

Diplomaterv

Fejmodellek továbbfejlesztése

Pozsgay Tamás

Témavezető: Takács György és Tihanyi Attila

Pázmány Péter Katolikus Egyetem

Információs Technológiai Kar

2007.

Nyilatkozat

Alulírott Pozsgay Tamás, a Pázmány Péter Katolikus Egyetem Információs Technológiai Karának hallgatója kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, és a diplomamunkában csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen a forrás megadásával megjelöltem. Ezt a Diplomamunkát más szakon még nem nyújtottam be.

Tartalomjegyzék

1. Bevezetés.....	8
1.1. A virtuális lények szerepe a mai világban.....	8
1.1.1. A szórakoztató iparban és a kereskedelemben.....	8
1.1.2. Tudományos téren.....	9
1.2. A VRML-ről röviden.....	11
2. Fejmodellek.....	11
2.1. Anatómiai modellek.....	12
2.2. Geometriai modellek.....	12
2.3. A végtelen sok modell problémájának feloldása.....	15
3. Az MPEG-4 Facial Animation szabvány.....	15
3.1. FP – Feature Point.....	16
3.2. FAP – Facial Animation Parameter.....	18
3.3. FAPU - Facial Animation Parameter Unit.....	19
3.4. XML.....	20
4. A konzultációk.....	20
5. Vizuális jellemzők.....	22
6. A PPKE fejanimáló rendszere.....	23
6.1. Az audiovizuális adatbázis felvétele:.....	24
6.1.1. A bemondók.....	24
6.1.2. A példamondatok.....	26
6.1.3. A felvételek.....	26
6.2. Neurális Hálózat.....	27
6.3. A Mobilkészüléken megjelenő végeredmény.....	27
7. Az általunk használt fejmodell:.....	28
7.1. A Lucia.....	28
7.2. Lucia mozgatása.....	28
8. Saját fejanimáló rendszerem.....	31
8.1. Az OpenGL:.....	33
8.2. Egyedülálló pont 2D-s statikus megfigyelése.....	33
8.3. Két, vagy több pont 3D-s statikus megfigyelése.....	36
8.4. Több FP statikus vizsgálata egy szakadással, a meghajlított síkon.....	40
8.5. Pontok áthelyezése a saját modellembe.....	44

<u>9. Javaslat a rajzolás gyorsítására.....</u>	<u>46</u>
<u>10. Következtetések.....</u>	<u>47</u>
<u>11. Irodalomjegyzék.....</u>	<u>50</u>

Összefoglalás

Célkitűzésünk a siketek és halláskárosultak számára is elérhető mobiltelefon alkalmazás létrehozása volt.

Aki siketekkel szeretne kommunikálni két lehetőség közül választhat: az egyik, hogy jeltolmácsok által is használt jelekkel él, a másik, hogy jól artikulálva beszél hozzájuk, bízva abban, hogy a nagy gyakorlatra szert tett siketek, vagy nagyothallók könnyűszerrel leolvassák a szájmozgást.

A mi koncepciónk ezen a második lehetőségen alapul. A feladatunk egy olyan animált fejmodell tervezése, amely részleteiben elég gazdag, elég nagy felbontású, elég szépen és pontosan artikulál, ahhoz hogy a szájáról le lehessen olvasni a beszédet. Az alkalmazásnak valósidejűnek kell lennie, vagy ahhoz közeleink és közvetlenül a beszédből kell megformálnia a szájállást reprezentáló animációt, tehát egy speech-to-animation konverzióra van szükség.

Az egyetemi projektben két csapat vesz részt. A feladat két párhuzamosan végezhető munkafolyamatán dolgoznak egyszerre. Az egyik csoport feladata, hogy megvalósítsa ezt az alkalmazást PC-s környezetben, míg a másik csoport átalakítja, és mobiltelefonon is futtathatóvá teszi. Munkám során a PC-s eredmények bemutatására szorítkozom, mert ezen a területen dolgoztam.

Fejlesztésünk az MPEG-4 adattömörítési eljárason alapul. Munkám során támaszkodtam a korábban, más kutatócsoportok által létrehozott fejmodellekre és a siketekkel való folyamatos konzultáció eredményeire, hogy végül egy egyszerű de jól használható új modellt tudjak létrehozni.

Részletes leírást adok az általam fejlesztett programról. A program segítségével az általunk használt fejmodellek hiányosságait és hibáit tudom felderíteni.

A következő megoldási javaslatot teszem a programommal végzett kísérletek alapján:

Az általunk használt nyílt forráskódú fejmodell összes tartópontjának hatókörét felül kell vizsgálni. Így megszüntethető a jelenlegi modellben a textúrának szétszakadása a nagy elmozdulásokkor. Ezzel jelentős javulás érhető el a szájról való leolvasásban.

Summary

Our aim was to create a mobile phone application for deaf and hearing impaired people.

There are two possibilities for communication with deaf. One of them is the usage of sign-language and the other one is to speak with emphasized articulation, counting on the good lip reading practice of deaf or hearing impaired people.

Our conception is based on the second possibility. Our task was to plan an animated head model, which is sophisticated in details, has quite high resolution and articulates precisely enough to allow reading speech from the mouth. The application should be real-time, or quasi-real-time and it has to realize lip-animation from the speech, thus we need a speech-to-animation conversion.

Our university project is divided into two subprojects. Two teams are working in parallel on separated tasks. The task of the first team is to realize the application in PC environment, and the second team converts the animation and implements it on mobile phone. I worked in the first team, so I will present our results in PC environment.

Our development is based on the MPEG-4 data compression method. I relied on the head models created by other research teams, a permanent consultation was carried out with deaf people in order to create a simple but useful head model finally.

A detailed description is given about the program I developed. By means of the program the defects and shortcomings of head models can be identified and analyzed.

On the basis of experiments by the program I suggest the following solution:

All scopes of the feature points of the currently applied open-source head model should be reconsidered. With this solution in the case of big movements we can avoid the texture to be broken. This way a significant improvement can be achieved in lip reading.

1. Bevezetés

Ahogy az Internet előfizetők száma rohamosan nő az egész világon, úgy hódítanak teret az online alkalmazások is. Ma Magyarországon több mint 3 millióan (30,4 %) használják rendszeresen az Internetet, de ugyanez a szám 2007 márciusára az Európai Unión belül 250 millióra emelkedett, ami az EU népességének több mint 50 százaléka. Sok kutató jóslata szerint a mai hatalmas iramú fejlődés nem fog a közeljövőben sem alábbhagyni. Itt elsősorban Gilder törvényére [GILDER] gondolok: az egész kommunikációs rendszerre számított sávszélesség minden 12 hónapban megtriplázódik. Ma már szinte bármit el lehet intézni az Internet segítségével: vásárlás, jegy rendelés (pl.: repülőjegy, koncert jegy), banki átutalás, online tv adások követése, ügyintézés, szórakozás stb. Ebben a világban próbálják az eligazodást segíteni a virtuális lények, melyek a vizuális beszédszintézis térnyerésével rohamos fejlődésnek indultak.

1.1.A virtuális lények szerepe a mai világban

Észre sem vesszük, de az élet szinte minden területén lépten-nyomon virtuális lényekbe, vagy virtuális világokba botlunk.

1.1.1.A szórakoztató iparban és a kereskedelemben

Ahogy az internetelőfizetők száma rohamosan nő az egész világon, úgy hódítanak teret az online alkalmazások is. Ma Magyarországon több mint 3 millióan (30,4 %) használják rendszeresen az Internetet, de ugyanez a szám 2007 márciusára az Európai Unión belül 250 millióra emelkedett, ami az EU népességének több mint 50 százaléka. Sok kutató jóslata szerint a mai hatalmas iramú fejlődés nem fog a közeljövőben sem alábbhagyni. Itt elsősorban Gilder törvényére gondolok: az egész kommunikációs rendszerre számított sávszélesség minden 12 hónapban megtriplázódik. Ma már szinte bármit el lehet intézni az Internet segítségével: vásárlás, jegy rendelés (pl.: repülőjegy, koncert jegy), banki átutalás, online tv adások követése, ügyintézés, szórakozás stb.

Ebben a világban próbálják az eligazodást segíteni a virtuális lények. Számos online játék rendelkezik óriási felhasználói bázissal. A több tízezer felhasználót számláló virtuális világok minden objektuma 3D-modellező eszközökkel jött létre, melyeket - némi korlátozással - maguk a játékosok tölthetnek fel. A tagoknak mindössze egy egyszeri pár dolláros tagsági díjat kell átutalniuk. Egy másik terület a kereskedelem.

Igen sok honlapon (elsősorban külföldin) lehet találkozni virtuális eladóval, aki/ami elmondja nekünk a kiválasztott árucikkről az információt amire kíváncsiak vagyunk. Sokszor valamilyen demó, vagy tutorial keretében tűnnek fel a virtuális lények, ahol elmagyarázzák, megmutatják például egy új program használatához szükséges tudnivalókat (1. ábra).



Thanks for coming. Let me show you
around the place.

ábra 1: Egy tutorial virtuális előadója

De a játék és egyéb szórakoztatóiparbeli alkalmazások, illetve az értékesítés mellett előfordulnak más, számunkra sokkal érdekesebb, pl.: tudományos területeken is ezek az apró, a képernyőn lakó emberkék.

