

TDK dolgozat

Tóth Ákos

Debrecen
2014



**Debreceni Egyetem
Informatikai Kar
Komputergrafika és Képfeldolgozás Tanszék**

Arcmodell MPEG-4 szabványosításának automatizálása ketrecalapú deformációs módszerrel

Témavezető:

Kunkli Roland
tanársegéd

Készítette:

Tóth Ákos
programtervező informatikus MSc
I. évfolyam

**Debrecen
2014**

Tartalomjegyzék

1. Bevezetés	1
2. Létező rendszerek, felmerülő problémák	2
2.1. Virtual Human Director	2
2.2. Sztereoképekből készített 3D fejmodell	4
2.3. Xface	5
2.4. Konzisztens paraméterezésen alapuló fejmodell rekonstrukció	6
3. Felhasznált módszerek	7
3.1. Baricentrikus koordináták	7
3.1.1. Harmonikus koordináták	9
3.2. A Catmull–Clark-féle subdivision technika	10
4. Önálló eredményeink, az arcmodell félautomata szabványosítása	12
4.1. A referenciamodell	13
4.2. A ketrecalapú deformációs módszer	14
4.3. Az automatizált folyamat ismertetése	15
4.4. Eredmény optimalizálásának elősegítése	16
4.5. Eredményeink összegzése	17
5. Összefoglalás	19
6. Saját munka leírása	20
7. Köszönetnyilvánítás	21
8. Irodalomjegyzék	22

1. Bevezetés

Napjainkra egyre több olyan kommunikációs csatorna jelent meg ember és gép között, amelyek esetében már nemcsak billentyűzet, egér vagy érintőképernyő segítségével kezelünk bizonyos eszközöket, hanem beszéddel, gesztusokkal, ahogy ezt mi emberek is tesszük egymás között. Ahhoz, hogy ez a kommunikáció egyre élethűbb és természetesebb lehessen, sokat segít számunkra az, ha látjuk azt a személyt, aki beszél hozzánk, ezzel együtt a beszédhez tartozó artikulációját is. Ez a vizuális információ pedig különösképpen javítja az érthetőséget például zajos környezetben vagy hallássérültek esetében. A számítógép segítségével generált személyek (avatárok) egyik legfőbb célja, hogy ezt a csatornát biztosítsa az ember-gép kommunikáció során.

Az avatárok kitalált vagy létező személy virtuális hasonmását megjelenítő karakterek, amiknek fontos szerepük lehet a jövőben, ugyanis segíthetnek nekünk az információcserében, egy webáruházból történő vásárlásban vagy akár útba is igazíthatnak minket. A két fél közötti kommunikációt többek között az emberi arc animációja biztosítja, mely már régóta tárgyalt témája az informatikának, egészen Frederick I. Parke [25] 1972-es úttörő munkájáig nyúlik vissza.

Azóta számos törekvés jelent meg a témával kapcsolatosan, de mindegyik ezen a téren végzett kutatás egyik legfőbb célja, hogy a kommunikáció valós idejű legyen és a lehető legkevesebb felhasználói beavatkozást igényelje. Voltak olyan irányzatok, melyek szerint a megfelelő animáció az ajkak szinkronizációjában, a szemek, szemöldök mozgásában rejlik. Mások szerint viszont ebben egyéb tényezők is közrejátszanak, mint például a textúrák és a színek. A nagy érdeklődés hatására a területhez kapcsolódóan olyan szabványok jelentek meg, melyek támpontot próbálnak állítani egy ilyen animáció megvalósításához. Ide sorolható az MPEG-4 szabványcsomag is.

Léteznek már olyan rendszerek, melyekkel virtuális avatárokat tudunk készíteni. Ezekben általában először az arc rekonstrukciója történik meg, ez történhet képek vagy szenzorok, szkennerek felhasználásával. Ezután következik az arc bizonyos részeinek a megjelölése, melyek az animációs technikák számára szolgáltatnak információkat. Majd az animálásra és a megjelenítésre is sor kerül.

Azonban ezen rendszerek legtöbbje vagy nem szabványspecifikus vagy meglehetősen sok manuális interakciót igényel. Ezért a már szakdolgozatomban [4] is tárgyalt és ezen dolgozatban is kifejtett rendszerek tanulmányozása után arra jutottunk, hogy konstruálunk egy olyan módszert, amelyet felhasználva egy virtuális avatár MPEG-4 szabvány szerinti felparaméterezése minél hatékonyabb és automatizáltabb lehet. Távoli célkitűzésünk, hogy ha egy személy helyet foglal egy *Kinect* szenzorral felszerelt számítógép előtt, akkor webes felületen keresztül pár pillanat alatt automatikusan legenerálhassa saját virtuális avatárját.

A dolgozatban részletesen bemutatjuk a témát illetően a különböző csoportba sorolható rendszerek közül a legmeghatározóbbakat. A módszerünk létrehozásához szükséges elméleti háttérrel is ismertetjük az olvasóval. Majd betekintést nyújtunk az elkészített eljárás lépésenkénti működésébe, képekkel illusztrálva azt. Illetve kifejtjük az elért eredményeinket és a jövőbeni terveinkről, az algoritmus javítható pontjairól is szót ejtünk.

2. Létező rendszerek, felmerülő problémák

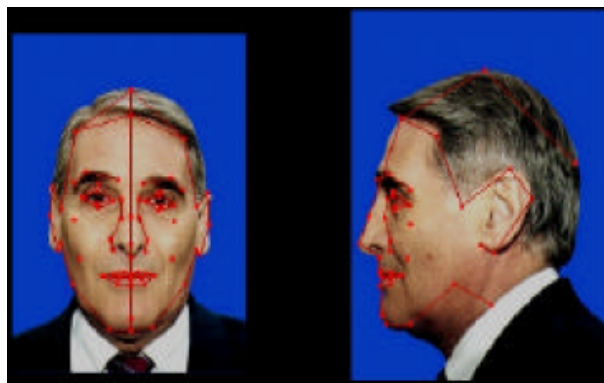
Az emberi test legkifejezőbb része az arc, érzelmeket tudunk leolvasni akár csak egy pillantásból, képesek vagyunk egyetlen apró mimikából eldönteni mások hangulatát. Az arcunk kifejezi érdeklődésünket, vagy akár nemtetszésünket egy dologra vonatkozóan. Ezért a számítógéppel generált arcok egy programban eszközei lehetnek annak, hogy a felhasználók úgy kommunikáljanak vele, mint azt a hétköznapiakban teszik egy illetővel. Viszont ebben az esetben elvárjuk azokat a jellemzőket, amik egy emberi archoz hozzátartoznak: a kimondott szövegnek megfelelően artikuláljon, legyen egyedi, mutasson ki emóciókat. Mégis talán a legfontosabb követelményünk az, hogy az animált arc egy átlagos teljesítményű és felszereltségű számítógépen is valós időben reagáljon.

Az elmúlt két évtizedben számos módszert és technikát alkottak meg abból a célból, hogy segítsék egy emberi arc rekonstrukciójának az elkészítését [30], majd animálását [29, 20]. Ezen módszereket akár csoportokba is sorolhatjuk, a legfontosabb csoportosítási szempont a szabványspecifikusság, így beszélhetünk szabvány alapú vagy szabványok igénybevétele nélkül megvalósított technikákról is. Szintén fontos szempont lehet az is, hogy a rekonstrukció milyen módon készül el, ez történhet fotókkal [6] vagy 3D szkennerek [15] felhasználásával. Nem elhanyagolható az sem, hogy egy ilyen virtuális avatár megkonstruálása mennyi időt és felhasználói interakciót igényel.

A következő alfejezetekben ezek közül az általunk is megvizsgált és meghatározónak vélt módszerek közül mutatunk be néhányat.

2.1. Virtual Human Director

Az egyik első ilyen rendszer, az ún. *Virtual Human Director* (VHD) [19] volt, ami teljes mértékben MPEG-4 szabvány kompatibilis, a modell rekonstrukciója pedig két fotóból (elől- és oldalnézet) született meg. Az animáció alapjául szolgáló tartópontokat (Feature Point – FP) egy felhasználói felület segítségével jelölhetjük meg a két fotón. A modell textúrázása teljesen automatikusan történik a képek egy komponált változatából.



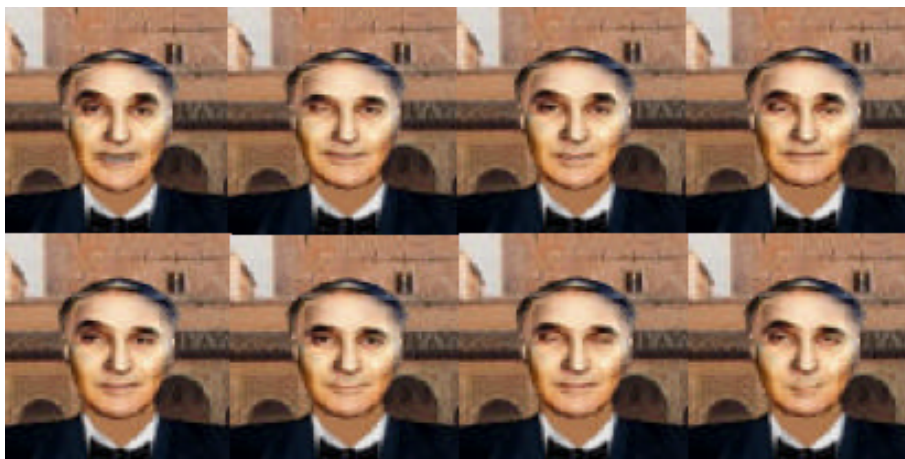
1. ábra. A két fotó, a megjelölt tartópontokkal [19].

