves and surfaces”. *Computer Aided Geometric Design* 59 (2018), 32–48. old.

[30] J. Kessenich, G. Sellers és D. Shreiner. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V*. Addison-Wesley Professional, 2016.

[31] W. K. Pratt. *Introduction to Digital Image Processing*. CRC press, 2013.

[32] G. Bradski. “The OpenCV Library”. *Dr. Dobb's Journal of Software Tools* (2000).

[33] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 4.14 edition, 2019. URL: `https://doc.cgal.org/4.14/Manual/packages.html`. Elérés időpontja: 2021. március 2.

[34] K. Hormann és N. Sukumar. “Maximum Entropy Coordinates for Arbitrary Polytopes”. *Computer Graphics Forum* 27.5 (2008), 1513–1520. old.

[35] X. Li és S. Hu. “Poisson Coordinates”. *IEEE Transactions on Visualization and Computer Graphics* 19.2 (2013), 344–352. old.

[36] D. Anisimov, D. Panozzo és K. Hormann. “Blended barycentric coordinates”. *Computer Aided Geometric Design* 52 (2017), 205–216. old.

[37] Z. Wang, A. C. Bovik, H. R. Sheikh és E. P. Simoncelli. “Image Quality Assessment: From Error Visibility to Structural Similarity”. *IEEE Transactions on Image Processing* 13.4 (2004), 600–612. old.

[38] Á. Tóth és R. Kunkli. “An automated method for generating customizable cages using barycentric coordinates”. *Annales Mathematicae et Informaticae* 49 (2018), 167–180. old.

[39] M. Garland és P. S. Heckbert. "Surface Simplification Using Quadric Error Metrics". *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques.* USA: ACM Press/Addison-Wesley Publishing Co., 1997, 209–216. old.

[40] M. E. Latoschik, D. Roth, D. Gall, J. Achenbach, T. Waltemate és M. Botsch. "The Effect of Avatar Realism in Immersive Social Virtual Realities". *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology.* 2017, 1–10. old.

[41] H. Pyun, Y. Kim, W. Chae, Hyung W. Kang és S. Y. Shin. "An Example-Based Approach for Facial Expression Cloning". *Proceedings of the 2003 ACM SIGGRAPH/ Eurographics Symposium on Computer Animation.* 2003, 167–176. old.

[42] T. Weise, H. Li, L. Van Gool és M. Pauly. "Face/Off: Live Facial Puppetry". *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* 2009, 7–16. old.

[43] C. Cao, Y. Weng, S. Lin és K. Zhou. "3D Shape Regression for Real-Time Facial Animation". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 1–10. old.

[44] F. I. Parke és K. Waters. *Computer Facial Animation.* CRC press, 2008.

[45] I. S. Pandzic és R. Forchheimer, szerk. *MPEG-4 Facial Animation: The Standard, Implementation and Applications.* New York, NY, USA: John Wiley & Sons, Inc., 2002.

[46] K. Balci. "Xface: MPEG-4 Based Open Source Toolkit for 3D Facial Animation". *Proceedings of the Working Conference on Advanced Visual Interfaces.* 2004, 399–402. old.

[47] P. Cosi, A. Fusaro és G. Tisato. "LUCIA a New Italian Talking-Head Based on a Modified Cohen-Massaro's Labial Coarticulation Model". *INTERSPEECH.* 2003, 2269–2272. old.

[48] F. de Rosis, C. Pelachaud, I. Poggi, V. Carofiglio és B. De Carolis. “From Greta’s mind to her face: modelling the dynamics of affective states in a conversational embodied agent”. *International Journal of Human Computer Studies* 59.1–2 (2003), 81–118. old.

[49] R. Rácz, Á. Tóth, I. Papp és R. Kunkli. “Full-body animations and new faces for a WebGL based MPEG-4 avatar (DEMO)”. *2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. Szerk. P. Baranyi, Á. Csapó és Gy. Sallai. 2015, 419–420. old.