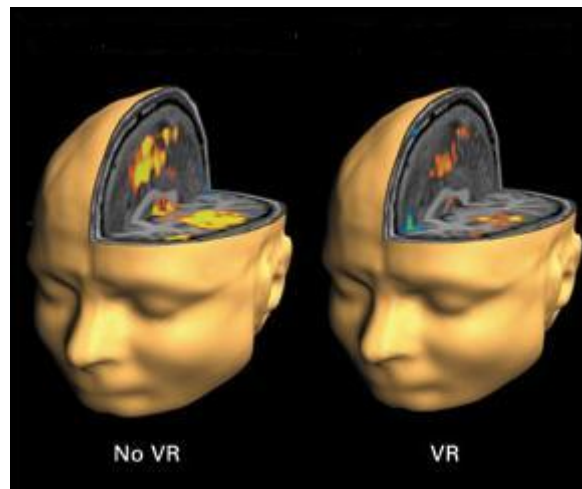
1.1.2. Tudományos téren

A gyógyításban kezdenek elterjedni olyan megoldások, amelyek egy virtuális világba helyezik a beteget. Például sok ember nem mer elmenni pszichológushoz, ha valamilyen számára megoldhatatlan lelki problémával kerül szembe. Ma már léteznek virtuális pszichológusok, így a páciensnek nem egy hús-vér embernek kell felfednie a magánéleti gondjait, vagy az élete folyamán szerzett mély sebeit, hanem egy virtuális embernek. További hatalmas előnye a virtuális lényeknek, virtuális világoknak, hogy testre szabhatóak a benne szereplő emberek, vagy tárgyak. A páciens megválogathatja a pszichológusait, esetleg ő ruházza fel bizonyos tulajdonságokkal (ha támogatja ezt a gyógyító szoftver), amire a valós világban – valljuk be – aligha lenne lehetősége, pedig milyen jó lenne néha.

Egy, a Washington Egyetemen működő kutatócsoport 1996 óta foglalkozik a fájdalomcsillapításnak azzal a módszerével, hogy egy virtuálisvalóság-sisak segítségével megpróbálják kiragadni a beteget a valós világból, megpróbálják elvonni a figyelmét a fájdalomérzésről.

Első kísérletük során azt vizsgálták, hogy milyen fájdalomélményről számol be két beteg, akik égési sérülésük kezelése előtt morfiumot kaptak, de a kötéscsere alatt az akkoriban elérhető legjobb virtuálisvalóság-programmal, játszottak. Az égési sérült betegeket, egy külön erre a célra kifejlesztett virtuális világba kalauzolták, ami valami

hideg, havas sarkvidéki, vagy magashegységi, hóemberekkel, pingvinekkel tarkított tájat ábrázolt, ezzel is segítve a sérülteket a valós világtól és a saját égési fájdalmuktól való elrugaszkodásban. A kísérlet eredménye az lett, hogy amikor a két beteg a programmal játszott, 40-50 százalékos fájdalomcsökkenésről számolt be (2. ábra).



ábra 2: Az agyi tevékenység csökkenése a virtuális világban

A jelenség magyarázata, hogy az emberi figyelem olyan, hogy kiválaszt, ráfókuszál valamilyen információ feldolgozására, a többit pedig figyelmen kívül hagyja, mivel csak korlátozott mennyiségű információt tudunk egyszerre befogadni. Amikor a betegek a virtuálisvalóság-programmal játszottak, figyelmük a számítógép által generált világra összpontosult, és kevesebb figyelmet fordíthattak fájdalmukra. Ezt a kutatók mérésekkel is igazolták. A betegeket fMRI-szkennerekbe fektették a kötéscsere idejére és azt figyelték meg, hogy játék közben jelentősen (50%) csökkent az agy aktivitása öt olyan területen, ami közvetlenül a fájdalom érzékeléséhez kapcsolódik.

Ebből a kis áttekintőből is látszik, hogy milyen sok alkalmazási területen jelennek meg a virtuális lények. Nagy előnyük a változatosságuk, hisz a már megírt modelleket a textúrázás segítségével annyiszor reprodukálhatjuk, ahányszor csak tetszik és mégis külsőre különböző lényeket kapunk.

A mi feladatunkhoz azonban főleg az egyszalakos modelleket készíteni, hisz a siketek és halláskárosultak szempontjából a fejnek, azon belül pedig a szájnak és környékének van kiemelkedően fontos szerepe. Elég volt csak a fejmodellekre koncentrálni a figyelmünket a kutatómunka folyamán.

A legtöbb fejmodell és általában a virtuális világok VRML nyelvben vannak megírva, ez az elterjedt nyelv a modellezők között. Ezenkívül a nagy 3D-s grafikus fejlesztő és modellező programok is ezt használják, és lehetőség van az ezekben a programokban felhasználóbarát módon fejlesztett (nem kellene hozzá előzetes ismeretek, és majdnem minden megvalósítható néhány egérgattintással) modelljeinket vrml formátumban elmenteni.

1.2.A VRML-ről röviden

A VRML-t (Virtual Reality Modelling Language) a Web3D konzorcium hozta létre. A nyelv alapvető célja, hogy egyszerű eszközökkel háromdimenziós világokat hozzunk létre, melyeket egy erre a célra készített böngésző képes megjeleníteni. Ezen a területen általában világokról beszélünk, erre utal a VRML fájl kiterjesztése is, a .wrl, ami az angol world, világ szóból származik. Ez a nyelv egy egyszerű scriptnyelv. A VRML nyelvnek az egyik legnagyobb erőssége az egyszerűségében rejlik, hiszen a világok létrehozásához a készítőnek csak egy szövegszerkesztőre van szüksége, mert a világoknak leírása egy text fájl megírását jelenti. Ebbe a fájlba az alkotó bizonyos szintaktikai szabályok szerint leírja a világot felépítő objektumokat, azok geometriáját, transzformációit, kinézetét, anyagjellemzőit, megvilágítását, textúráit egymáshoz való viszonyát stb. A böngészők, vagy egyéb megjelenítő programok ezt a szöveges dokumentumot értelmezik, és jelenítik meg az alkotó által megálmodott világot.

A VRML 1.0 verziója lehetővé tette az alakzatok leírását viszonylag egyszerű szintaktikával. Ez a verzió azonban még nem volt képes arra, hogy az alkotó animációkat is elhelyezzen a világba. Az animációk nélkül azonban nem igazán képzelhető el a virtuális valóság. Így rövid idő elteltével – 1997-ben – megjelent a VRML 2.0-ás változata, melyet VRML97-nek is szoktak nevezni. Ebben a verzióban sokkal szabadabb kezet kapnak a modellezők.

2. Fejmodellek

Azonban a modellezéssel már sokkal régebb óta foglalkoznak, jóval az első VRML modellek megjelenése előtt. Két különböző filozófia szerint lehet fejmodelleket

tervezni. Az egyik az anatómiát és a fej felépítését veszi alapul, míg a másik a geometriára épít, és geometriai elemekből áll össze a modell.

2.1. Anatómiai modellek

Az 1970-es évek végén Paul Ekman és W. V. Friesen kidolgozott egy olyan modellt, amely az arc biológiáján alapszik (FACS – Facial Action Coding System ≈ Arcmozgás kódoló rendszer) [2]. Arcmozgásról elemeztek videofelvételeket, és azt írták le pontosan, hogyan húzódnak össze az arc izmai külön-külön, illetve azokat az izommozgásokat rendszerezték, amelyek kihatással vannak egymásra (3. ábra). Ennek a rendszernek az egységei (action unit) nem izmok, melynek két oka is van. Több esetben nem csak egy izom van egy egységbe tömörítve, hanem több, mert némely mozgás nem elkülöníthető. Más esetekben viszont egy izom által létrehozott mozgást kettő, vagy több egységre osztottak. Ekman és munkatársai ezzel a rendszerrel 64 mozgatható egységet definiáltak.



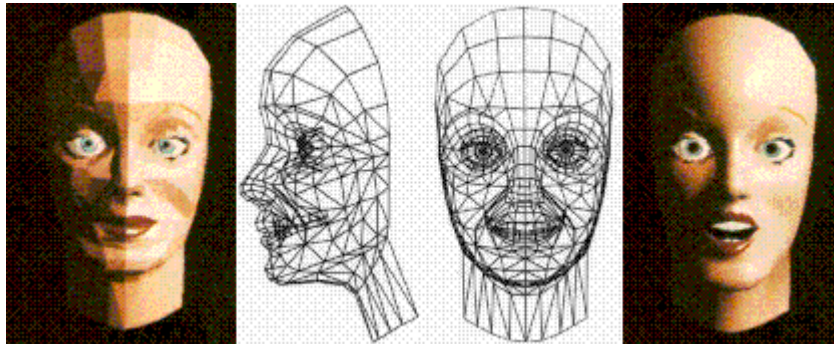
ábra 3: A FACS modell

Ennek a modellnek óriási hátránya, hogy nagyon számításigényes, ezért sohasem terjedt el széles körben.

2.2. Geometriai modellek

Az anatómiai modellezéstől különböző geometriai alapú modellezési stílus vált elterjedtté. A geometriai alapú modelleknek köszönhetően az építőelemek mérete, elhelyezkedése és, adott esetben, a kapcsolódási pontjaik könnyen változtathatók lettek, ezáltal paraméterezhető lett a fejmodell. Parke 1982-ben alkalmazta ezt a

módszert a fejanimálásra [5]. Az ő fejmodellje 800 négyszögletű polygonból állt, mely az arc részleteit is (szemek, szemöldök, fogak és ajak) tartalmazta (4. ábra).