Azonban ennek a módszernek is megvannak a hibái. Az MPEG-4 84 darab FP pontot definiál a szabványban – melyek elengedhetetlenek a realisztikus animációhoz –, ezek közül néhány nem látszik a fotón, ezért a bejelölésük nem lehetséges. A legszembetűnőbb példa erre a fogakon és a nyelven található négy-négy pont, de probléma van a fejtetőn elhelyezkedő tartópontokkal is. Nehézség fakad abból is, hogy bizonyos pontok közel helyezkednek el egymáshoz a képeken, ha az animálni kívánt személy összezárt ajkakkal szerepel a fotón, akkor az ajkakat definiáló legfontosabb pontok nehezen meghatározhatóak. Ezek következtében összesen 73 darab FP adható meg a képekről.

Ahhoz, hogy a fotókból egy 3D modellt kapjunk, egy általános modellt kell deformálni úgy, hogy a tartópontokat kontrollpontként tekintjük. Ezt a deformációt pedig a kontrollpontok megfelelő interpolációja fogja jelenteni. A kontrollpontok konvex burka tartalmazza a 3D modell deformálásához szükséges legtöbb pontot, de a megfelelő eredmény eléréséhez szükség van további extra pontokra, szám szerint 27 darabra. A fogak és szemek egy eltolással és egy skálázással fognak a helyükre kerülni a deformálást követően.

A textúra automatikus elkészítésében is fontos szerepe lesz majd az előzőekben bejelölt pontoknak. Azonban mielőtt ezen pontok alapján illesztenénk a modellre a textúrát, szükséges annak megalkotása is. Erre a feladatra a bemenetként adott képek szintén meg fognak felelni. Először a képek összeillesztését végzi el a rendszer, az előlnézeti fotó jobb oldalához illeszti az oldalnézetből készített képet, a bal oldalához pedig ugyanennek a tükrözött változatát. A még élethűbb hatás keltéséhez az illesztett fotók határvonalait elmossa a szoftver, az ún. *Gaussian pyramid decomposition* [8] technikával. Ezek után már csak a textúra illesztése marad hátra.

Miután elkészült az MPEG-4 alapú modellünk, megszólaltathatjuk azt. A VHD felhasználói felületével lehetőségünk van erre is. Mozgathatjuk a kreált virtuális arcot, testet és a kamerát is, előre definiált arc- és testanimációkat, emóciókat játszhatunk le, de akár egy begépett szöveg elmondására is képes az animált arc, melyet akár rögzíthetünk is.



2. ábra. Az elkészült virtuális modell, különböző arckifejezésekkel [19].

A módszer ereje a hatékony rekonstrukcióban rejlik, azonban mára ez már nem feltétlen igaz, az olyan megjelent eszközök mellett, mint a 3D szkennerek vagy a Microsoft Kinect szenzorok, amelyek várhatóan egyre több háztartásban megtalálhatóak lesznek.

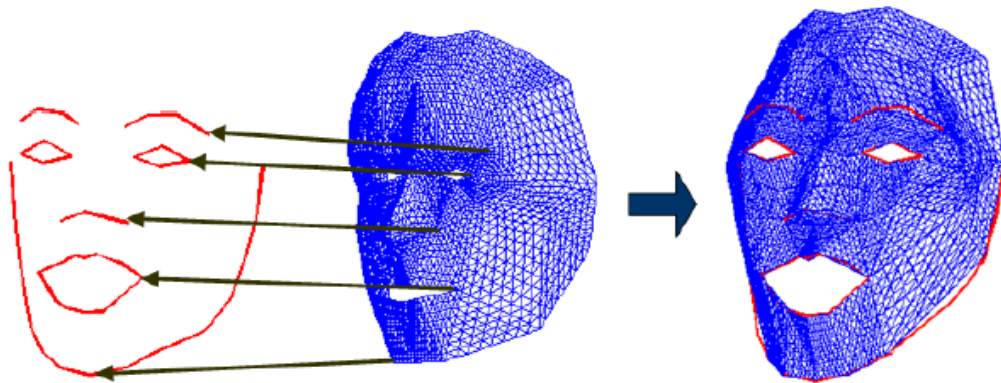
2.2. Sztereoképekből készített 3D fejmodell

Az alábbi módszer [27] egy 3D fejmodell elkészítését két sztereoképpel és egy önszerveződő neurális hálóval valósítja meg. Segítségével különböző jellemvonású és kifejezésű emberi arcokat tudunk gyorsan és automatikusan generálni.

A módszer kis számú tartópontokkal dolgozik – csakúgy, mint a 2.1.-es alfejezetben elemzett módszer –, melyek felhasználásával rekonstruálni tudjuk a fejmodellt. Ezen pontok képeken való megjelölése automatikusan történik, ez a lépés egy mintaillesztési problémára vezethető vissza. A probléma megoldásához a *Cootes* által definiált *Active Appearance Model* (AAM) [11] egy változatát, a független AAM-et [22] használták a szerzők, azért mert alacsony számítási idővel dolgozik és valós idejű alkalmazásokban is használható.

Miután megtörtént a sztereoképeken a tartópontok kiválasztása, a módszer kiszámolja a hozzájuk tartozó térbeli pontokat, amelyek a 3D fejmodellt meghatározzák. Ehhez először kamerakalibráció szükséges, majd a kamera fókuszaiban és a tartópontokban felállított egyenesek metszéspontjai fogják adni a térbeli pontokat.

Mivel kisszámú tartóponttal dolgozik a módszer, így egy nagyon egyszerű 3D modellt kapunk, ezért egy általános modellt próbál meg illeszteni ezen tartópontokra. Ennek megvalósításában működik közre a *Kohonen* [14] neurális hálózat. Ennek a hálónak a nagy előnye, hogy képes tetszőleges számú rendezetlen pontot viszonylag rövid időn belül rendezni. A tanulási folyamat végére pedig egy topológiát megtartó ponthálót kapunk, melyek a tartópontok térbeli pozícióit követik.



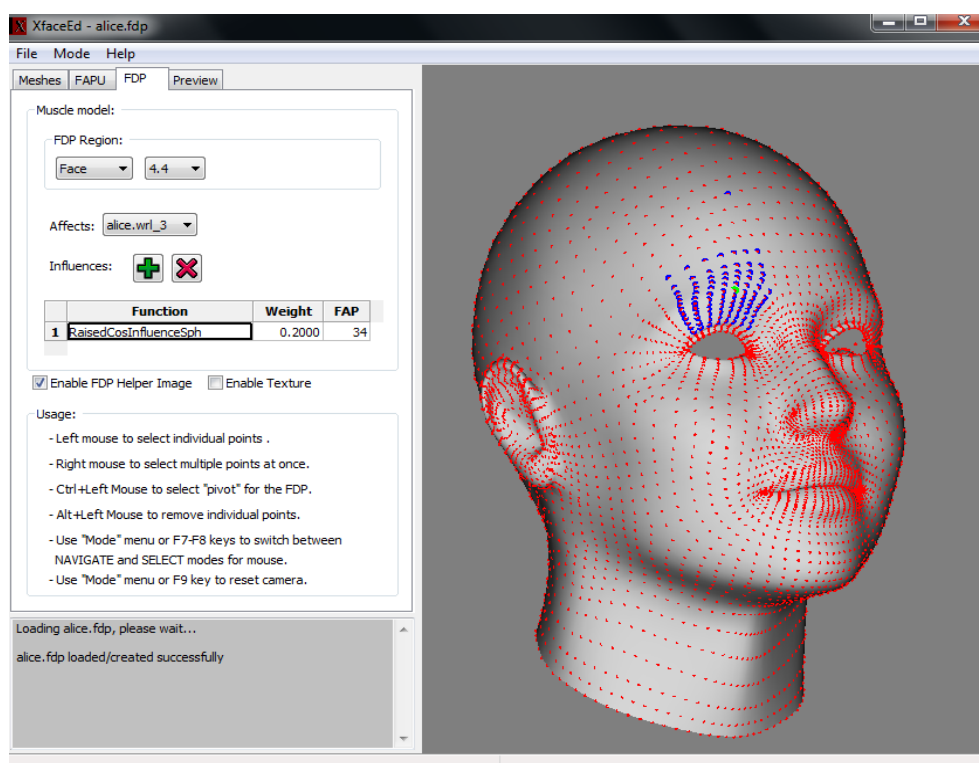
3. ábra. Az általános modell illesztése a rekonstruált tartópontokra [27].

A módszer hatékonyságát az automatikus volta adja, azaz a két sztereokép elkészítésén kívül nincs más dolgunk. A képeken a szükséges tartópontokat egy arcfelismerési technikával önállóan választja ki a rendszer, melyekből ezután térbeli pontokat állít elő. Ezek után egy neurális hálóval egy általános modellt illeszt a pontokra.

Viszont az alacsony számú tartópontok miatt a kapott modell a rekonstruálni kívánt személyhez való hasonlósága, egy 3D-s eszközzel készített modellével nem versenyezhet.

2.3. Xface

Az Xface [5] egy olyan nyílt forráskódú, platformfüggetlen eszköztár, melynek segítségével 3D virtuális asszisztenseket tudunk készíteni az MPEG-4 szabvány szerint. A projekt 2003-ban indult, majd 2004-ben megjelent az eszköztár első verziója. Az évek folyamán pedig folyamatosan karbantartották és új funkciókkal bővítették. A széles felhasználói kör miatt a fejlesztők célja volt, hogy az Xface könnyen konfigurálható és bővíthető legyen. Az eszköztárat *C++* nyelven implementálták, míg az animációk az *OpenGL* API-n alapszanak.