[50] Y. Sheng, P. Willis, G. G. Castro és H. Ugail. “PDE-Based Facial Animation: Making the Complex Simple”. *Advances in Visual Computing*. Springer Berlin Heidelberg, 2008, 723–732. old.

[51] Á. Tóth és R. Kunkli. “An Approximative and Semi-automated Method to Create MPEG-4 Compliant Human Face Models”. *Acta Cybernetica* 23 (2018), 1055–1069. old.

[52] Visage Technologies. *Visage Technologies – Face tracking, Analysis & Recognition*. URL: `http://visagetechnologies.com/mpeg-4-face-and-body-animation/`. Elérés időpontja: 2021. március 2.

[53] P. Fergus, A. El Rhalibi, C. Carter és S. Cooper. “Towards an avatar mentor framework to support physical and psychosocial treatments”. *Health and Technology* 2.1 (2012), 17–31. old.

[54] J. Jia, S. Zhang, F. Meng, Y. Wang és L. Cai. “Emotional Audio-Visual Speech Synthesis Based on PAD”. *IEEE Transactions on Audio, Speech, and Language Processing* 19.3 (2011), 570–582. old.

[55] Blender Online Community. *Blender – a 3D modelling and rendering package*. URL: `http://www.blender.org/`. Elérés időpontja: 2021. március 2.

[56] Á. Tóth és R. Kunkli. *Standardize Me.* URL: `https://arato.inf.unideb.hu/kunkli.roland/software/standardize-me-toolkit/` és `https://github.com/dragostej/standardizeme`. Elérés időpontja: 2021. március 2.

[57] The MakeHuman team. *MakeHuman.* URL: `http://www.makehumancommunity.org/`. Elérés időpontja: 2021. március 2.

[58] Autodesk, Inc. *Autodesk® 123D Catch®.* URL: `https://www.autodesk.com/solutions/123d-apps`. Elérés időpontja: 2021. március 2.

[59] Visual Computing Lab ISTI - CNR. *MeshLab.* URL: `http://meshlab.sourceforge.net/`. Elérés időpontja: 2021. március 2.

[60] L. Yin, X. Wei, Y. Sun, J. Wang és M. J. Rosato. "A 3D Facial Expression Database for Facial Behavior Research". *7th International Conference on Automatic Face and Gesture Recognition (FGR06).* 2006, 211–216. old.