ábra 4: Parke modelljének felépítése

A térben elhelyezkedő pontokat nevezzük vertexeknek, míg a vertexek által kifeszített, valamilyen sokszög oldalú felületeket polygonnak. Az arc mozgatására körülbelül 50 paramétert használt, azok segítségével mozgatta a vertexeket, aminek hatására a polygon felületek is deformálódtak. Ezek a deformációk szolgálták a fej animálására. Parke a paraméterek meghatározására nem használt anatómiai megfontolásokat, mint Ekman, tehát ebből a szempontból sokkal egyszerűbb a modellje. Ő inkább a megfigyelésekre hagyatkozott. Waters 1987-ben (Ekman modelljétől függetlenül) dolgozta ki a saját sémáját, az izommozgást alapul véve, melyet alkalmasnak talált az arckifejezések reprezentálására. [4] Ez a módszer közelebb állt a valósághoz, azonban alkalmatlannak bizonyult az ajakmozgások reprezentálásához az izomzat bonyolultsága miatt. Mint ahogy az Ekman modell, Waters modellje is áldozatul esett a bonyolultságnak és az ebből fakadó nagy számításigénynek. Parke direkt paraméterezési modellje megvalósíthatóbbnak látszott, aminek következtében szinte az összes fejanimáló rendszernek ez lett az atyja.



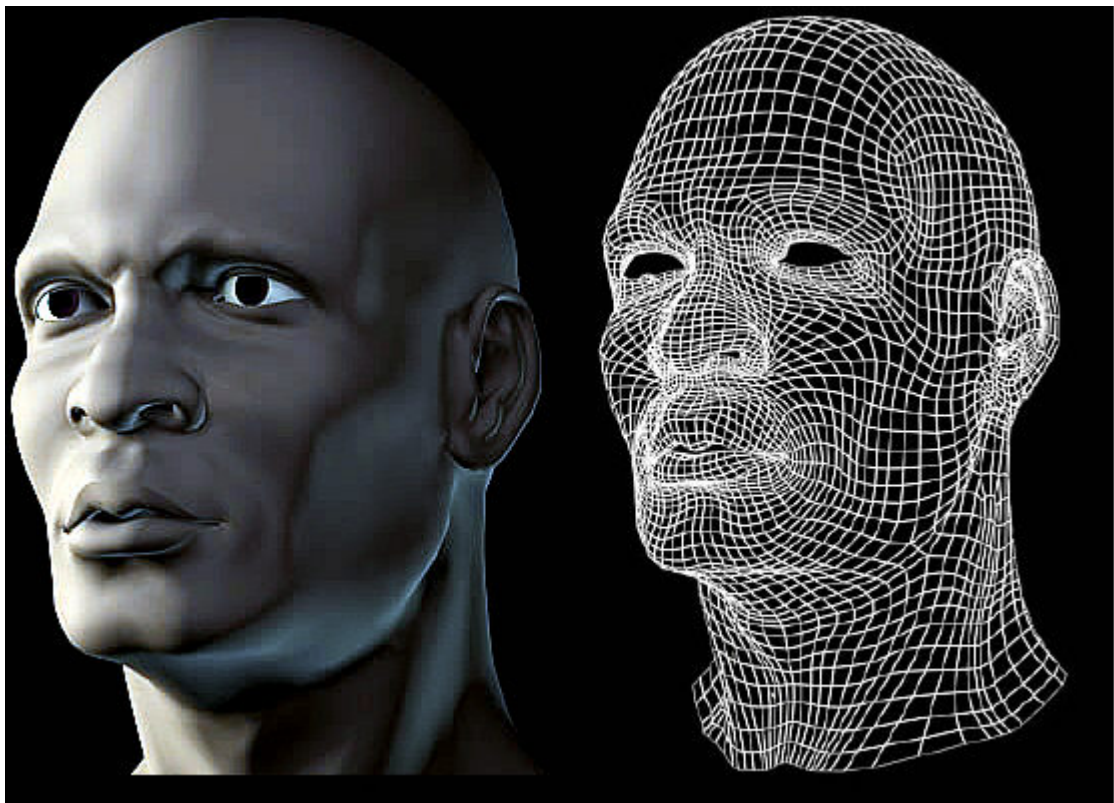
ábra 5: Megvásárolható szájmodell

A mai grafikus fejlesztő programokban (3D Studio, vagy ehhez hasonló) lehetőség van bármely általunk kívánt modell létrehozására. Az emberi arcot geometriai

építőkövekből lehet összerakni, olyan vertexekből és polygonokból, melyek jól közelítik a fej alakját (5. ábra). Az így kapott 3 dimenziós vázra kell a texturát ráhúzni, ami a modell külső megjelenítését adja meg. A textúra alatt lévő több ezer rácspontot egyenként lehet mozgatni, a modell „bőre” pedig idomul hozzá, felveszi a rácspontok által kirajzolt alakzatot, együtt mozog velük.

Azonban egy ilyen modell elkészítése meghaladta volna a képességeinket, ezek a programok túl bonyolultak ahhoz, hogy egy ilyen részletességű és ennyire összetett fejmodellt létrehozzunk bennük.

Egy másik lehetőség lett volna, hogy veszünk egy előre elkészített fejmodellt az interneten keresztül, azonban ezeknek általában igen magas az áruk.[15] Itt még arra is lett volna lehetőség, hogy csak egy szájmodellt vásárolunk (6. ábra).



ábra 6: Egy, az Interneten megvásárolható fejmodell

Néhány éve létezik egy olyan újfejlesztésű modell, amely a száját egy ellipszissel modellezi [8]. Az ellipszis két paramétere az ajakszélességnek és az ajaknyílásnak felel meg ebben a modellben. A modell egyik legnagyobb erőssége, az eddig tárgyalt összes többivel szemben, hogy összesen két paraméter segítségével írja le a szájmozgást, így paraméter csökkentésre kiválóan alkalmas. A mi modellünknel,

ahogy a későbbiekben ez majd kiderül, ez tízszer kevesebb paraméterrel dolgozik. Mi mégsem dolgozhattunk sajnos ezzel a modellel, mert ragaszkodnunk kellett az MPEG-4 szabványhoz.

2.3.A végtelen sok modell problémájának feloldása

Az előző, igen tömör, csak a legfontosabb modelleket megemlítő összefoglalóból is kiderül, hogy minden kutató, művész, animációs stúdió a saját maga által kifejlesztett modellt használta és ezek a modellek között semmiféle átjárhatóság nem létezett. Az általuk kidolgozott rendszerek modellfüggők. Ez a legnagyobb hátrányuk. A paraméterek nem egy általános fejanimáló rendszerhez vagy szabványhoz lettek igazítva, például a paraméterek számát vagy értelmezési tartományát tekintve, hanem az adott modellekhez. Ezek más fejformára vagy fejmodellre nem működnének. Ezért nőtt meg az igény az 1990-es évekre, az arc és az test animálásának valamilyen szintű szabványosítására, főleg a paraméterezés sokszínűségének általánosítására és a modellfüggőség feloldására. Ez a szabvány lett az MPEG-4 Face and Body Animation, ami a mai napig egyedülálló.

3. Az MPEG-4 Facial Animation szabvány

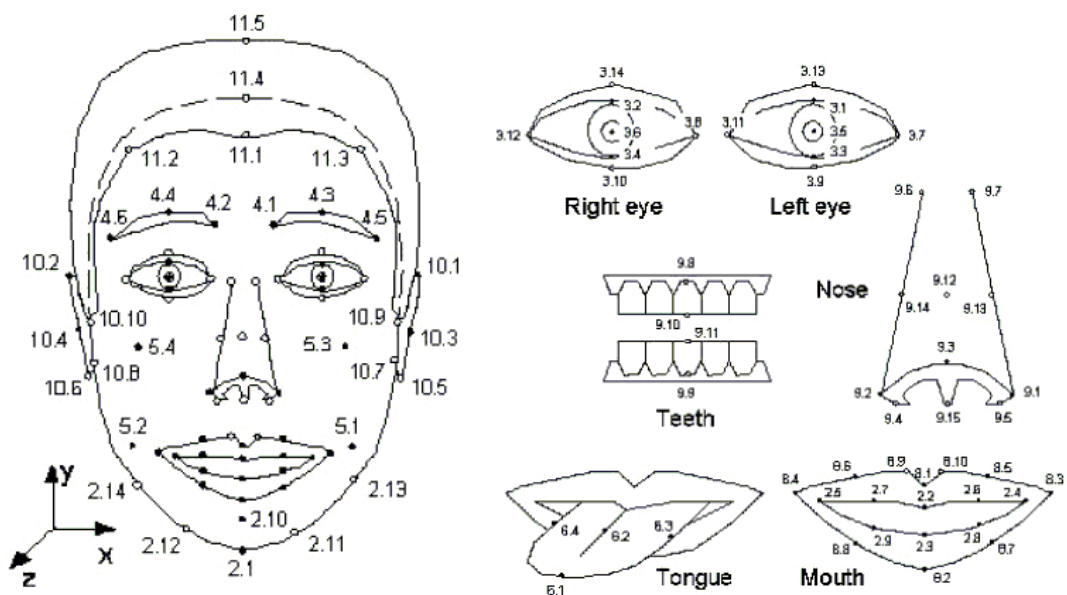
Az MPEG (Moving Picture Expert Group) szabványok legfőbb célja a hang és videó jelek tömörítése. A tömörítés lényege, hogy minél kevesebb információ veszteséggel tudjunk kódolni adatokat a tárolás és adatátvitel hatékonyságának növelése érdekében. Az MPEG-4 alapjait az 1990-es évek végén fektették le, amivel egy még ma is bővülő, igen szerteágazó szabványt hoztak létre fejlesztőknek, szolgáltatóknak és végfelhasználóknak kialakítva. A kódolás célja az alacsonyabb sávszélességen történő audio-video átvitel. Az MPEG-4-et nem csak nagyobb mértékű tömörítés érdekében fejlesztették ki, hanem figyelembe vettek olyan multimédia alkalmazásokat is, mint 3D-s animációk, szintetizált hangok, képek, szövegek külön vagy akár együttes kezelése és élethű megjelenítése [1].