4. ábra. Az XfaceEd alkalmazás felhasználói felülete.

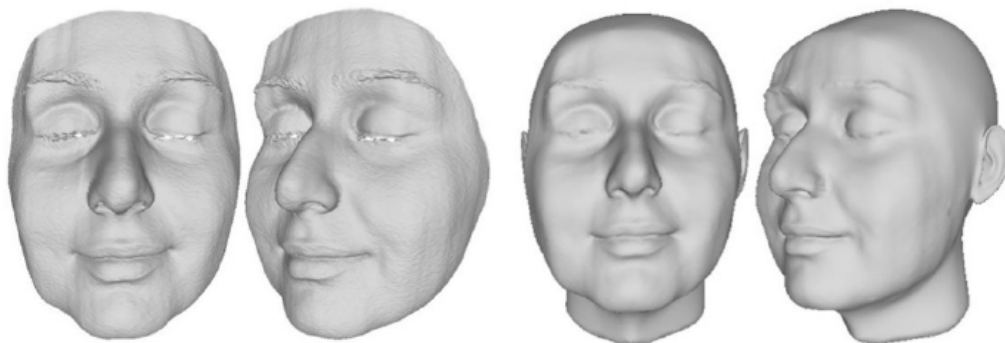
Az Xface négy szoftvert foglal magában. A fő könyvtárat, azon fejlesztők számára, akik szeretnének virtuális beszélő modelleket elhelyezni saját alkalmazásaikban, az XfaceEd nevű szerkesztőt, melyben rekonstruált modelljeinket alakíthatjuk MPEG-4 kompatibilissé, az XfacePlayer lejátszót, ahol kipróbálhatjuk az elkészített modelleket és az XfaceClient alkalmazást, amit szkriptszerkesztőként vagy akár kommunikációs eszközként is használhatunk a weben.

A rendszer előnyei között mindenképpen fontos megemlítenünk azt, hogy az MPEG-4 szabvány szerint működik, pozitívum továbbá, hogy többféle 3D bemeneti fájltypussal használhatjuk az alkalmazást. Az egészen kiváló felhasználói felület és a modellek megjelenítése megkönnyíti a felparaméterezés folyamatát, viszont az, hogy az összes paraméter kiválasztása manuálisan történik, rettentő időigényes és ez nagyon sok kellemetlen percet okozhat nekünk a szabványpontok helyes megjelölésében.

2.4. Konzisztens paraméterezésen alapuló fejmodell rekonstrukció

Egy nemrégiben megjelent PhD disszertációban [3] olvashatunk egy olyan rendszerről, amely a konzisztens paraméterezésen alapul. Az újszerűsége és komplexitása miatt tartjuk lényegesnek a rövid bemutatását.

A robosztus és hatékony rendszer igénybevételével különböző egyedi kifejezésű arcmodellek rekonstrukcióját végezhetjük el. Mivel a rendszer fő mozgatórugója a modell manipulációja, így egy előre definiált általános fejmodellt működtet, mely az arcanimáció struktúráját írja le. Kétféle bemenettel dolgozhatunk, megadhatunk pontfelhőt, ekkor a rendszer egy felületrekonstrukciót végez, vagy a már tesszellált felületet is használhatjuk. Az általános modell manipulációja a rekonstruálni kívánt modellen történő kontrollpontok megadásával kerül végrehajtásra, mely manuálisan zajlik, felhasználói interakció igénybevételével. A bemeneti és az általános modell közötti hozzávetőleges megfeleltetésért a módosított RBF alapú deformáció, míg a szinte tökéletes megfeleltetésért egy hengervetület alapú technika felel. Ez a két lépéses adaptáció és a felületen alkalmazott subdivision technika garantálja az általános és a bemeneti modell közötti hasonlóságot a szerző szerint.



5. ábra. Bal oldalon a bemeneti pontfelhő, míg jobb oldalon az ezek alapján konstruált fejmodell [3].

A rendszer kitűnő szolgálatot tesz nekünk egy-egy rekonstrukciókor, tekintve azt, hogy fél automata, illetve az egyes folyamatok végrehajtása rendkívül gyorsan lejátsszódik. Ugyanakkor nem szabványalapú, és a hengervetület használata miatt meglehetősen hibátlan rekonstrukcióra van szükségünk, hogy az eredmény kifogástalan legyen. Az ilyen szintű rekonstrukciót végző eszközök pedig meglehetősen sokba kerülnek. A szerzők például egy *Cyberware* lézerszkennerral dolgoztak, melynek piaci értéke meghaladhatja akár a 15 millió forintot is.

A létező rendszerek és módszerek alapos tanulmányozása után megállapítottuk, hogy olyan rendszer, mely MPEG-4 szabványspecifikus és kevés felhasználó interakciót igényel, egyelőre nem létezik. Ezért arra jutottunk, hogy szeretnénk elkészíteni egy olyan rendszert, mely az arcmodell MPEG-4 szabvány szerinti felparaméterezése a lehető legkevesebb felhasználói interakciót igényelje.

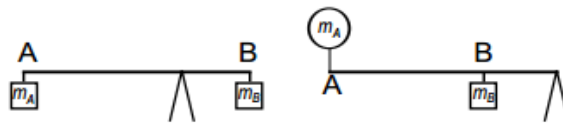
3. Felhasznált módszerek

A következő alfejezetekben szeretnénk röviden összefoglalni azokat a módszereket és eszközöket, melyeket a célunk eléréséhez és megvalósításához igénybe vettünk és felhasználtunk.

3.1. Baricentrikus koordináták

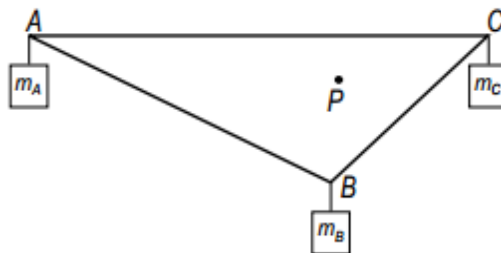
A baricentrikus koordinátákról [18] először *August Ferdinand Möbius* írt az 1827-ben publikált „*The barycentric calculus*” címet viselő könyvében. Egy súlytalan pálcára elhelyezünk két különböző pontban súlyokat, ezután szeretnénk megtudni, hogy hol helyezkedik el a pálca tömegközéppontja vagy más néven baricentruma. Másképpen keressük azt a pontot a pálcán, ahova támasztékot helyezve a pálca egyensúlyban lesz. Ezen kérdés megválaszolásában nyújtanak segítséget a baricentrikus koordináták.

Möbius bevezette a negatív súlyokat is, melyek felfelé irányuló erővel hatnak a pálcára. Így egy negatív és egy pozitív súlyt elhelyezve a pálcán a gravitációs középpont biztosan nem a két súly között fog elhelyezkedni. A baricentrum meghatározásához koordinátákat vezetett be, amiket a pálcán elhelyezett súlyok aránya határoz meg, ahogy már fent említettük. Így a 6. ábrán lévő baricentrumot egy $(m_a : m_b)$ koordinátapárral jelölhetjük.



6. ábra. Bal oldalon két pozitív, míg jobb oldalon egy pozitív és egy negatív súllyal csatolt pálca [18].

Möbius később ezt az ötletet kiterjesztette három súlyt tartalmazó rendszerekre, ami formálisan egy háromszöget jelent úgy, hogy a három súlyt csúcsokként tekintjük. Így a háromszög baricentrumát jelöljük P -vel, amit az $(m_a : m_b : m_c)$ arány fog meghatározni (lásd 7. ábra).



7. ábra. A háromszögre kiterjesztett baricentrikus rendszer [18].

Ha a 7. ábrán lévő súlyokat egységesen csökkentjük vagy növeljük, akkor a súlyok aránya változatlan marad, azaz az általuk meghatározott baricentrum sem fog változni. Így ugyanaz a P pont több súlyhármassal is lehet baricentrum.

Az $(u : v : w)$ koordinátákat gyakran normalizáljuk, hogy az összegük pontosan 1 legyen, azaz $u + v + w = 1$, így könnyítve a velük való számolást [2]. A $P(u : v : w)$ koordinátahármas normalizálásának egyenlete a következőképpen alakul:

$$\frac{1}{u + v + w}(u : v : w) = \left(\frac{u}{u + v + w}, \frac{v}{u + v + w}, \frac{w}{u + v + w} \right).$$

A baricentrikus koordinátákat tovább általánosíthatjuk úgy, hogy 3 helyett n darab pont baricentrumát keressük. Ha a $v_1, \dots, v_n$ pontok, súlyai rendre $w_1, \dots, w_n$, akkor az általuk meghatározott baricentrumot v -vel jelöljük, és a következő módon kapjuk meg:

$$v = \frac{w_1(v)v_1 + \dots + w_n(v)v_n}{w_1(v) + \dots + w_n(v)}.$$

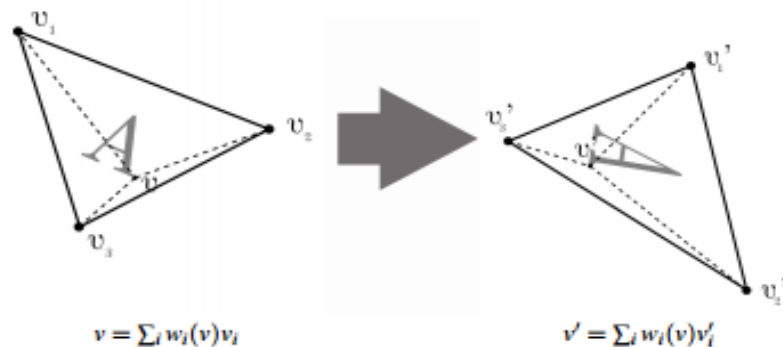
A normalizálás egyenlete:

$$\lambda_i(v) = \frac{w_i(v)}{\sum_i w_i(v)}, \quad \sum_{i=0}^n \lambda_i(v) = 1.$$

Azt felhasználva, hogy a baricentrum lineárisan változhat bizonyos határok között a háromszögön kívül vagy belül, egy ϕ interpolációs függvényt határozhatunk meg a pontokhoz definiált $\phi(v_i)$ értékeket alkalmazva az alábbi módon, a poligon belső pontjainak vonatkozásában:

$$\phi(v) = \sum_{i=0}^n \lambda_i(v) \phi(v_i).$$

Ez az interpolációs technika széleskörben használatos a komputergrafikában, például a renderelési algoritmusoknál, vagy a *Phong* és a *Gouraud* árnyalásnál is. Ugyanez a módszer adja a geometriai deformálások alapját is úgy, hogy ha egy háromszög egy pontját elmozgatjuk, de a súlyok által meghatározott csúcsok és baricentrum közötti kapcsolatot megtartjuk, akkor a belső pontok is új helyre kerülnek (lásd 8. ábra).