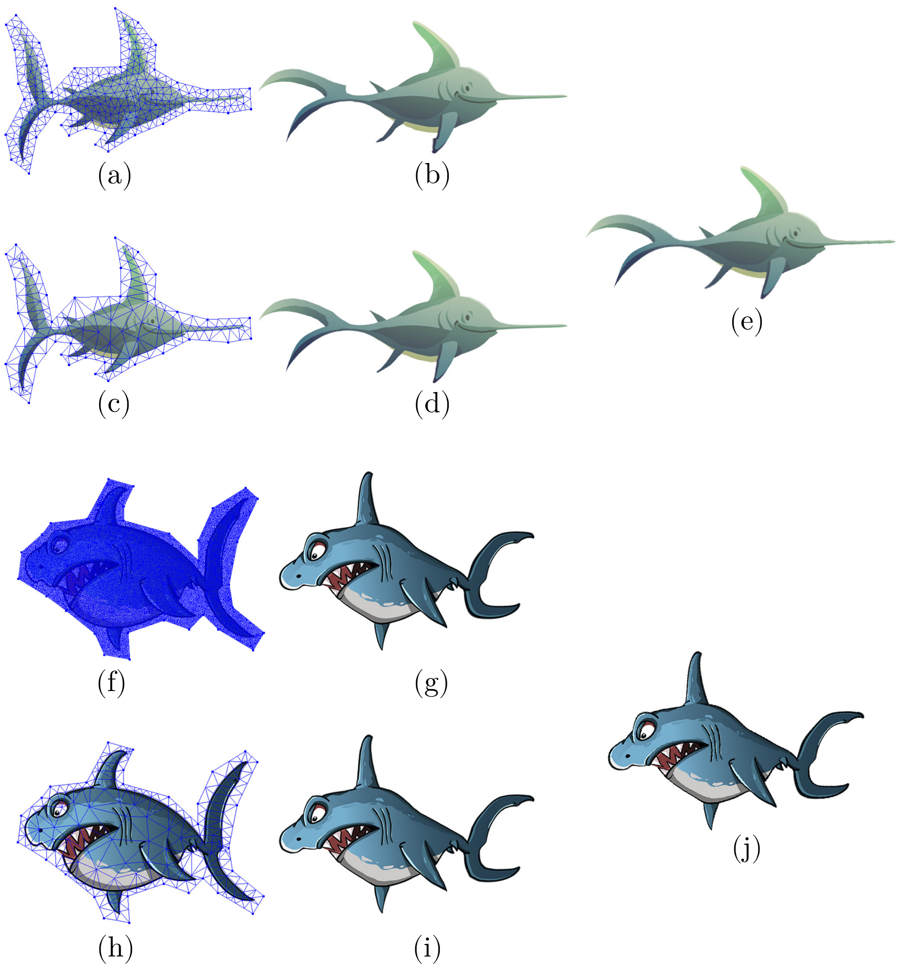
[61] D. Lundqvist, A. Flykt és A. Öhman. *The Karolinska Directed Emotional Faces (KDEF). CD ROM from Department of Clinical Neuroscience. Psychology section, Karolinska Institutet, ISBN 91-630-7164-9.* 1998.

A. Képdeformációhoz használt háromszögelési technikák összehasonlításának kimenetei

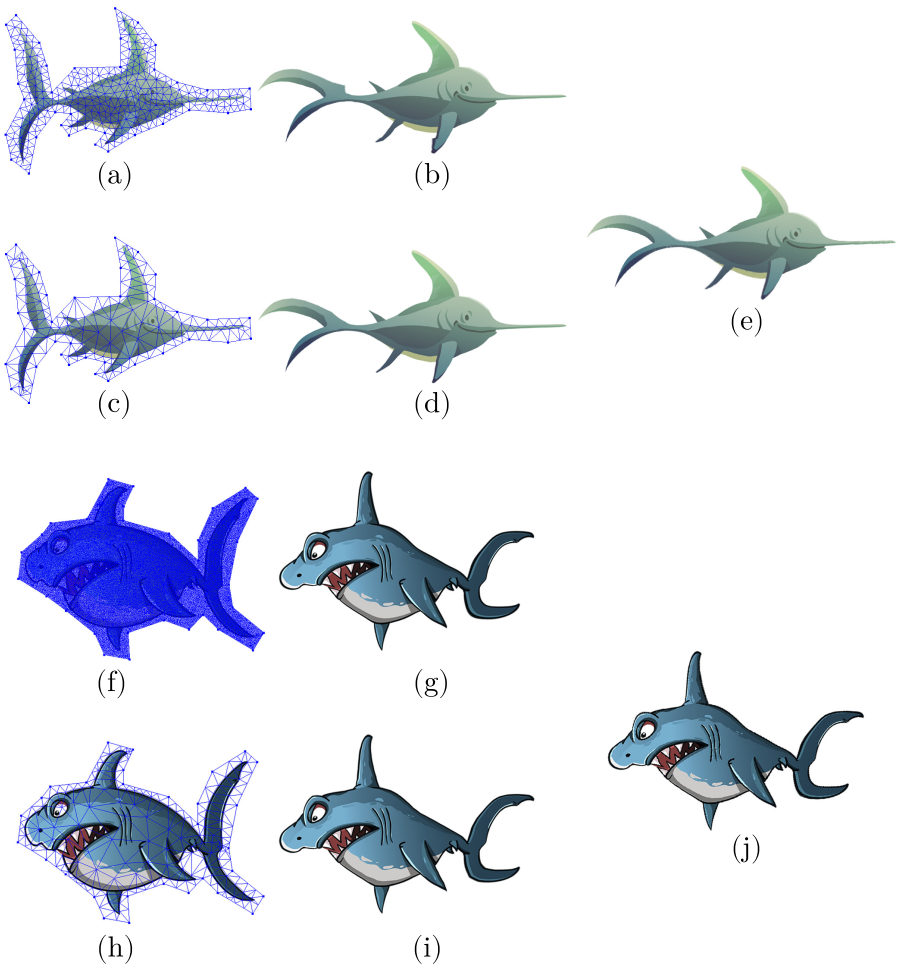
		Cápa		Kardhal	
	Koordináta-módszer	E	NE	E	NE
Háromszögek száma	középérték	33 035	375	617	298
	Poisson	33 035	375	617	298
	maximum entrópia	33 035	387	617	316
SSIM	középérték	0,978	0,973	0,986	0,987
	Poisson	0,978	0,973	0,986	0,987
	maximum entrópia	0,977	0,973	0,987	0,987
Háromszögelés számítási ideje (s)	középérték	2,341	0,001	0,002	0,001
	Poisson	2,341	0,001	0,002	0,001
	maximum entrópia	2,351	0,001	0,002	0,001
Koordináták számítási ideje (s)	középérték	0,026	0,001	0,003	0,001
	Poisson	0,254	0,003	0,003	0,001
	maximum entrópia	0,291	0,003	0,003	0,001