Az MPEG-4 szabvány számunkra legérdekesebb része a fej és az emberi test megjelenítésére és mozgására (FBA - Face and Body Animation) szolgáló rész. Az FBA-ra vonatkozó szabványrészlet leírja az arc és a test alakjának és mozgásának kódolási alapelveit. Az FBA megvalósításakor egy bonyolult és igen összetett problémával találták szemben magukat a szabványkészítők: ahány emberi fej, annyi

féle karakter, annyi féle jellegzetes arc. Az hamar kiderült, hogy ennek a szabványnak sokszínűnek, képlékenynek kell lennie, hisz semmiféle fix paraméterezéssel nem lehet két egymástól különböző emberi arcot leírni. Ebből a felismerésből következett a kódolás-dekódolás problémára talált válasz: a többi MPEG szabványtól eltérően itt nem adják meg a kódolási és a dekódolási eljárást, nincs minden szigorú keretek közé szorítva annak érdekében, hogy megmaradjon az alkotók (akár művészek) szabadsága. Csak a küldött adat formáját és értelmezését határozza meg az MPEG-4 FA (Facial Animation).

3.1.FP – Feature Point

Ahány emberi fej, annyi féle karakter, annyi féle arc és ebből adódóan annyi féle paraméterezés. Minden kutatócsoport másképp csinálta. Ezen az állapoton kellett a szabványnak segíteni, hisz így az egymástól külön dolgozó csapatok meg sem tudták egymással osztani a kutatási eredményeiket, mivel nem voltak kompatibilisek a fejmodelljeik. A szabványkészítők a közös jellemzőket vették alapul.



ábra 7: Az MPEG-4 által definiált FP-k

Az MPEG-4 FA 86 darab Feature Point-ot (FP-t) definiál az arcon (7. ábra), amelyek segítségével le lehet írni a különböző arcmodelleket. Ezeknek a pontoknak az átfogó neve a Face Definition Parameter (FDP), azaz arcdefiníciós paraméter. Az összes MPEG-4 szabvánnyal kompatibilis modellen fel kell venni ezeket a feature point-okat. Az így definiált modell a semleges (neutrális) arckifejezést jeleníti meg, az arcot alap

állapotában fogja mutatni. Az arc normál állapotáról akkor beszélünk, ha nem tükröz semmilyen arckifejezést, ha nem tükröz semmilyen érzelmet (emotion), ha a szája csukott alapállásban van (nem látszanak sem a modell fogai, sem a nyelve), előre néz a tekintetével és nem pislog. A számítógép számára is érthető nyelven leírva:

- A koordináta rendszer jobbsodrású,
- A fej tengelyei párhuzamosak a világ tengelyeivel,
- Az összes izom az arcon ernyedtt, alapállapotban van,
- A szemhéjak érintik az íriszt,
- A pupilla átmérője az íriszének egyharmada,
- A tekintet z tengely irányában van,
- Az ajkak összeérnek, a belső ajak vonala vízszintes, és egy magasságban van a külső ajakszélekkel,
- A száj zárva van és a felső fogsor érinti az alsót,
- A nyelv lapos és vízszintesen helyezkedik el a szájban, a nyelvcsúcs az összezárt felső és alsó fogsort érinti.

Az FP az MPEG-4-nek az a paramétere, aminek a segítségével le tudunk írni bármilyen arcot. Az FP-kbe van kódolva a különböző arcok jellegzetességei. Ebből már látható, hogy ha ez minden MPEG-4 kompatibilis arcon megtalálható (márpedig ez következik a kompatibilitásból), akkor az animálást is ennek a segítségével kell megvalósítani, hisz ezek a pontok ha nem is azonos helyen találhatók a fejmodelleken, de mindenképpen azonos területeket írnak le. Az animáció az FP-keket mozgatja és így az FP-k vonzáskörébe tartozó részeket is.

Az FP-keket az animálás alapján két részre lehet osztani: közvetlenül mozgatott tartópontokra és más FP-k által mozgatott tartópontokra. Az előbbi pontok halmaza külön-külön mozgatható, tehát nem csak arcdefiniáló szerepe van, hanem animáló is. Ilyen pontok például a szájon elhelyezkedő FP-k. Az utóbbi pontok csak definiálják a fej alakját, de közvetlen mozgásra nem képesek. Mozgásuk más tartópontok mozgásától függ. Ide tartoznak például az orrnyergén, homlokon stb. elhelyezkedő FP-k. [20]

Arra a kérdésre azonban még nem kaptunk választ, hogy mi alapján mozognak a tartópontok. A választ a FAP-okban kell keresnünk.

3.2.FAP – Facial Animation Parameter

Az arcanimáló paraméterek határozzák meg az FP-k mozgását. A FAP-ok igen közel állnak az arc anatómiai felépítéséhez, elhelyezkedésük alapját az arc izomzata adta a szabványkészítőknek.

A szabvány 68 FAP-ot különböztet meg, feladatuk, hogy egy vagy több FP-t mozgassanak. Az első két FAP magas szintű, ez azt jelenti, hogy ezekkel előre beállított komplexebb mozgást lehet kivitelezni.

Az első FAP egy megadott vizéma (vizéma: a fonéma látható, képi megfelelője [8]) szerinti megjelenést határoz meg. Az angol fonémakészletnek megfelelő vizémákat állítják elő. Használatakor két vizémának a kódját kell megadni. Ebből a két vizémából tud kevert vizémákat előállítani. Nulla súly esetén az első, míg egyes súly esetén a második vizémát eredményezi. Köztes súlyok köztes vizéma átmeneteket képeznek. A második FAP a hat alap érzelem megjelenítésére szolgál, amelyeknek a keverésével az összes többi érzelem is kifejezhető. [18] Ez a hat alap érzelem a következő: öröm, bánat, harag, félelem, undor és meglepetés.

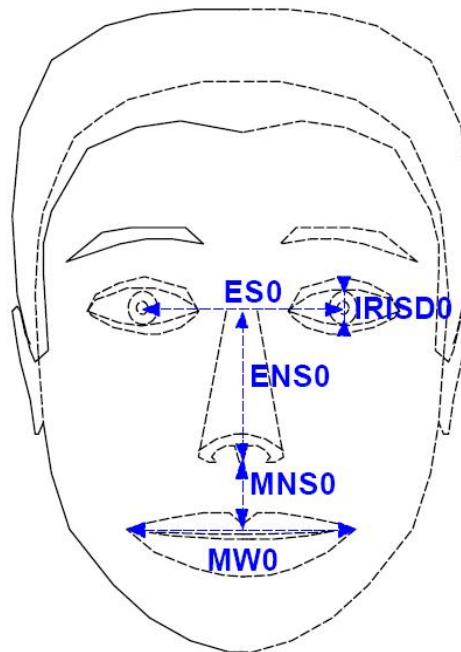
A többi 66 FAP alacsony szintű, mely 10 csoportba van sorolva az alapján (7.ábra), hogy az arc mely részegységét mozgatja. Ezek abban különböznek a magas szintű FAP-októl, hogy itt a mozgás irányát és amplitúdóját kell megadni, nem pedig egy összetett feladatra előre összeszerkesztett mozgásvezérlést kell kezdeményezni.

Az alacsony szintű FAP-ok általában egy-egy tartópontot mozgatnak. Előfordulhat azonban olyan mozgás is, amely az összes FP-re hatással van, de közvetlenül csak egy pontot mozgatnak, ilyen például a fej forgatása. Az alacsony szintű FAP-ok prioritása magasabb, mint a magas szintű paramétereké. [1] De hogyan valósul meg a mozgás? Az egyértelmű, hogy a FAP nem reprezentálhat egy koordináta értéket, hiszen minden modell máshol és máshogy helyezkedik el a saját világában, nem is beszélve a méretbeli különbségekről. Akkor a FAP reprezentáljon valamilyen elmozdulást. De mihez képest? Az origóhoz, vagy a modell egy adott pontjához? Mivel mind az origó, mind egy „adott” pont nem elég konkrét, visszavezették a problémát az FP-khez. Hiszen ezek a modellt a normál állapotában reprezentálják. Ezért a FAP minden esetben a normál állapottól való eltérést adja meg, nem pedig egy előző állapothoz képest. Ám még mindig problémát okoznak a méretbeli különbségek. Egy nagyobb és egy kisebb fejű modellt még mindig nem lehet egységesen animálni. Ezért találták ki a FAPU-t.

3.3.FAPU - Facial Animation Parameter Unit

A FAPU, az arcanimáló paraméter egység határozza meg a fentebb leírt mozgások mértékegységét és felállít egy olyan mértékrendszert aminek a segítségével egymáshoz lehet igazítani a különböző animált modelleket (méretek, arányok összeegyeztetése). De hogyan lehetséges ez az összeegyeztetés?

Megítélésem szerint ez a Face and Body Animation leglényegesebb, legérdekesebb megoldása. Egy leleményes trükkel sikerült azt megoldani, hogy minden modell FAPU-ja különbözhessen, azaz még ezen a ponton se kelljen túl szigorúan megszabni a modellek paraméterezését. Ugyanis a FAPU-k az arc jellegzetes pontjainak távolságából származtathatók (8. ábra). Ezek a távolságok például a szemek egymástól vett távolsága, a csukott száj szélessége, vagy az orr hosszúsága.



ábra 8: A FAPU-k

Tehát ha a FAPU-k minden modellre egyediek, akkor a dekódolási folyamat során figyelembe kell ezt venni. Minden FP mozgatásához a FAP-okban meg van határozva, hogy a hat FAPU közül melyikből származtatható az elmozdulása. Az elmozdulás egysége a jellegzetes távolságok 1024-ed része. A hat FAPU:

- $IRISD = IRISD0 / 1024$: a szembogár átmérője

- $ES=ES0/1024$: a szemek távolsága
- $ENS=ENS0/1024$: az orr hossza
- $MNS=MNS0/1024$: az orr és a száj közötti távolság
- $MW=MW0/1024$: a száj szélessége
- $AU=10^{-5} \cdot \text{rad}$: az elfordulás egysége radiánban.