8. ábra. A háromszög csúcspozícióinak a megváltoztatására a belső pontok elhelyezkedése is megváltozik, ha a súlyokat – melyek a v_1, v_2, v_3 pontok és a baricentrum közötti kapcsolatot írják le – nem módosítjuk [24].

3.1.1. Harmonikus koordináták

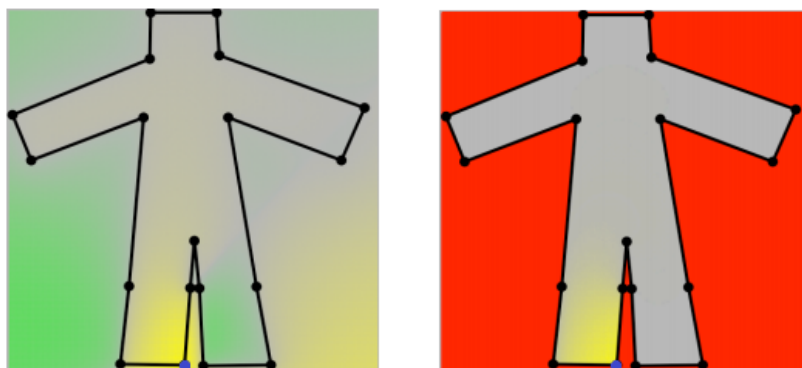
Karakterek mozgatásának a meghatározásához használt deformációs technika, mely fontos szerepet tölt be az animációs filmek készítésekor. A módszert 2007-ben mutatta be [16] a *Pixar Animation Studios*, ami az akkor már létező deformációs technikák továbbfejlesztett változataként jelent meg, de azóta is jó néhány módszert publikáltak [13, 26].

A deformációt egy topológiailag flexibilis, rugalmas struktúra, más néven egy ketrec (kontrollháló) szabályozza, melyen belül az a modell helyezkedik el, amelyet manipulálni szeretnénk. Ez a ketrec pontokból, élekből és háromszöglapokból épül fel, melyek a még precízebb változásokért felelnek.

Az első olyan módszer, ami a különböző modellek manipulációjához ketrec alapú deformációt használt, a 2005-ben megjelent *Mean Value Coordinates* [17] volt. Az *MVC*-nek számos alkalmazási területe van, azonban, ha magas szintű karakter deformációra szeretnénk használni, akkor nemkívánatos viselkedések jelennek meg, mégpedig a következők:

- Belső elhelyezkedés: A koordinátáknak a modell- és a ketrecpontok között mért távolságok függvényében kellene változni, ahol a távolság csak a ketrecen belüli távolságokat jelenti.
- Nem-negativitás: Ha egy modellpont koordinátája negatív a ketrecpontjához képest, akkor a modell- és a ketrecpont ellentétesen mozog.

Ezen hibák kiküszöböléseként jelentek meg a harmonikus koordináták, ami az általánosított baricentrikus koordináták kiterjesztése tetszőleges számú dimenzióra.

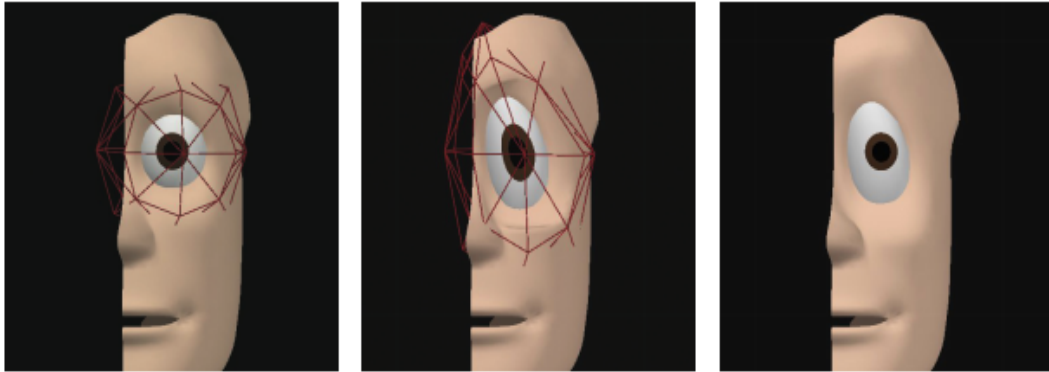


9. ábra. A késsel jelölt pont esetén a koordináta értékek alakulása jobb oldalon az *MVC*, míg bal oldalon a harmonikus koordináták szerint. Sárgával a pozitív, zölddel a negatív, pirossal a nem befolyásolt értékű pontokat jelöltük [16].

Ahogy a 9. ábrán is látszik, harmonikus esetben a koordináták, melyek értéke 0 és 1 között változhat, csak és kizárólag pozitívak, míg az *MVC* alkalmazásakor negatívak is lehetnek. Minél közelebb van egy belső pont a ketrec egy pontjához, az értéke annál inkább 1-hez közelít, ellenkező esetben 0-hoz. Az is megfigyelhető, hogy a harmonikus koordináták módszere csak a ketrec belső pontjaira van hatással, a külső pontok nem manipulálhatóak. Ezért a sima, egyenletes eredmény érdekében a ketrecnek a deformálni kívánt modell minden pontját tartalmaznia kell.

Érdekes tulajdonsága a módszernek az, hogy belső pontokat is megjelölhetünk ketrecpontként, mely ugyanolyan hatással lesz a belső pontokra, mintha ketrecpont lenne. Ezzel

egyidejűleg sajnos csorbát szenved a modell simasága a kontrollpontként használt belső pont környezetében.



10. ábra. A karakter szemgolyóján végzett deformáció, a körülötte felállított ketrec segítségével [16].

3.2. A Catmull–Clark-féle subdivision technika

Talán a leghatásosabb és legegyszerűbb módja annak, hogy a rekonstruált felületeinket még egyenletesebbé és simábbá tegyük, a különböző subdivision technikák használata. Ez egy olyan rekurzív folyamat, mely a durva felület mindegy egyes háromszöglapját további háromszöglapokra osztja. Többfajta subdivision szabályrendszer is létezik, közülük a legmeghatározóbbak: *Loop*, *Butterfly*, *Doo-Sabin*, *Midpoint* vagy a *Catmull-Clark*, melyet részletesebben is áttekintünk.

Amikor egy poliéderen alkalmazunk egy *Catmull-Clark* subdivisiont [12], akkor a következőképpen módosul a test topológiája.

Ha a felületet alkotó lapokat F -el és csúcsait $v_1, v_2, \dots, v_n$ jelöljük, akkor az új lap egyik pontja v_F lesz és az alábbi módon számoljuk:

$$v_F = \sum_{i=1}^n \frac{1}{n} v_i.$$

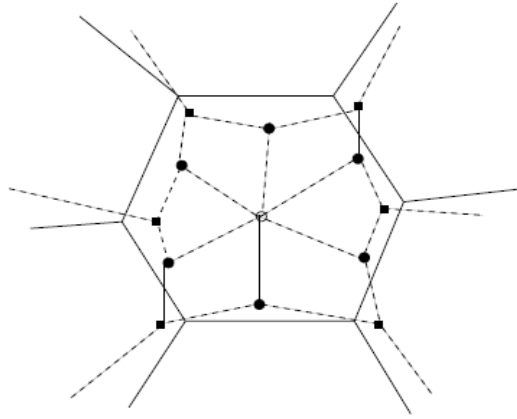
Ha egy F_1 és F_2 lapokhoz tartozó élt E -vel és végpontjait v -vel és w -vel jelöljük, akkor az új élpont v_E lesz, melyet a v, w, v_{F_1} és v_{F_2} pontok átlagaként kapunk meg:

$$v_E = \frac{v + w + v_{F_1} + v_{F_2}}{4}.$$

Az új pontok számítása komplexebb. Legyen adott egy v pont, Q az átlaga azon lapok pontjainak, melyek v -t tartalmazzák és R azon n élek középpontjának átlaga, melyek tartalmazzák v -t, így az új pontot v' jelöljük és a következő módon határozzuk meg:

$$v' = \frac{1}{n} Q + \frac{2}{n} R + \frac{n-3}{n} v.$$

Ezeket az új pontokat összekötve kapjuk az új felületlapokat a következő módon: minden egyes új lap v_F pontját összekötjük az új élpontokkal, melyek az F -et határoló élek lesznek, illetve minden egyes új v' pontot összekötünk az új élpontokkal, mely élek végpontja a v lesz.