E - egyenletes, NE - nem egyenletes

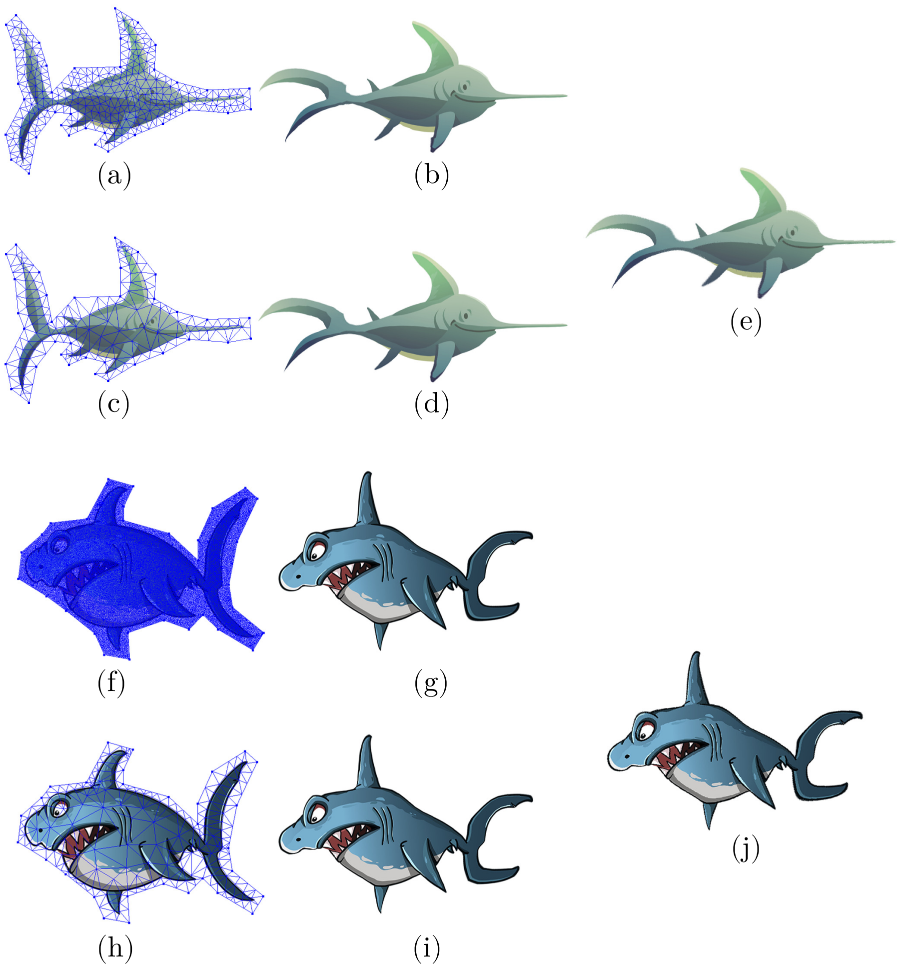
A.1. táblázat. Az egyenletes és a nem egyenletes háromszögelési technika összehasonlítása