3.4.XML

Már csak arra a kérdésre kerestük a választ, hogy hogyan helyezzük el a vrml modelleken az MPEG-4 FP-eket? Kutatásaink során találtunk az Interneten egy ingyenes, szabadon letölthető 3D-s talking head-et, az Xface-t (15. ábra) [7]. Ez az alkalmazás egy Európai Unió kutatás keretén belül született a trentoi egyetem Kognitív és Kommunikációs Technológia Karán (The Cognitive and Communication Technologies – TCC – Division) [6]. Az Xface egy MPEG-4 szabványos fejmodell, amely egy általános adatszerkezetet, az XML-t használja a modell objektumainak a tárolására. Külön fájlban tárolja a fejet, a szemeket, a fogakat, stb. és az objektumokhoz tartozó FDP-eket és FAPU-kat. Ez a módszer nekünk tökéletesen megfelelt, ennek a programnak a segítségével tudtunk eligazodni a .vrml fájlakon belül, ennek segítségével tudtuk a vertex csoportokat azonosítani.

4. A konzultációk

Azután, hogy megismerkedtünk ezzel a szabvánnyal, ami megkerülhetetlen ebben a témában hogy ha olyan alkalmazást akarunk létrehozni, amit könnyű lesz a jövőben továbbfejleszteni vagy más rendszerekhez hozzáilleszteni, és mielőtt rátérnék az általunk használt fejmodellre, hadd mutassam be az ezzel párhuzamosan futó és legalább ilyen fontos munkát, a siketekkel való együttműködést. Nagyon fontos megismernünk az élethelyzetüket, az elvárásaikat, az igényeiket, meghallgatnunk a konstruktív ötleteiket.

Kiemelném a személyes kapcsolt fontosságát, hisz ezek az emberek zárkózottak és nem szeretnek a problémájukról beszélni. De egy rövid bemutatkozó periódus után, könnyen kialakult a csapatmunka, és az együttműködés is zökkenőmentesen működött. Rengeteg kutatást végeztünk nagyothalló és siket fiatalok bevonásával,

ami elengedhetetlen, ha a projektünkben megfogalmazott célokat meg akarjuk valósítani. A munkánk folyamán sokat foglalkoztunk, konzultáltunk velük, hogy beelássunk az ő világukba, hisz egy átlagember nem is sejti, nem is sejtheti mennyi problémával küszködnek, még a legegyszerűbb szituációkban is (bevásárláskor, kérdezősködéskor). Meg kellett tanulnunk mik a nehézségeik a mindennapi életben; mit értenek, mit nem; mennyire tudnak szájról olvasni, mik a zavaró körülmények. Ahhoz hogy ezeket feltérképezzük sok, általunk készített tesztet csináltattunk meg velük.

Első lépésben különböző fejmodelleket mutattunk nekik azzal a céllal, hogy kiderítsük, az Interneten fellelhető modellek mennyire élethűek, mennyire lehet a szájukról leolvasni a beszédet. Az interjúk során elmondták, hogy ezek a rendszerek nagyon jó grafikával rendelkeznek, igen élethűek, de alkalmatlanok a szájról olvasásra. Azt is sikerült kiderítenünk, hogy nem mindegy számukra, hogy egy teljesen mozdulatlan, merev fejmodell szájmozgását próbálják leolvasni, vagy egy olyanét, aminek van egy véletlenszerű egyszerű mozgása, pl. bizonyos időközönként pislog, vagy enyhén valamelyik oldalra billenti a fejét. Ezek az apró mozgások nem szolgálnak semmiféle többlet információval a beszédre, egyszerűen csak arról van szó, hogy természetesebbnek hat az animált fejmodell, és nem vonja el – még ha tudat alatt is – a figyelmüket a beszédre a modell arcának kifejezőtelensége.

Többlet információt a beszélő gesztikulálása vagy arcjátéka szolgálna, hisz a siketek nem hallják, a nekünk, hallóknak is sok mindent eláruló hangsúlyokat, hanglejtéseket. Vannak ilyen irányú kutatások is, de a projektnek ebben a fázisában nem vehettük figyelembe ezeket a tényezőket [17]. Tehát ezek a modellek nem feleltek meg az elvárásaiknak.

Egy szintén nagyon fontos teszt alkalmából saját magunk által készített videó felvételeket mutattunk nekik. Első lépésben projektorral kivetítve, majd utána korszerű mobiltelefonokat adtunk körbe, ugyanazokkal a felvételekkel. Ebben a kísérletben vizsgáltuk többek között, hogy elég nagy felbontással rendelkeznek-e az általunk használt mobiltelefonok; hogy mennyire számít a háttér színe, megvilágítása; hogy szemből, vagy valamilyen szöggel elforgatva készítsük el a felvételeket; kipróbáltuk a beszélő megvilágításának beállításait. A kiértékelés után sok lényeges dolog derült ki számunkra: nagyon fontos a helyes artikuláció, a száj jó láthatósága, a száj és környékének jó megvilágítása; a bajusz, vagy szakáll az zavaró lehet számukra, sokat számít a beszélő személy mimikája és egyéb gesztikulációja. A szöveggörnyezet is döntő fontosságú a kommunikációjuk

szempontjából. A gyors témaváltásokat, a váratlan szituációkat, a visszakérdezéseket nem szeretik, mert nem tudják követni a beszélgetés fonalát.

Nagy figyelmet szenteltünk a beszéd vizuális jellemzőinek elemzésére.

5. Vizuális jellemzők

Elsősorban azt kellett kiderítenünk hogy mitől lesz egy beszélő animált fej élethű, és mely jellemzők akadályozzák a siketeket és nagyothallókat a szájról olvasott szöveg helyes felismerésében.

Azzal mindenki tisztában van, hogy a beszéd érthetőségét javítja, ha látjuk a beszélő személy arcát és artikulációját. A hangképzésben az önálló mozgásra képes, aktív beszédszerveknek – elsősorban az ajkak, a nyelv, a fogak, a lágyszájpad és a nyelvcsap – van kitüntetett szerepe. Ebből a felsorolásból is rögtön látszik, hogy a jellemzők nagy része nem látható a siketek számára. A látható rész, általában az ajkak, még számunkra, jól hallóknak is nagy segítséget nyújtanak, sok többlet információt hordoznak, főleg zajos környezetben, vagy suttogáskor. Nagyothallók és siketek esetében ez még fontosabb. Számukra a mesterséges vagy természetes beszédből nem a hallás közvetít információt, hanem az artikuláció, illetve annak vizuális jellemzői. Arra is fény derült a siketekkel folytatott megbeszéléseink alkalmával, hogy mennyire fontos számukra a nyelv és a fogak állása (ha láthatók).

A vizuális jellemzők vizsgálatára azért van nagy szükség, mert csak ezek ismeretében lehet élethű modellt alkotni, ami megfelelő segítséget nyújt a nem hallók számára.

Nem látható különbségek például a zöngéesség, a nyelvcsap állása, a nyelv elhelyezkedése a szájüregben, a szájüreg alakja, a hang kiejtésének időtartama, a hang erőssége. Ugyanennek a projektnek a keretein belül folynak olyan kutatások is az egyetemünkön, amelyek pl.: a zöngéesség „láthatóvá tételével” foglalkoznak. Egy vibrotaktilis kijelzőn dolgoznak a kollégáim, ami a következő elven működik: a hangjel elemzéséből kinyert nem látható információt egy rezgésen alapuló kijelző segítségével jelenítik meg. Ha megoldható ennek az alkalmazásnak az összekapcsolása a mobiltelefonokkal, akkor a siketek egy olyan készülékhez juthatnak, amellyel szájról olvasva tudnak kommunikálni és pl.: a zöngés-zöngétlen (baba-papa) hangpárok, vagy a nyelvhelyzettel meghatározott felpattanó zárhangok között is különbséget tudnak tenni.

A számunkra legfontosabb vizuális jellemzők a következők [8]:

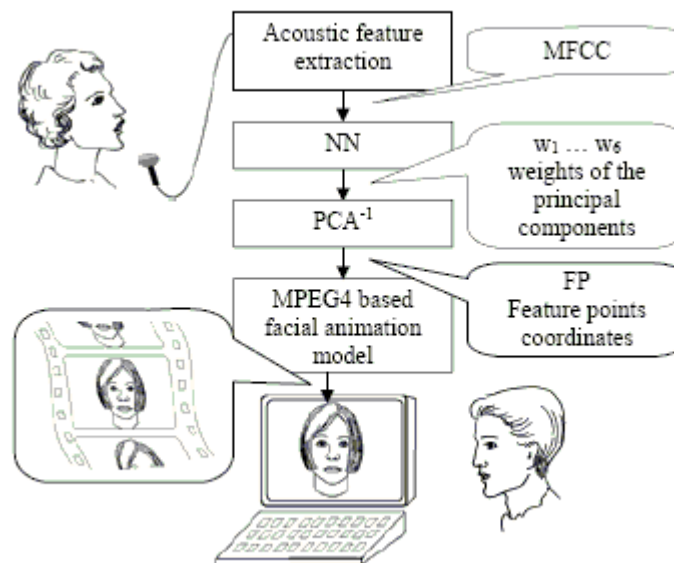
- Az ajkak nyitva vannak-e? Az ajaknyílásnak az aránya eltérő ajakkerekítéses és ajakréses hangoknál. Az ajkak előre mozgása az ajakkerekítéses hangokra jellemző, pl.: csücsörítés. A legtöbb fejmodell ezt nem tudja jól ábrázolni. Az áll mozgása is összefüggésben van a száj nyitottságával és a fogak láthatóságával, hiszen az alsó fogsor az állal együtt mozog. Ez triviálisnak tűnik, de az első fejmodelljeinknél (lásd az alábbiakban: Lucia) nem voltak egymáshoz kötve ezeknek a paramétereknek a mozgása, némelyik modell le tudta húzni az állát, úgy, hogy a fogsora zárva maradt.
- Az ajkak területe. Attól függ, mennyire vannak összeszorítva az ajkak. Például: a felpattanó zárhangok, vagy nazális hangok esetén.
- Az ajkak nyílásának vízszintes és függőleges irányú mérete. Például: ajkkerekítés, csücsörítés (o, ö, u),
- Látszanak-e a fogak? Erre a problémára készült már jól működő automatizált eljárás az egyetemünkön. [20]
- Látszik-e a nyelv? Az előbb említett program képfeldolgozási eljárásokkal a nyelv láthatóságát is meg tudja állapítani.