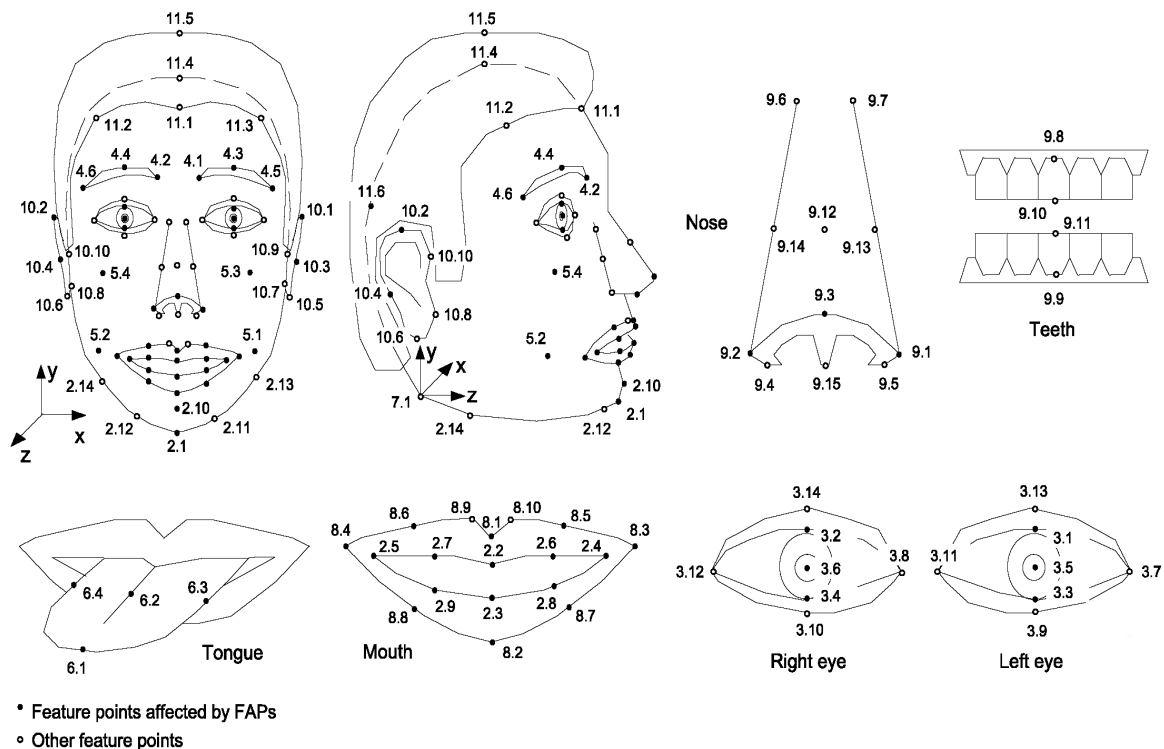
11. ábra. Végrehajtott *Catmull-Clark* subdivision után az üres kör jelöli az új lapok v_F pontját, a kitöltött körök jelölik az új v_E élpontokat, míg a fekete négyszögek az új v' pontokat.¹

¹http://hi.csdn.net/attachment/201010/13/0_1286960368F4Hr.gif, Elérés időpontja: 2014. november 4.

4. Önálló eredményeink, az arcmodell félautomata szabványosítása

Ahogy a bevezetésben is említettük a kutatásunk legfőbb célja nem volt más, mint egy olyan módszer megkonstruálása, melynek felhasználásával egy virtuális avatar MPEG-4 szabvány szerinti felparaméterezését hatékonyabbá tehetjük a lehető legkevesebb felhasználói interakció igénybevétele mellett. Távoli célkitűzésünk az, hogy ha egy felhasználó helyet foglal egy *Kinect* szenzor előtt, akkor pillanatok alatt el tudjuk készíteni az ő virtuális avatarját. Céljaink elérésében fontos állomás az, hogy a szabványosítás folyamatát amennyire csak lehet, megpróbáljuk automatizálni.

Szakdolgozatomban [4] sikerült megalkotnunk egy olyan rendszert, amiben egy virtuális avatar elkészítéséhez a rekonstruált modellt szabványosíthatjuk, azonban azt tapasztaltuk, hogy ezen eljárás rengeteg időt vesz igénybe. Az MPEG-4 szabvány [23] pontosan 84 darab tartópontot (Feature Points - FP) határoz meg, melyekhez tetszőleges számú hatásköri pont is tartozik. Mindezen pontok száma egy kellően részletes rekonstruált modellen, akár több százas nagyságrendűvé is válhat. Így ezek manuálisan történő megadása és pontosítása a minél szebb animáció elérése érdekében egy több órás feladattá is fajulhat.



12. ábra. Az MPEG-4 szabvány által meghatározott 84 tartópont a fejmodellen [21].

Az automatizálás ötlete

Az elképzelésünk az volt, hogy definiálunk egy referenciamodellt, amin előre megadunk minden egyes tartópontot és a hozzájuk tartozó hatásköri pontokat. Elkészítjük arról a személyről – akinek a virtuális avatárját szeretnénk létrehozni – a rekonstrukciót akár fotó alapú technikával, akár *Kinect* szenzor igénybevételével. Majd a referenciaként szolgáló modellünket megpróbáljuk úgy alakítani, módosítani, hogy az minél jobban közelítsen a rekonstruált személy 3D modelljéhez. Ezzel elérve azt, hogy a szabványpontok megjelölésének procedúrája tulajdonképpen egy az egyben elhagyható.

4.1. A referenciamodell

Az általunk használt referenciamodell, a 2.3. fejezetben említett Xface [9] nyílt forrású alkalmazás egy példa modelljének a módosított változata, mely az *Alice* nevet viseli. A *MakeHuman* [28] elnevezésű ingyenes és szintén nyílt forrású szoftver alkalmazásával hozták létre, és mivel már az MPEG-4 szabvány szerint felparaméterezett modell, így számunkra tökéletes választásnak bizonyult. Az eredeti modell hátsó részét, haját, fogsorát és nyelvét eltávolítottuk a későbbi módosítások, alakítások megkönnyítése érdekében, így közelítőleg egy 5000 pontot és 9000 poligonlapot tartalmazó modellt kaptunk.



13. ábra. Az általunk használt, Xface szoftverből importált referenciamodell.

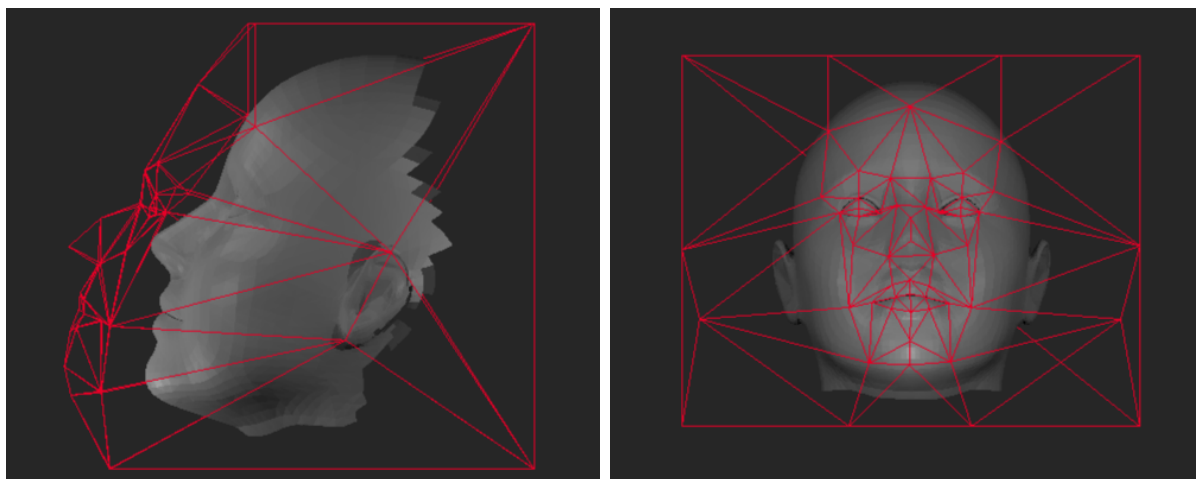
Ahogy a 13. ábrán is látható, egy ilyen számítógéppel generált modell minősége szinte kifogástalan. Ma a felhasználók otthonaiban használatos rekonstrukciós eszközök semelyikével sem vagyunk képesek hasonló eredményre. Hiába veszünk igénybe egy *Kinect* szenzort vagy alkalmazunk fotó alapú rekonstrukciót. Így ha a referenciamodellen alkalmazott alakítások hatására az eredmény elegendően hasonlít a rekonstruálni kívánt személyre, akkor egy kellően élethű virtuális avatárt tudunk létrehozni. A továbbiakban ezen alakítások és módosítások bemutatására kerül sor.

4.2. A ketrecalapú deformációs módszer

A bemutatott referenciamodellel végzett alakításokat, a 3.1.1. alfejezetben részletesen tárgyalt harmonikus koordinátáknak a módszerével végeztük. Ahogyan ott is tárgyaltuk, a modell deformációja egy körülötte felállított kontrollháló vagy más néven ketrec segítségével történik. Ha a ketrec pontjait mozgatjuk, akkor az ezen pontok hatáskörébe eső felületpontok pozíciói az elmozdulás mértékének a függvényében változni fognak.

Ezt a tulajdonságot kihasználva a deformálandó referenciamodell körül definiáltunk egy ketrecet, ami ezen modell megfelelő manipulálásáért lesz felelős. Így a ketrec alkalmazásán választott deformációjával elérhető az, hogy a referenciamodellünk a lehető legjobban közelítsen a rekonstruálni kívánt személy modelljéhez. Ennek a deformációnak a kiszámításához szükségünk lesz a rekonstruálni kívánt arcmodell körül felállított ketrecre is. Ezután a két ketrec pontjaiból és struktúrájából származtatni tudjuk azon alakításokat, melyek a végeredményt fogják szolgáltatni.