42. ábra. Bemeneti képek egyenletes (a, f), illetve nem egyenletes (c, h) háromszögelési technikával felosztott határoló sokszögekkel, illetve deformációs eredményeik (b, g és d, i) a **középérték** koordináták használatával. A referenciaként szolgáló deformációs eredmények (e, j) az utolsó oszlopban láthatóak.

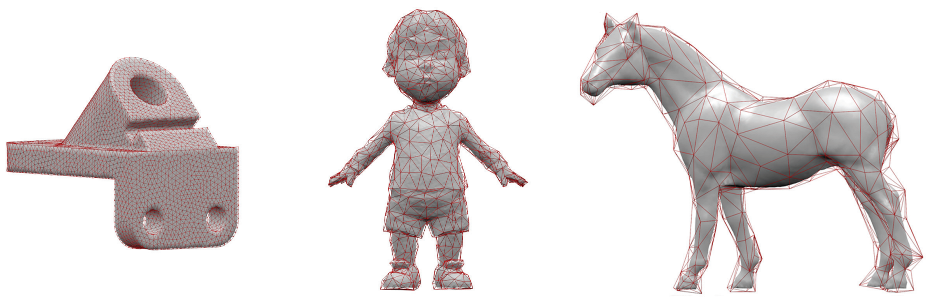


43. ábra. Bemeneti képek egyenletes (a, f), illetve nem egyenletes (c, h) háromszögelési technikával felosztott határoló sokszögekkel, illetve deformációs eredményeik (b, g és d, i) a **Poisson** koordináták használatával. A referenciaként szolgáló deformációs eredmények (e, j) az utolsó oszlopban láthatóak.

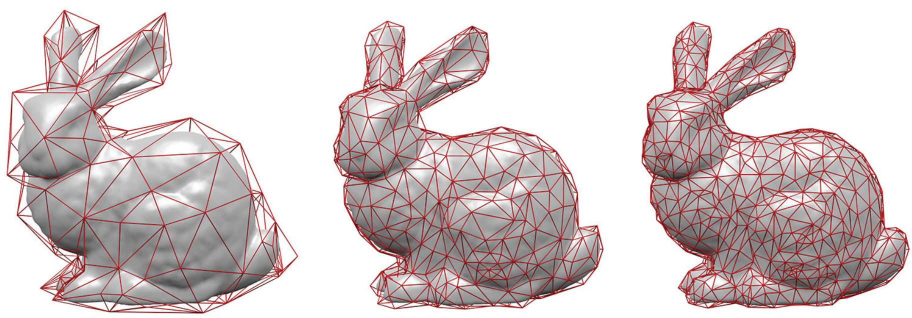


44. ábra. Bemeneti képek egyenletes (a, f), illetve nem egyenletes (c, h) háromszögelési technikával felosztott határoló sokszögekkel, illetve deformációs eredményeik (b, g és d, i) a **maximum entrópia** koordináták használatával. A referenciaként szolgáló deformációs eredmények (e, j) az utolsó oszlopban láthatóak.