A legfontosabb jellemzők az ajkak és a nyelv működtetéséhez tartoznak. A száj szélessége az ajaknyitással és az ajkkerekítéssel, illetve az ajakréssel, áll összefüggésben. A nyelvállást a nyelv függőleges helyzete, vízszintes mozgása, hajlítása, és a nyelvhegy formája befolyásolják.

A szakirodalomból és a megfigyeléseinken alapuló eredményeket mindig megmutattuk a siket segítőinknek, hogy véleményükkel, észrevételeikkel pontosítsák és kiegészítsék azokat.

6. A PPKE fejanimáló rendszere

A PPKE-n kifejlesztett rendszer sokban különbözik a más kutatók, vagy egyetemek által kifejlesztett rendszerektől (9. ábra) [9]. A mi rendszerünk lényegében egy text-to-animation rendszer, vagy még inkább egy voice-to-animation rendszernek felel meg. Első lépésben a saját audiovizuális adatbázisunkat állítottuk össze, jeltolmácsok segítségével. Ezután a beszédjelet, feldolgozás után egy neurális hálózat bemeneteire adjuk. A kimeneten pedig megjelenik az MPEG4 szabványos FAP-ok által mozgatott animált fej.



ábra 9: A saját rendszerünk felépítése

6.1. Az audiovizuális adatbázis felvétele:

Sokat gondolkodtunk azon, hogyan is csináljuk ezeket a felvételeket előrelátóan, úgy hogy ne kelljen majd újra és újra fölvenni. Ez elég nehéz feladatnak tűnt, és mint utóbb kiderült nem is sikerült nekünk.

6.1.1. A bemondók

Készítettünk sok próbafelvételt a projektben résztvevőkkel és évfolyamtársaink bevonásával. Ezek egyszerű felvételek voltak még, többek között azt teszteltük velük, hogy:

- bárki lehet-e bemondó, vagy szükséges ehhez valamilyen képzettség,
- hogyan állítsuk be a kamerákat, hogy a lehető legjobb felvételeket tudjuk készíteni,
- milyen háttér előtt és milyen megvilágítással dolgozzunk,
- elegendő-e csak a szájat és környékét felvenni, vagy szükség van az egész, jól látható fejre.

A sikertől kapott visszajelzések egyértelműen azt mutatták, hogy az egyetem hallgatóival készített felvételek nem elég jók és nem elég olvashatók, mert nem artikulálnak a szavak számára megfelelően a szöveget felmondók [10] [16]. Tehát jeltolmácsot kellett felkérnünk, hogy legyen segítségünk a felvételek elkészítésében, hisz ők az idejük nagy részében siketek és nagyothallók

társaságában élik az életüket, így már teljesen alkalmazkodott a beszédük, sokkal szebben artikulálnak, mint mi, jóhalló emberek.



ábra 10: Egy elhomályosított bemondó

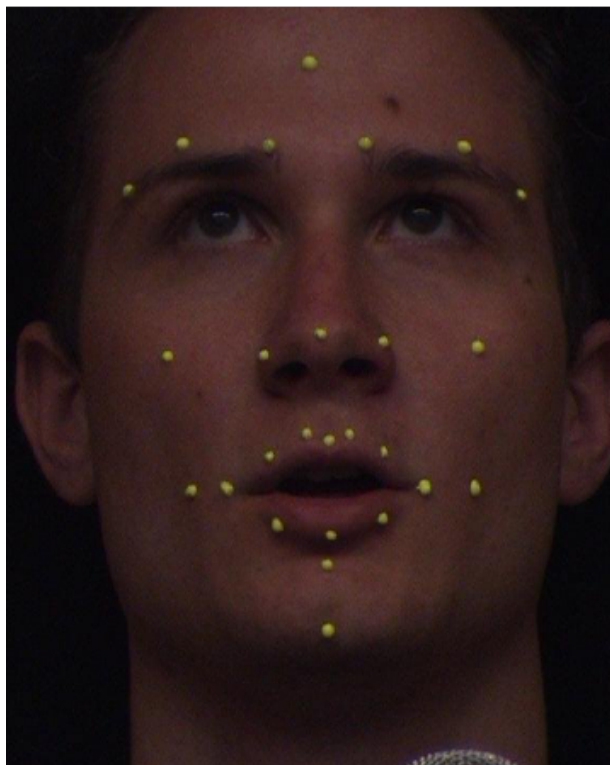
Az is kiderült, hogy az nem elég, hogyha csak a szájat vesszük föl és kitakarjuk, elhomályosítjuk (angolul: blur) a körülötte lévő részeket (10. ábra) ugyanis nagyon fontos a siket ember számára a mimikából, az érzelmek kifejeződéséből, vagy a modalitástól függő (nemcsak a hangsúllyal, hanem gesztusokkal is kifejezzük egy mondat modalitását, pl.: felhúzzuk a szemöldökünket kérdő mondat esetén) arckifejezésekből kapott többlet információ. Hogy mennyire fontos az egész fej, azt a következő kísérlet szemlélteti igazán jól. Csináltunk olyan teszteket beszélő fejmodellekkel, hogy megmutattuk ugyanazt a modellt kétszer, egyszer teljesen mozdulatlan volt, máskor pedig egy apró random mozgást állítottunk be a modellnél, beszéd közben. Az a fejmodell, amely véletlenszerű apró mozgásokat végzett a beszéd közben, sokkal élethűbbnek hatott a siketeknek, (mivel nem vonja el a figyelmüket a modell természetellenes mozdulatlansága és merevsége) és sokkal jobban le tudták olvasni a szájáról a beszédet, mint arról a fejmodellről, ami teljesen mozdulatlan volt, a száját kivéve. Ebből is látszik, hogyha csak egy szájat animálnánk, az természetellenesnek hatna és nem tudnák olyan jól értelmezni, mint egy egész fejes modellt. Még az is kiderült a felméréseinkből, hogy az arcszőrzet is zavaró lehet a számukra, hisz a szakáll, vagy bajusz viselete miatt, nehezebben követhető a száj mozgása és ezáltal az artikuláció.

6.1.2.A példamondatok

A felvételek előtt volt még egy nagyon fontos és sok körültekintést igénylő feladat, mégpedig a példamondatok összeállítása. Úgy gondoltuk, hogy nem elég csak példamondatokat felmondadni a jeltolmácsokkal, ezért diádokat is olvastattunk velük a mondatok után, úgy hogy azok a lehetséges fonéma együttállásokat a lehető legjobban lefedjék. A bemondóknak 60 darab példamondatot és kb. 200 diádot kellett felolvasniuk, melyek a magyar fonémakészletet a nyelv átlagos eloszlásával használták.

6.1.3.A felvételek

A bemondóknak az arcát pontokkal kellett megjelölni, hogy később a felvételeken követhetők legyenek. Ezért az MPEG4 szabványban leírtaknak megfelelő pontokat rajzoltunk rikító sárga színű – semmiféle mérgező anyagot nem tartalmazó – festékkel a jeltolmács arcára. Továbbra is sokat konzultálva a siketekkel, olyan FDP pontokat vettünk fel az arcon, elsősorban a száj környékén, ami számukra fontos a szájról olvasás szempontjából, és ezek mellé néhány, a beszéd alatt általában fix, nem mozgatott referencia pontot (pl.: a homlokon, az orron és az állcsúcson), ami nekünk segítség. Az utómunkában például a fej elforgatásakor segíthetnek ezek a pontok a visszatranszformálásban (11 ábra). Azért választottuk a sárga színt, hogy könnyen tudjuk követni a pontokat a felvételeken.



ábra 11: A felpöttyözött bemondónk

A későbbiekben erre készült is egy MatLab-ban írt program, egy hallgatótársam jóvoltából, ami a videójelből automatizálja a pontok detektálását néhány képfeldolgozási eljárással és utána azok követését.

A videó felvételeket egyszerű, a nagyobb, elektronikai cikkek árusításával foglalkozó boltokban is kapható digitális videokamerákkal vettük fel. Ezek a kamerák PAL szabványúak, 720x576 képpont felbontásúak és mi a szabványos másodpercenkénti 25 képkocka mintavételezést alkalmaztuk a felvételek készítésekor.

A hangot külön mikrofonnal vettük fel egy erősítőn átvezetve, ugyanis a kamerák beépített mikrofonjaival nem lehetett elég jó minőségű felvételeket készíteni.