Sikerült elkészítenünk egy olyan alkalmazást, melynek felhasználásával egy tetszőleges fejmodellhez a deformációhoz szükséges kontrollháló automatikusan generálható. Az alkalmazásnak bemenetként szüksége van azon modellre, melyhez a ketrecet készítjük, és egy információs fájlra, melyben bizonyos egyedi arcvonásokat leíró pontpozíciókat tárolunk. Ezeket a pontokat az MPEG-4 szabványból is ismert tartópontokból határoztuk meg, melyek a helyes ketrecnek az elkészítését garantálják számunkra. Szám szerint 48 darab ilyen szabványpont megjelölésére van szükség. Ezen pontok megjelölésében segítséget nyújthat a szakdolgozatomhoz mellékelt *Standardize Me!* [1] nevet viselő alkalmazás. Ugyan nem kimondottan erre a feladatra lett specializálva a szoftver, de teljes mértékben használható a probléma megoldásához. Terveink között szerepel a módosítása és útmutatási funkciókkal való kiegészítése a könnyebb használat érdekében.



14. ábra. A referenciamodellhez generált ketrec oldal- és előlnézetben.

Mivel a harmonikus koordináták használatának az egyik feltétele az, hogy a manipulálni kívánt felület minden egyes pontja a ketrecen belül helyezkedjen el, így a ketrec létrehozásakor ez is egy lényeges követelmény volt. Ennek a teljesítésére a legbiztosabbnak tűnő megoldás az volt, hogy különböző segédpontokat alkalmaztunk a felhasználó

által megadandó 48 ponton kívül. A segédpontok halmazát a modellpontok x, y és z tengely szerinti maximumai és minimumai alkotják. Ezen pontok növekménnyel való ketrecekbe építésével garantálhatjuk azt, hogy a modell összes pontja a kontrollhálón belülre essen.

A 14. ábrán is látható, hogy az arcmodellekhez tartozó ketrecek megalkotásakor fontos szempont volt az, hogy a ketrecpontok az arcon egy-egy fontos, jellegzetes pontot jelöljenek, ahogy az MPEG-4 szabvány tartópontjai (lásd 12. ábra) teszik ezt. Ugyanis ezen pontok mozgatásakor az arc jellegzetes területei manipulálhatóak, így a rekonstruálni kívánt modellhez minél közelebbi deformációt fogunk kapni.

4.3. Az automatizált folyamat ismertetése

Az előző fejezetekben betekintést nyertünk mind a referenciamodell, mind a deformációs módszer jellemzésébe, nem maradt mást hátra, mint hogy ezeket felhasználva egy automatizált folyamatot hozzunk létre. Szerettük volna elérni azt, hogy a felhasználó által kötelezően megjelölendő szabványpontok meghatározásán kívül a procedúra automatikusan kerüljön végrehajtásra.

Ennek az implementálásában nagy szerepe volt a *Blender* [10] nyílt forrású szoftver alapját alapját képező *Python* szkriptrendszernek [7]. Ugyanis a modellező programban a harmonikus koordináták csakúgy, mint a Catmull-Clark subdivision egy-egy használható és beépített funkcionalitásként jelenik meg. Így nem kellett időt „pazarolnunk” a módszerek kódolásával, összpontosíthattunk a minél kifogástalanabb végeredmény elérésére. A automatizált folyamat lépésenkénti felépítése:

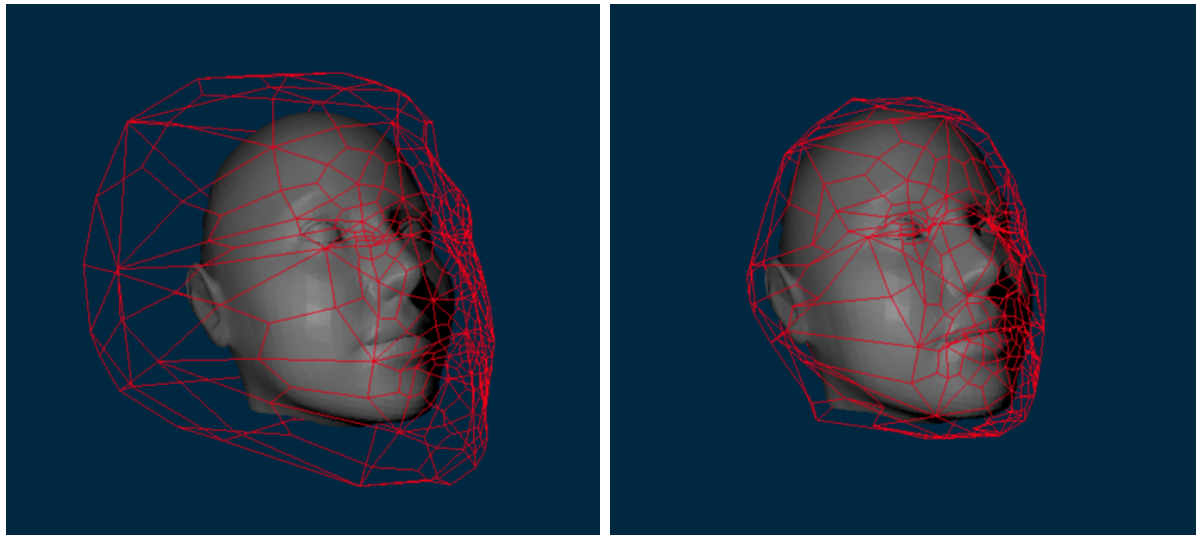
- Modellekhez tartozó ketrecek generálása.
- Egy 2 iterációs Catmull-Clark subdivision végrehajtása a ketrecekén.
- A ketrecek és a modellek közötti távolság minimalizálása.
- Harmonikus koordináták alkalmazása a modelleken a hozzájuk tartozó ketrecekkel.
- A referenciamodell ketrecpontjainak elmozgatása a rekonstruálandó modell azonos indexű ketrecpontjaiba.

Az eljárás első lépése a 4.2. fejezetben is említett modellekhez tartozó ketrecek legenerálása.

Majd ezt követően egy Catmull-Clark subdivision (3.2. fejezet) hajtódik végre, melynek szintén nagy jelentősége van abból a jellemzőből kiindulva, hogy a minél több pontból felépülő kontrollhálóval finomabb modellmanipuláció eszközölhető. Mi egy 2 iterációs változatot használtunk a szkriptben, így egy több, mint 300 pontot és 600 poligonlapot tartalmazó ketreccel dolgoztunk tovább.

Ezután a ketrecek és a modellek közötti távolság minimalizálásával folytatódik tovább a folyamat (lásd 15. ábra), amit a ketrecpontok z tengely szerinti vetítésével és az adott ponthoz tartozó legközelebbi modellpont meghatározásával kaptunk meg. Kihasználva ezzel azt a tényt, hogy a modellhez minél közelebb elhelyezkedő ketrecpontok elmozdításával nagyobb mértékű deformáció adódik. Erre szintén létezik beépített eszköz a *Blender*ben,

így a helyes paraméterezésen múlik csak a kívánt ketrec-átalakítás. Mindenképpen figyelni kell arra, hogy a modell összes pontjának továbbra is a ketrecen belül kell elhelyezkednie, így a harmonikus koordináták módszere továbbra is hiba nélkül alkalmazható.



15. ábra. A referenciamodell ketrecén alkalmazott *Catmull-Clark* subdivision, majd *Shrinkwrap*.

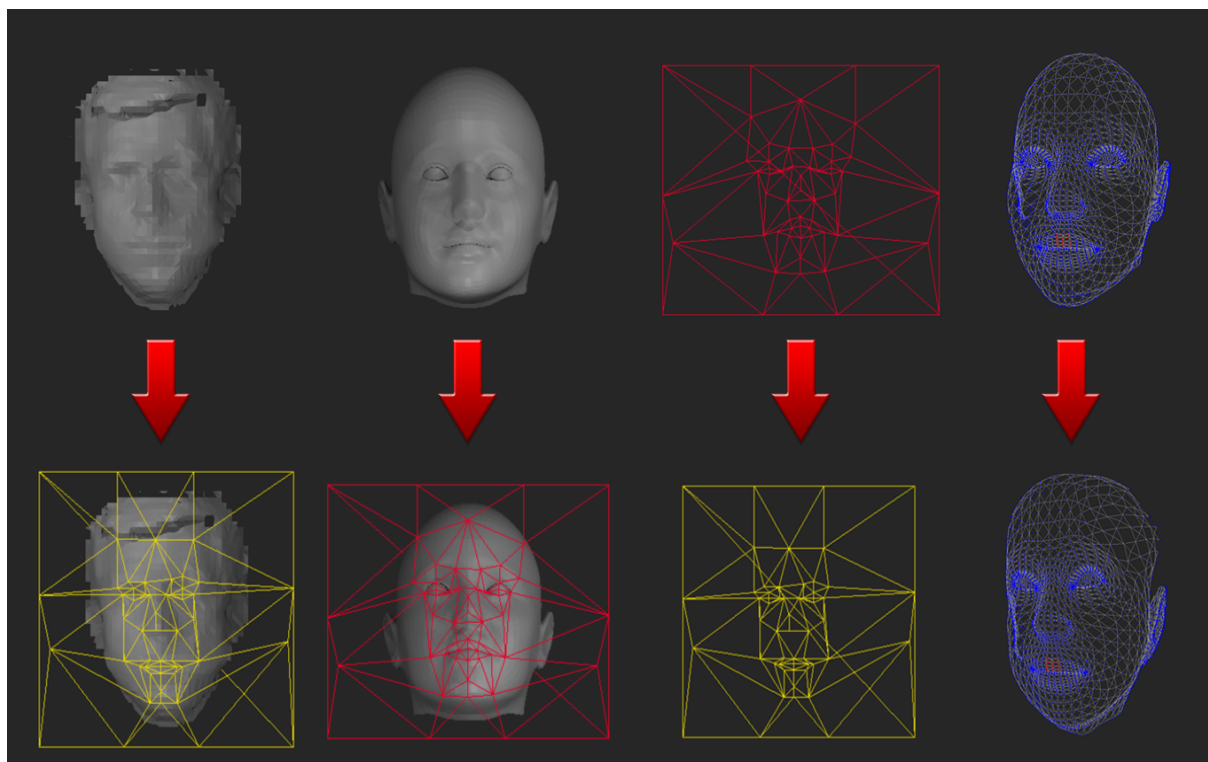
Így a számunkra kedvezően módosított ketreceket most már használhatjuk a harmonikus koordináták módszerével a modelleken. Ha a referenciamodellhez tartozó kontrollháló mindegy egyes pontját a rekonstruálandó modell kontrollhálójának megegyező indexű pontjába mozgatjuk el, akkor azt érzük el, hogy a referenciamodellhez tartozó kontrollháló a másik ketrec struktúráját, alakját veszi fel. Azaz a referenciamodell a rekonstruálandó személyhez tartozó arcmodellen megjelölt szabványpontok elhelyezkedésének a függvényében fog deformálódni. Számunkra előnyös paraméterezéssel a várt eredményhez egy kellően jól közelítő virtuális avatar modelljét kaphatjuk meg (lásd 17. ábra).