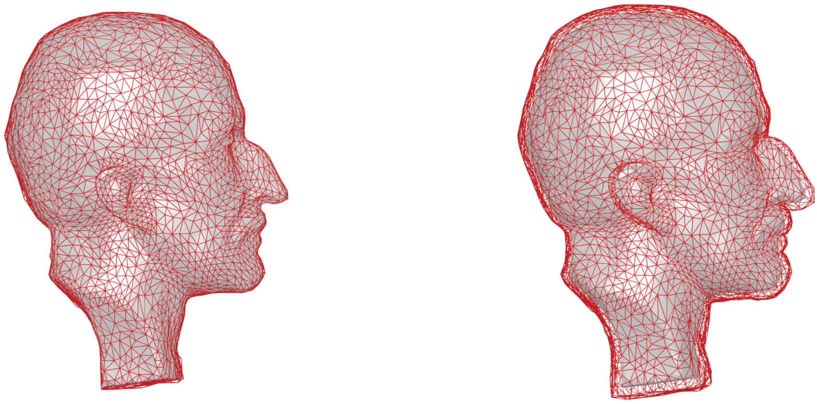
B. Saját ketrecgeneráló algoritmus kimenetei



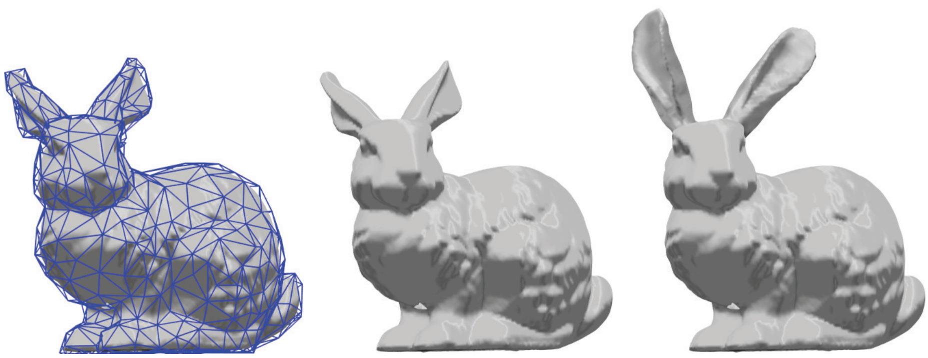
45. ábra. Bemeneti modellek szürkével és a hozzájuk generált ketrecmodellek pirossal, drótvázként megjelenítve



46. ábra. A *Stanford Bunny* mint bemeneti modell és a különböző csúcsszámmal generált ketrecei. A csúcsok száma balról jobbra: 143, 448 és 805.

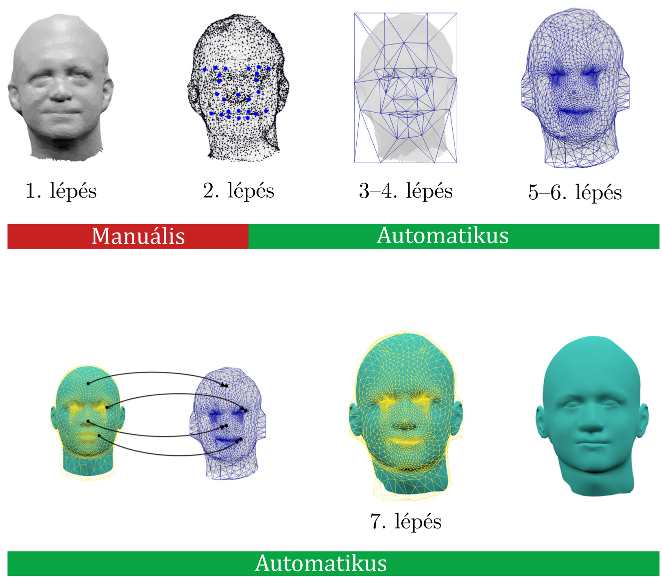


47. ábra. Bemeneti modellek és ketrecek különböző súlyokkal: $w_1 = w_2 = w_3 = w_4 = \frac{1}{4}$ (balra), illetve $w_1 = w_2 = \frac{2}{100}$, $w_3 = \frac{95}{100}$ és $w_4 = \frac{1}{100}$ (jobbra)