6.2.Neurális Hálózat

Az így létrehozott audiovizuális adatbázist használhatjuk a neurális hálózatunk tanítására. Más ilyen irányú kutatások felépítésével szemben, a mi tanítóhalmazunk nem a magunk soraiból vett bemondók szájmozgását tartalmazta, hanem ahogy már többször említettem, jeltolmácsokét, mert az ő beszédük alkalmazkodott a siketek elvárásaihoz, mind a beszédtempó, mind az artikuláció tekintetében. Ráadásul több jeltolmáccsal is felmondattuk ugyanazt a szöveget, hogy a neurális hálózatunknak még tökéletesebb tanulóhalmazt tudjunk biztosítani. Az általunk készített Neurális hálózat 100 000 epochon keresztül tanult. [9] Ez tömörítés nélkül több hónapig is eltartott volna, mi viszont főkomponens analízis (PCA) segítségével tömörítettük az FP-k helyzetét. Így nagyobb hatásfokot tudtunk elérni, és az információ veszteségünk nem haladja meg a 3%-ot, ami 1-2 pixel eltérést jelenthet egy FP helyzetére nézve, ami elfogadhatónak számít.

6.3.A Mobilkészüléken megjelenő végeredmény

Tehát a neurális hálózat kimenetén a FAP-ok jelennek meg, a PCA algoritmus szerint tömörítve. A mobiltelefon kijelzőjén pedig a visszaállított FAP-okat a fejmodellünk kirajzolja, amit a siketek reményeink szerint könnyen leolvashatnak. Az általunk használt mobilkészülékek kijelzője elegendő, hiszen a felbontása DPI-ben a modern monitorokkal egy kategóriába esik [11] [12]. Ez a célnak megfelel, elegendő a szájról olvasáshoz, amint azt a tesztjeink is igazolták.

7. Az általunk használt fejmodell:

A rengeteg új ismeretünknek köszönhetően és sok a megszerzett tapasztalattal, új szemszögből kezdtünk el létező fejmodellek után kutatni. Nagy szerencsénkre találtunk is egy ingyenes, az Internetről szabadon letölthető, nyílt forráskódú, jól működő fejmodellt: a Luciát.

7.1.A Lucia

Az előző félékben többek között azt a feladatot kaptam, hogy ismerjem meg a Lucia program felépítését. A legfőbb nehézséget az jelentette számomra, hogy ugyan a program nyílt forráskódú, semmiféle leírást nem találtam hozzá a felépítéséről, sem a működéséről, így azt nekünk kellett visszafejtenünk. Továbbá, az is kiderült elég hamar, hogy valószínűleg több ember írhatta, mert jelöléseiben, jelrendszerében nem következetes. A programkódban, az eligazodást segítő kommentek csak nagy ritkán lelhetők fel, ha mégis találunk, akkor viszont olasz nyelven, ami nem meglepő, hisz egy olasz professzor csoportja írta (a padovai Institute of Cognitive Sciences and Technologies kutatói írták Cosi Professzor irányítása alatt) [14], a gond csak a megértésükkel volt. De abba is hamar bele lehetett rázódni.

Lucia egy OpenGL grafikai rendszerrel rendelkező MPEG-4 szabványos fejmodell (12. ábra). Tehát minden elvárásunknak eleget tesz. Kiváló grafikával rendelkezik, mind a modell, mind a textúra jól kidolgozott, nagyon részletes. A program indításakor az előre gyártott VRML fájlból állítja elő a modell ponthálóját. A pontháló sok vertexből áll, főleg a fontosabb részeknél, mint pl. a száj, és környéke, ezáltal szépen lekerekített felületeket, gazdag részleteket láthatunk. A fej különböző részeit tetszés szerint el is lehet rejtetni pl.: szemek, fogak, haj, stb. A textúrákat külön képfájlokban tárolja, indításkor azokból olvassa ki és helyezi el a ponthálón. A modellt körbe tudjuk forgatni három dimenzióban, illetve a zoomolásra is van lehetőség.

7.2.Lucia mozgatása

A paraméterek két osztálya írja le a modellt és annak mozgását. A FAP-ok és az FDP-k. Az FDP a modell alakját, míg a FAP a modell mozgását határozza meg, ahogy az az MPEG-4 fejezetből kiderül. Minden FAP köré definiálva van egy hatókör, és azokat a vertexeket tudja mozgatni, amelyek az ehhez tartozó területen belül helyezkednek el. A hatókör szélén elhelyezkedő pontok kevésbé veszik át a FAP

mozgását, mint a közeliek. Ez a mozgáselhalás nem lineáris a távolság függvényében, hanem koszinuszos.



ábra 12: Lucia modell

A mozgás egy TempFile.fap nevű vezérlőfájlban van leírva. A fájl első sora, a header azt kódolja, hogy hány képkockát jelenít majd meg másodpercenként a modell, illetve, hogy összesen hány frame-et. A továbbiakban minden képkockához két sor tartozik. Az első sor egy logikai sor, csupa 0-ákból és 1-esekből áll össze, amelyeknek a segítségével meghatározza a mozgatandó paramétereket.

A második sorban az első sorban kijelölt paraméterek értéket kapnak, tehát ez a mozgásra szolgál. Első eleme az aktuális frame sorszáma. A sor elemei egész számok lehetnek. Ha nulla értéket kapnak, akkor a paraméterek a normál, természetes arckifejezés helyére állnak vissza, azaz az alapértelmezett értékbe. A fájl feltérképezése során kiderült, hogy maximum 68 paraméter változtatható.

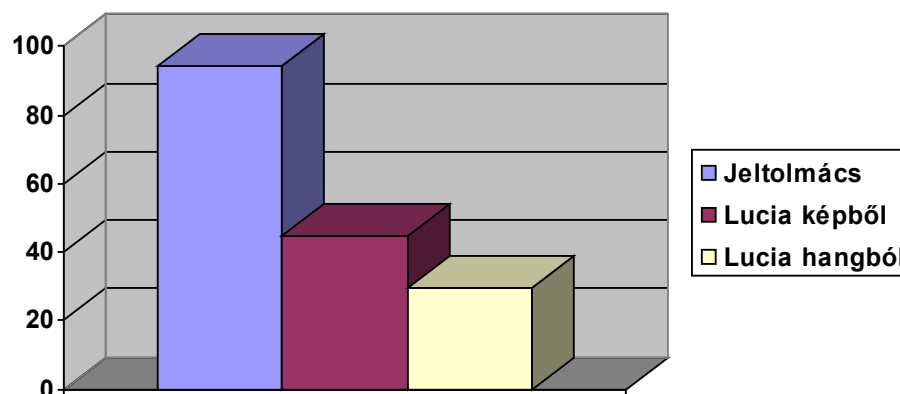
A 68 paraméter lényegében valamely FAP bizonyos irányú elmozdulása vagy elforgása. Ezek elmozdulásának mértékét elvileg az MPEG-4 szabványban szereplő FAPU-k határozzák meg. Sajnos a program forrásában erre utalóan következetlenséget találtunk, mert az egyes paraméterek mozgása egy-egy konstans

számmal van skálázva, és nem a modell ponthálójából származtatják. Ettől függetlenül ez jól működhet Luciára, de más modellre nem.

A kód visszafejtése után és néhány módosítással ellátva a program mozgató függvényeit, észlelhető javulás állt be Lucia szájmozgásában. Azonban azt le kellett ellenőriznünk, hogy vajon jó úton haladunk-e és hogy ez a program megfelel-e a projektünk céljának. Újra össze kellett ülni a siket segítő társainkkal és ismét tesztet írni velük.

A teszten három különböző módú beszéd érthetőségét mértük fel. Az első egy videófelvétel volt, amin a jeltolmács beszéde volt látható. A második fajta videón Lucia beszélt, akit a jeltolmácsról rögzített videóból származtatott FAP-ok mozgattak. A harmadikon is Lucia szerepelt, de szájmozgását a saját rendszerünkkel generáltuk, a hanggal tanított neuronháló kimenetéből számolta az elmozdulásokat. Összesen 70 darab videót mutattunk, melyből véletlenszerű sorrendben, de egyenlő eloszlással volt mind a három fajtaból. A videókat hang nélkül játszottuk le.

Az 13. ábra 13 olyan siket találati arányát mutatja, akik nem először vettek részt a tesztjeinken vagy beszélgetéseinken. Az öt leggyakorlottabb siket esetén a kép alapján származtatott Luciát 50%-ban, míg a hang alapján beszélő Luciát 38,7%-ban olvasták le helyesen. A felvételek között mindössze három olyan volt, amit senki sem talált el. [10]



ábra 13: A felismerés aránya

8. Saját fejanimáló rendszerem

Ha nem vesszük figyelembe a jeltolmács által mondott szövegnek a felismerési arányát, akkor azt tapasztaljuk, hogy nincs szignifikáns különbség a két utóbbi teszt eredménye között. Tehát a siketek, a Lucia által kétféle képpen számolt szájmozgást közel azonos találati aránnyal ismerték fel. A két tanítási módszer azonban teljesen különböző alapokon nyugszik, így arra lehetett következtetni, hogy nagy valószínűséggel Lucia modellnek a megjelenítésével van a probléma.

A program amit írtam lényegében egy fejmodell modellezése. Azért van erre szükség, hogy lássuk milyen irányba haladjunk a Lucia fejlesztésével. A kísérleteinkből, amiket a siketekkel végeztünk az derült ki, hogy a mi modellünk beszédét kicsit több mint 50%-os arányban értik meg (ennyit tudnak a szájáról leolvasni). Ezen eredmények után további kísérleteket kellett elvégezni és kérdőíveket kitölteni, hogy megtaláljuk azokat a sarkalatos pontokat, amelyeknek a módosításával javulást érhetünk el a szájról olvasás arányában. Ennek a fontosságát már bemutattam a vizuális jellemzők című fejezetben.