4.4. Eredmény optimalizálásának elősegítése

Ahogy a folyamat harmadik lépésében említettük, a kapott eredménynek a rekonstruált modellhez való hasonlósága nagy mértékben függ az ideális paraméterek megválasztásától. Ezért elkészítettünk egy olyan alkalmazást is, mely a különböző kapott és várt eredmény közötti hasonlóságot vagy különbözőséget méri és egy pozitív egész számként közli velünk.

Az eltérés mérésének az ötlete a következők szerint alakult. Felállítunk a referenciamodell minden pontjában egy azon átmenő és a z tengellyel párhuzamos egyenest. Majd a rekonstruálni kívánt modell minden poligonjának a pontjai által meghatározott síkokat írjuk fel. Kiszámoljuk minden egyenes metszéspontját a megfelelő síkkal, ezután a metszéspontok és a hozzájuk tartozó egyenespontok közötti távolságokat összegezve megkapjuk a két modell közötti teljes eltérést.

Ezen mérési folyamat felhasználásával tulajdonképpen kiszámíthatjuk azt a legjobb paraméterkombinációt, mely a legkedvezőbb eredményt fogja biztosítani.



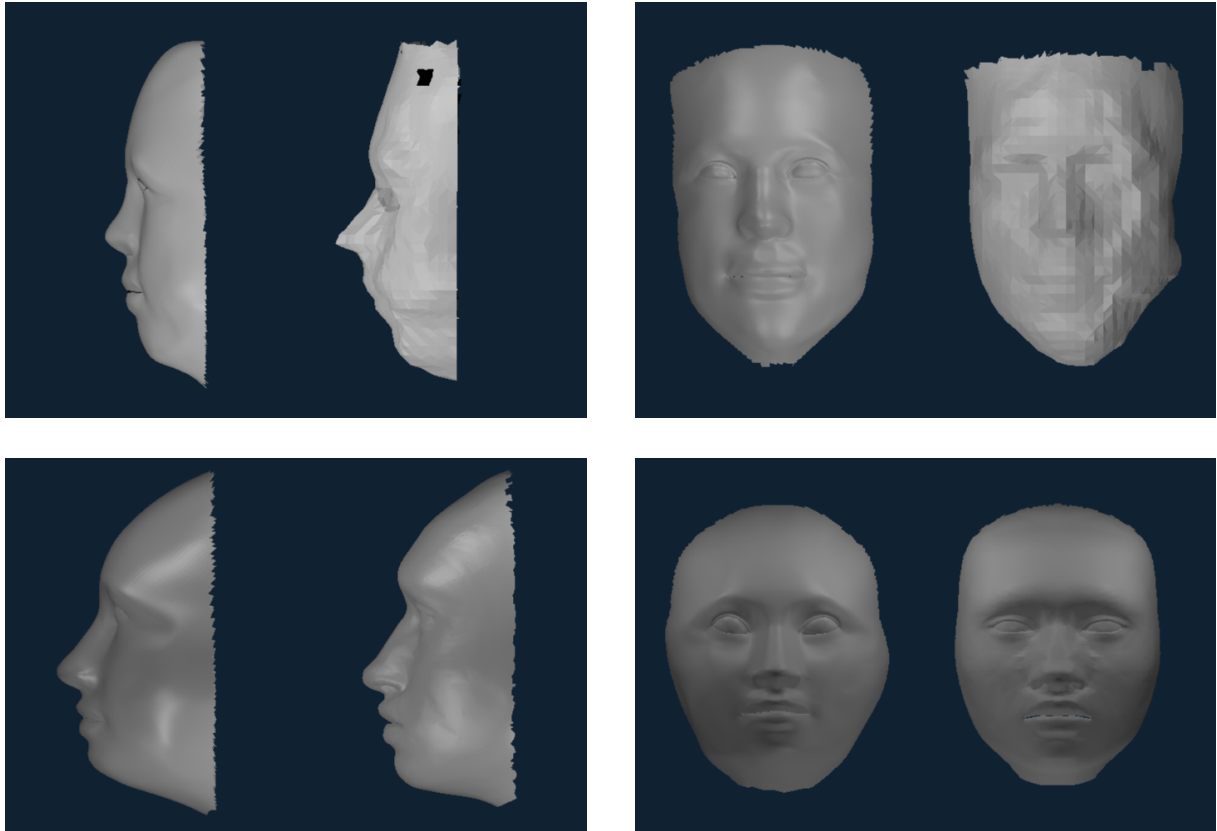
16. ábra. A félautomata folyamat legfontosabb lépéseit szemléltető folyamatábra. A folyamat első lépésében a kontrollháló kerül legenerálásra az egyes modellekhez (lásd 1. és 2. oszlop). Ezután a referenciamodell kontrollhálójának minden egyes pontja eltolásra kerül a rekonstruálni kívánt modell kontrollhálójának megfelelő pontjába. Majd a folyamat végeredményeként a rekonstruálni kívánt modellhez egy jól közelítő, szabványosított modellt kapunk. A 4. oszlopban szereplő modelleken pirossal jelöltük az ugyanazon FP ponthoz tartozó szabványpontokat.

4.5. Eredményeink összegzése

Céljaink között szerepelt egy olyan automatizált módszer megtervezése és elkészítése, mely segítségével virtuális avatárok szabvány szerinti felparaméterezését végezhetjük el. Már léteznek hasonló rendszerek, melyek felhasználásával megvalósítható a paraméterezés, azonban mindegyiknek meg vannak a maga hibái. Ilyen rendszer a *Virtual Human Director* (lásd 2.1. fejezet) is, mely kétdimenzióban, két fotó alapján működik. Nagy hátránya, hogy különböző szabványpontok elérhetetlenek, megjelölésük nehézkes, ezért hibát véthetünk a folyamatban.

Ezen hibák kiküszöbölésének érdekében olyan háromdimenziós rendszerek jelentek meg, amelyek esetében hasonló jellegű hibák már nem fordulhatnak elő. Ilyen eszköz az *Xface* (lásd 2.3. fejezet) is. A szoftver képes különböző forrásból származó 3D arcmodellek betöltésére, melyen a szabványpontok megadása jóval egyszerűbb, mint a fotókon.

Azonban, ahogy az előzőekben is említettük, az összes szabványpont manuálisan történő megadása időigényes folyamat. Ebből kifolyólag megalkottunk egy olyan folyamatot, mely elvégzi egy virtuális avatár MPEG-4 szabványosítását, úgy hogy az a lehető legkevesebb felhasználó interakciót igényelje.



17. ábra. A deformáció után kapott szabványosított modell és a kiindulási rekonstrukció.

Ahogy a 17. ábrán is látszik, az eredmény még nem tökéletes, de rengeteg olyan pontja van az algoritmusnak, ahol tovább javítható a modell. Gondolhatunk itt a ketrec pozícionálására vagy a ketrec és a modell közötti differencia más módon történő mérésére.



18. ábra. A textúrázott modell.

Nyilvánvaló, hogyha a modellhez elkészítjük a textúrákat jó minőségű fotók alapján (lásd 18. ábra), akkor ezen különbségeket szabad szemmel már semennyire vagy nehezebben vesszük majd észre.

5. Összefoglalás

A virtuális avatárt készítő rendszereknek fontos részét képezi mind a megfelelő rekonstrukció lépése, mind a már kész modell helyes paraméterezése annak érdekében, hogy a végeredmény minél élethűbb legyen. A rekonstrukciónak a fotorealisztikusságot célul kitűző rendszerek esetében valóságúnak és részletesnek kell lennie, hogy elhiggyük, hogy egy létező személy beszél hozzánk a monitorról, nem pedig egy számítógéppel generált fiktív figura. Míg a paraméterezésnek az artikulálásban és beszédképzésben van elengedhetetlen szerepe.