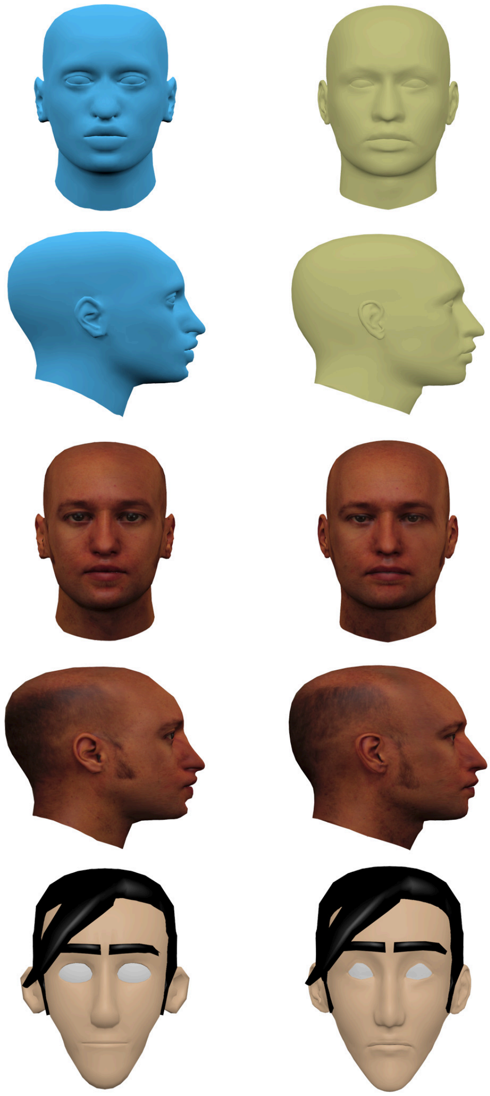


48. ábra. A *Stanford Bunny* modellhez generált ketrec felhasználása a modelldeformációban

C. A fejmodell-kalibrációt támogató módszerünk folyamatábrája és kimenetei



49. ábra. A fejmodellek MPEG-4 kalibrációját megvalósító félig automatikus algoritmusunk. Minden egyes részábra az aktuális lépés eredményét mutatja. Kizárólag az első két lépés manuális, a felhasználónak meg kell határoznia egy bemeneti modellt és a szükséges pontokat jelölnie kell rajta. A további lépések automatikusak; az algoritmus a fejmodellhez tartozó ketrecet a 3–6. lépésben generálja. Ezután az általános modell ketrecpontjait eltolja a hozzájuk tartozó fejmodell ketrecpontjainak a pozíciójába. Mivel a harmonikus koordináták módszerét alkalmazza az algoritmus az általános modellen, így az a bemeneti modellt közelíteni fogja.



50. ábra. Bemeneti modellek (balra) és a kalibrációs folyamat után kapott eredmények (jobbra). A használt textúrák a *Karolinska Directed Emotional Faces (KDEF)* adatbázisból [61] származnak. Az utolsó példa esetén a haj egy külön modell, így az manuálisan lett hozzáadva az eredménymodellhez.

D. Publikációs jegyzék

Referált folyóiratcikkek

[F1] **Á. Tóth** és R. Kunkli. “The Effects of Different Triangulation Techniques for Cage Based Image Deformation Using Generalized Barycentric Coordinates”. *Journal of WSCG* 28.1–2 (2020), 155–162. old.
DOI: 10.24132/jwscg.2020.28.19.

[F2] **Á. Tóth** és R. Kunkli. “An Approximative and Semi-automated Method to Create MPEG-4 Compliant Human Face Models”. *Acta Cybernetica* 23 (2018), 1055–1069. old.
DOI: 10.14232/actacyb.23.4.2018.5.

[F3] **Á. Tóth** és R. Kunkli. “An automated method for generating customizable cages using barycentric coordinates”. *Annales Mathematicae et Informaticae* 49 (2018), 167–180. old.
DOI: 10.33039/ami.2018.11.002.

[F4] **Á. Tóth**. “Comparison and affine combination of generalized barycentric coordinates for convex polygons”. *Annales Mathematicae et Informaticae* 47 (2017), 185–200. old.

Könyvrészletek

[K1] **Á. Tóth** és R. Kunkli. “An approximative and semi-automated method to create MPEG-4 compliant human face models”. *The 11th Conference of PhD Students in Computer Science, Volume of short papers, CS^2*. 2018, 21–24. old.

[K2] **Á. Tóth** és R. Kunkli. “Semi-automatic MPEG-4 standardization of 3D head models”. *VIII. Magyar Számítógépes Grafika és Geometria Konferencia.* Szerk. L. Szirmay-Kalos és G. Renner. Neumann János Számítógép-tudományi Társaság. 2016, 53–60. old.

Demó absztraktok

[D1] J. Ozsváth, B. Erős, **Á. Tóth** és R. Kunkli. “An MPEG-4 based talking head for real time voice chatting on Android platform”. *2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom).* Szerk. P. Baranyi. 2017, 369–370. old.

[D2] R. Rácz, **Á. Tóth**, I. Papp és R. Kunkli. “Full-body animations and new faces for a WebGL based MPEG-4 avatar”. *2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom).* Szerk. P. Baranyi, Á. Csapó és Gy. Sallai. 2015, 419–420. old.

[D3] R. Zs. Buda, G. Boldizsár, **Á. Tóth**, Sz. Szeghalmy, R. Tornai és R. Kunkli. “Extended capabilities for a WebGL based talking head system”. *2014 5th IEEE Conference on Cognitive Infocommunications (CogInfoCom).* Szerk. P. Baranyi. 2014, 459. old.

[D4] B. Katócs, **Á. Tóth**, R. Zs. Buda, G. Boldizsár, T. Török, Sz. Szeghalmy, R. Kunkli és A. Fazekas. “New features for an MPEG-4 talking head”. *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom).* Szerk. P. Baranyi, Esposito A., M. Niitsuma és Bjorn Solvang. 2013, 935–936. old.

További konferenciaelőadások

[E1] **Á. Tóth** és R. Kunkli. “The Effects of Different Triangulation Techniques for Cage Based Image Deformation Using Generalized Barycentric Coordinates”. 28. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2020 (WSCG 2020). Virtuális konferencia. 2020.

[E2] **Á. Tóth** és R. Kunkli. “Non-Uniform Triangulation for Image Deformation Using Barycentric Coordinates”. Conference on Geometry Theory and Applications (CGTA 2019). Innsbruck, Ausztria. 2019.

[E3] **Á. Tóth** és R. Kunkli. “An automated method for creating cages for high definition 3D meshes”. The 10th International Conference on Applied Informatics (ICAI 2017). Eger. 2017.

[E4] **Á. Tóth** és R. Kunkli. “An efficient cage generation algorithm for 3D triangulated meshes”. Conference on Geometry Theory and Applications (CGTA 2017). Pilzen, Csehország. 2017.

Poszterprezentációk

[P1] **Á. Tóth** és R. Kunkli. *A Customizable Application for Image Deformation Based on Barycentric Coordinates*. The 12th Asian Forum on Graphic Science (AFGS 2019). Kunming, Kína. 2019.

[P2] **Á. Tóth** és R. Kunkli. *Automatic cage generation for 3D triangulated meshes with different topology*. 5th Winter School of PhD Students in Informatics and Mathematics. Debrecen. 2018.