A dolgozat keretein belül megalkottunk egy olyan hatékony módszert, mely segítségével egy már tetszőlegesen rekonstruált emberi fejmodell MPEG-4 szabvány szerinti paraméterezése félautomata módon elvégezhető, ezzel felkészítve a későbbi animálásra a virtuális avatárt. Különösképpen fontosnak éreztük a minél nagyobb szintű automatizálást, ugyanis a manuálisan végzett szabványosítás rengeteg időt vesz igénybe.

Az ötletünk az volt, hogy meghatározunk egy referenciamodellt, melyen az összes szabványpont már meg van jelölve. Ezt a modellt pedig úgy próbáljuk alakítani, hogy a rekonstruálni kívánt személy modelljéhez minél jobban közelítsen. Ezzel a szabványosítás folyamata elhagyható.

A módszer alapját a baricentrikus koordináták általánosított változata, a harmonikus koordináták képezi, melyet a *Pixar Animation Studios* mutatott be 2007-ben. Megannyi kutató munka és kísérletezés után arra jutottunk, hogy a számos deformációs technika közül ez lesz a megfelelő arra, hogy egy emberi arc rekonstruált modelljéből egy virtuális avatárt könnyen el tudjunk készíteni.

A módszer fontos része a modell körül felállított kontrollháló vagy ketrec, mellyel az deformálható, alakítható. Így feladatunk volt egy olyan algoritmus elkészítése is, mely egy tetszőleges modell ketrecét automatikusan legenerálja egy információs fájl segítségével. A teljes algoritmus egyetlen olyan része, melyet manuálisan végez a felhasználó. Ketrec definiálása közben fontos szempont volt az, hogy olyan pontokból alkossuk meg, melyek a modellen az emberi arc egy-egy jellegzetes vonását jelölik. Így deformáláskor ezen területekre jut a legnagyobb hangsúly, ezzel a hasonlóság mértéke növelhető.

Megalkottuk a referenciamodell módosításáért felelős szkriptet is, melynek egy rekonstruált arcmodellt és a hozzá tartozó ketrecet bemenetként megadva automatikusan kapjuk az eredményt. Elkészült egy olyan szoftver is, ami az ugyanazon személyhez kapott végeredmények közötti különbségek összevetésére alkalmas. Ezzel próbáltuk finomítani a kapott modellek minőségét. Megjegyzendő, hogy a kapott eredménnyel még mi sem vagyunk elégedettek, de jó pár olyan potenciális pontja van a szkriptnek, melynek módosításával drasztikusan javítható a kapott modell és a virtuális avatár alanyát képző személy közti hasonlóság.

Céljaink között szerepel majd ezen részek javítása, optimalizálása. Miután a hibákra megfelelő megoldásokat találtunk, szeretnénk elkészíteni egy olyan webes felületet, mely használatával egy Kinecttel felszerelt számítógép előtt ülve, bárki elkészítheti saját virtuális avatárját.

6. Saját munka leírása

- A harmonikus koordináták módszeréhez szükséges ketrec 3d fejmodellekre való általánosítása.
- Automatikus ketrecgenerálás tetszőleges fejmodellhez.
- Automatikus fejmodell-deformációt végző szkript elkészítése.
- Eredmény optimalizálásában segítő alkalmazás fejlesztése.

7. Köszönetnyilvánítás

Ezúton szeretném megköszönni témavezetőmnek, *Kunkli Rolandnak*, hogy szakdolgozatom elkészítése után bizalmát bennem megtartva a TDK dolgozatom megírásának is koordinátora volt. Hálás vagyok, hogy segítőkészségére mindig számíthattam. Köszönettel tartozom, hogy gondjaimmal és nehézségeimmel bármikor kereshettem, idejét nem sajnálva segített nekem. Szakmai tanácsai, ötletei és meglátásai nélkül ez a dolgozat nem készülhetett volna el.

8. Irodalomjegyzék

- [1] R. Kunkli Á. Tóth: Standardize me! http://www.inf.unideb.hu/~kunkli.roland/project/first/standardize_me.html. Elérés időpontja: 2014.10.14.
- [2] Zachary Abel: *Barycentric Coordinates*. 2007. http://zacharyabel.com/papers/Barycentric_A07.pdf. Elérés időpontja: 2014.10.04.
- [3] Niloofer Aghayan: *Reconstruction of Complete Head Models with Consistent Parameterization*. PhD értekezés (University of Ottawa). 2014.
- [4] Tóth Ákos: Kinecttel rekonstruált 3d fejmodell mpeg-4 szabvány szerinti vezérlése, 2014. Szakdolgozat (Debreceni Egyetem Informatikai Kar).
- [5] Koray Balcı: Xface: Open source toolkit for creating 3d faces of an embodied conversational agent. In *Smart Graphics* (konferenciaanyag). 2005, Springer, 263–266. p.
- [6] Stefano Berretti – Alberto Del Bimbo – Pietro Pala: 3d face reconstruction from two orthogonal images for face recognition applications. *International Journal of Digital Library Systems (IJDLs)*, 1. évf. (2010) 3. sz., 42–58. p.
- [7] Blender python api. http://www.blender.org/api/blender_python_api_2_72_1. Elérés időpontja: 2013.10.13.
- [8] Peter J. Burt – Edward H. Adelson: A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, 2. évf. (1983) 4. sz., 217–236. p.
- [9] Cognitive – Communication Technologies (TCC):. <http://xface.fbk.eu/>. [Szoftver] Elérés időpontja: 2013.10.14.
- [10] Blender Online Community:.. <http://www.blender.org>. [Szoftver] Elérés időpontja: 2013.10.13.
- [11] Timothy F. Cootes – Gareth J. Edwards – Christopher J. Taylor: Active appearance models. In *Computer Vision (ECCV'98)*. 1998, Springer, 484–498. p.
- [12] Tony DeRose – Michael Kass – Tien Truong: Subdivision surfaces in character animation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98 konferenciasorozat. New York, NY, USA, 1998, ACM, 85–94. p. ISBN 0-89791-999-8. 10 p.
- [13] Tamal K. Dey – Pawas Ranjan – Yusu Wang: Eigen deformation of 3d models. *The Visual Computer*, 28. évf. (2012) 6-8. sz., 585–595. p.
- [14] Miklós Hoffmann: Modified kohonen neural network for surface reconstruction. *Publ. Math. Debrecen*, 54. évf. (1999), 857–864. p.
- [15] Shahram Izadi – David Kim – Otmar Hilliges – David Molyneaux – Richard Newcombe – Pushmeet Kohli – Jamie Shotton – Steve Hodges – Dustin Freeman – Andrew Davison és mások: Kinectfusion: real-time 3d reconstruction and interaction using a

- moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (konferenciaanyag). 2011, ACM, 559–568. p.
- [16] Pushkar Joshi – Mark Meyer – Tony DeRose – Brian Green – Tom Sanocki: Harmonic coordinates for character articulation. In *ACM Transactions on Graphics (TOG)* (konferenciaanyag), 3. köt. 2007, ACM, 71. p.
- [17] Tao Ju – Scott Schaefer – Joe Warren: Mean value coordinates for closed triangular meshes. In *ACM Transactions on Graphics (TOG)* (konferenciaanyag), 3. köt. 2005, ACM, 561–566. p.
- [18] Christina Koblbauer: *Barycentric Coordinates*. Waterloo, 2012. http://koblbauermath.weebly.com/uploads/1/3/1/9/13192946/barycentric_coordinates.pdf. Elérés időpontja: 2014.11.02.
- [19] Won-Sook Lee – Marc Escher – Gael Sannier – Nadia Magnenat-Thalmann: Mpeg-4 compatible faces from orthogonal photos. In *Computer Animation, 1999. Proceedings* (konferenciaanyag). 1999, IEEE, 186–194. p.
- [20] Hao Li – Jihun Yu – Yuting Ye – Chris Bregler: Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics (TOG)*, 32. évf. (2013) 4. sz., 42. p.
- [21] Jonas Lundberg: Mpeg-4 facial feature point editor. 2002. Diplomamunka (Linköping University).
- [22] Iain Matthews – Simon Baker: Active appearance models revisited. *International Journal of Computer Vision*, 60. évf. (2004) 2. sz., 135–164. p.
- [23] The moving picture experts group. <http://mpeg.chiariglione.org/standards/mpeg-4>. Elérés időpontja: 2013.09.09.
- [24] Jesus R. Nieto – Antonio Susin: Cage based deformations: a survey. In *Deformation Models*. 2013, Springer, 75–99. p.
- [25] Frederick I. Parke: Computer generated animation of faces. In *Proceedings of the ACM annual conference-Volume 1* (konferenciaanyag). 1972, ACM, 451–457. p.
- [26] Guodong Rong – Yan Cao – Xiaohu Guo: Spectral mesh deformation. *The Visual Computer*, 24. évf. (2008) 7-9. sz., 787–796. p.
- [27] Levente Sajó – Miklós Hoffmann – Attila Fazekas: A 3d head model from stereo images by a self-organizing neural network. *Journal for Geometry and Graphics*, 13. évf. (2009) 2. sz., 209–220. p.
- [28] The MakeHuman team.: <http://www.makehuman.org>. [Szoftver] Elérés időpontja: 2014.09.26.
- [29] Thibaut Weise – Sofien Bouaziz – Hao Li – Mark Pauly: Realtime performance-based facial animation. *ACM Transactions on Graphics (TOG)*, 30. évf. (2011) 4. sz., 77. p.

- [30] Michael Zollhöfer – Michael Martinek – Günther Greiner – Marc Stamminger – Jochen Süßmuth: Automatic reconstruction of personalized avatars from 3d face scans. *Computer Animation and Virtual Worlds*, 22. évf. (2011) 2-3. sz., 195–202. p.