Amit már eddig is sejtettünk, az beigazolódni látszott. A Lucia számunkra ideális kezdőmodellnek bizonyult, és nem is várhatunk el többet egy fejmodelltől sem, de számos hibája van, amit orvosolni kell, ha a projektünket sikerre akarjuk vinni. Például sok esetben nincsenek a modellben jól meghatározva az egyes FP-knek a hatósugarai, vagy egy pont túl erőteljes mozgásakor a textúra nem tudja követni és valahol „szétszakad”. Ugyanez következhet be, ha a pontoknak az egymásra hatása nincsen jól kezelve, ha egy pontot egyszerre több FAP is szeretne rángatni (14. ábra).

Az is kiderül a tapasztalatainkból és a felméréseink is azt igazolják, hogy a Lucia mozgása, az igen sok alkotópont és az ebből következő rengeteg számítás miatt lassúnak és néhol darabosnak, szögletesnek mondható.

Ha még a számítógépen futó alkalmazásnak sem áll össze a mozgása egyenletessé, akkor mi lesz majd a mobilos alkalmazással? Ami egy leegyszerűsített változat lesz ugyan, de a mobiltelefon processzorának a kapacitása is véges.

Ezeknek a hibáknak a felderítését és a modell hiányosságainak a pótlását tűztem ki célul magamnak. Az előző félévekben a Lucia kódját próbáltam módosítani, úgy hogy a mozgása finomabbá váljon, de egy idő után be kellett látnom, túl nagy fába vágtam a fejszém. Maga a program olyan hosszú (60 ezer sor körül van a terjedelme), hogy túl sok időt vett igénybe a kibogozása, míg az ezzel járó fejlődés arányaiban nem volt

elég látványos. Többször megtörtént, hogy valamilyen ismeretlen paraméter felderítésébe kezdtem és az állítgatás következtében elromlott valami a Luciában. Viszont olyan is előfordult, hogy egy, már ismert paraméterrel kísérleteztem és egészen máshol romlott el valami a kirajzolás során. Van még egy nagy hibája a kódnak, mégpedig az, hogy sok helyen nem következetes, hisz valószínűleg több fejlesztő dolgozott rajta.



ábra 14: A helyes textúra, majd néhány frame múlva, a szétszakadt

Ezek az indokok vezettek arra a következtetésre, hogy egy sokkal egyszerűbb, az eddig megismert fejmodellektől teljesen független modellt hozzak létre, amin jól lehet demonstrálni azokat a sarkalatos pontokat, amiktől jelentős javulást várunk. Ha megvan ez a modell és működnek az általunk javasolt változtatások, akkor már nincs más dolgunk, mint ezeket a módosításokat beépíteni a Lucia modellbe.

Nem azt vizsgálom a kísérleteim folyamán, hogy egymáshoz képest hogyan mozognak a pontok, hiszen a jeltolmácsokkal készített videókból a tökéletes mozgás a birtokunkban van. Arra szeretnék matematikai összefüggést találni, hogy hogyan tudjuk mi magunk beállítani a pontoknak a paramétereit, illetve azok peremfeltételeit, hatóköreit, hogy a lehető legjobban lehessen a szájat modellezni a siketek és nagyothallók számára.

A projekt kivitelezéséhez az inkrementális programozás metodikáját választottam, azaz a váz elkészítése, és egy alap feladatokat ellátó alkalmazás prezentálása után időről időre újabb tulajdonságokat és képességeket adtam hozzá a meglévő programhoz.

Fejlesztő környezetnek a Microsoft Visual C++ 6.0 -t választottam, segítségül pedig az OpenGL API-t használtam fel és annak a GLUT könyvtárát is.

Mielőtt az általam fejlesztett programra rátérnék, adok egy rövid összefoglalást az OpenGL-ről és hogy miért pont ezt a grafikai megoldást választottam.

8.1.Az OpenGL:

Az OpenGL egy nagyon jól használható, számunkra igen hasznos térgrafikai alkalmazásprogramozási felület. 1992-ben a Silicon Graphics nevű cégnél fejlesztették ki. Az elnevezés az Open Graphics Library -ből adódik, és maga az API a Specification szabványleírásban látott napvilágot. Az OpenGL-t használó programok ezt a szabványt alkalmazzák a síkbeli és a térbeli geometriai alakzatok, adatok, képek megjelenítésére. [13] [18]

Ez egy igen hatékony API, lehetővé teszi a valós idejű leképezést, egyszerű a használata és széles körben támogatott (platformfüggetlen). Az OpenGL-ről összességében elmondható, hogy számos hardverarchitektúrán fut, és a különböző megvalósításai jól dokumentáltak. Tartozik hozzá egy szabványos alkalmazás-csomag, a GLU, amely az OpenGL Utilities-nek a rövidítése és számos előre megírt magasabb rendű alkalmazást, rutint, függvényt tartalmaz, a programozás egyszerűbbé és gyorsabbá tételéért. Ugyanígy a GLUT könyvtárát is használtam a munkám folyamán, amely az ablakozásban, a bemeneti (billentyűzet, egér), vagy felhasználói események kezelésében volt segítségemre, ugyanis az OpenGL nem tartalmazza ezeknek a támogatását, mert általában a számítógép rendszerekben ezek platformfüggően vannak megoldva. A GLUT könyvtár pedig ezeket a szolgáltatásokat platformfüggetlenül támogatja, így a mi céljainknak tökéletesen megfelel.

Még egy hatásos érv, hogy miért ezt használtam: Lucia is OpenGL-ben van megírva.

8.2.Egyedülálló pont 2D-s statikus megfigyelése

Első lépésként felvettem egy NxN -es síkot, feltöltöttem pontokkal. Ezen a síkon egy tetszőlegesen felvett pontot vizsgáltam két dimenzióban. A ponthoz rendeltem hatósugarat is, így tekinthető egy feature point (FP) leegyszerűsített, sematikus modelljének. Ennek a pontnak nem kell feltétlenül a sík pontrácsából származnia, de akár származhat onnan is, a szabványban nem található ilyen irányú megkötés. A

pontnak úgy kell működnie, hogy megfeleljen az MPEG4 szabványban foglaltakkal. Amennyiben a pontot elmozdítjuk az eredeti helyéről, a hozzá rendelt hatókör összes rácspontjának a helye módosul az elmozdulás függvényében.

Végeztünk olyan kísérleteket a siket segítő társainkkal, amelyben azt vizsgáltuk, vajon szükség van-e egy valódi három dimenziós modellre, vagy elég ha az arcot egy síkban ábrázoljuk, azaz két dimenzióban? Mutattunk nekik háromdimenziós beszélő fejet és mellette olyat aminek a harmadik, 'z' koordinátáit csupa nullával töltöttük fel, így csinálva belőle kétdimenziósat. Arra alapoztuk ezt a kísérletet, hogy csak nagyon kevés vizémánál fordul elő, hogy a harmadik dimenzióknak jelentősége van. Ezt a siketek is alátámasztották, mondván, ők úgylis csak szemtől szembe tudnak szájról olvasni, tehát számukra nem, vagy csak igen kismértékű többletinformációt tartalmazna, ha nem kétdimenziós modellt használnánk.



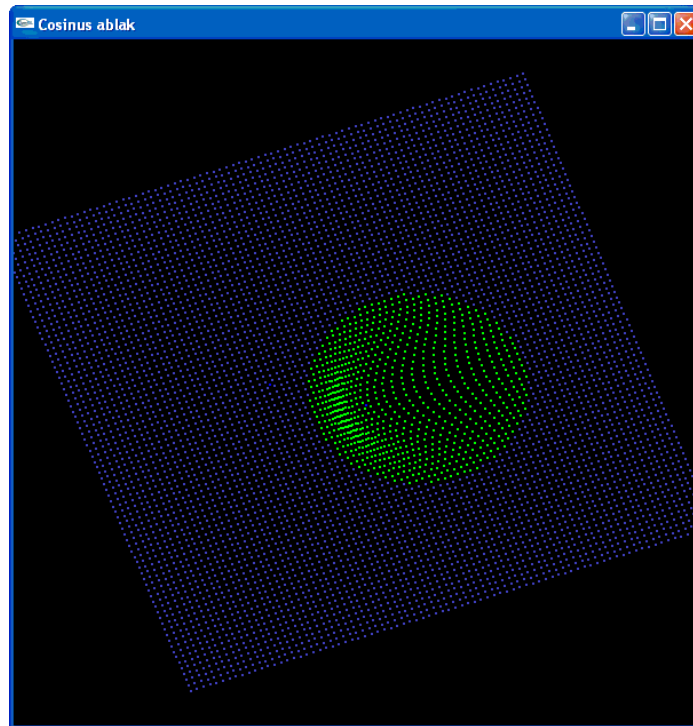
ábra 15: Alice az 'i' hang kiejtésekor és undorral az arcán.

Ezért végeztem az első kísérletet síkban. A sík és az FP megjelenítéséhez egy GUI (Graphical User Interface) ablakot készítettem a programomhoz, az OpenGL GLUT könyvtárának a segítségével.

A kétdimenziós statikus esetben azt kell vizsgálni, hogy hogyan viselkednek az FP hatósugarán belül elhelyezkedő pontok. Ha a síkban egy irányba mozgatom a pontot, akkor a hatókör egyik felén besűrűsödnek a sík pontjai, míg az ellenkező oldalon megritkulnak. A beszédben ezt a jelenséget, tehát az arc pontjainak a

sűrűsödését akkor (pl. az arcbőr felgyűrődése;) figyelhetjük meg, amikor az egyes érzelmeket fejezzük ki (pl.: a haragos ember felhúzza a felső ajkát), vagy az illabiális (ajakréses) magánhangzók, pl.: á, e, é, i, í kiejtésekor. (15. ábra).

A 16. ábrán látható az arcbőr felgyűrődése egy FP vízszintes irányú mozgásának a hatására.



ábra 16: A mozgás hatása a hatókör pontjaira

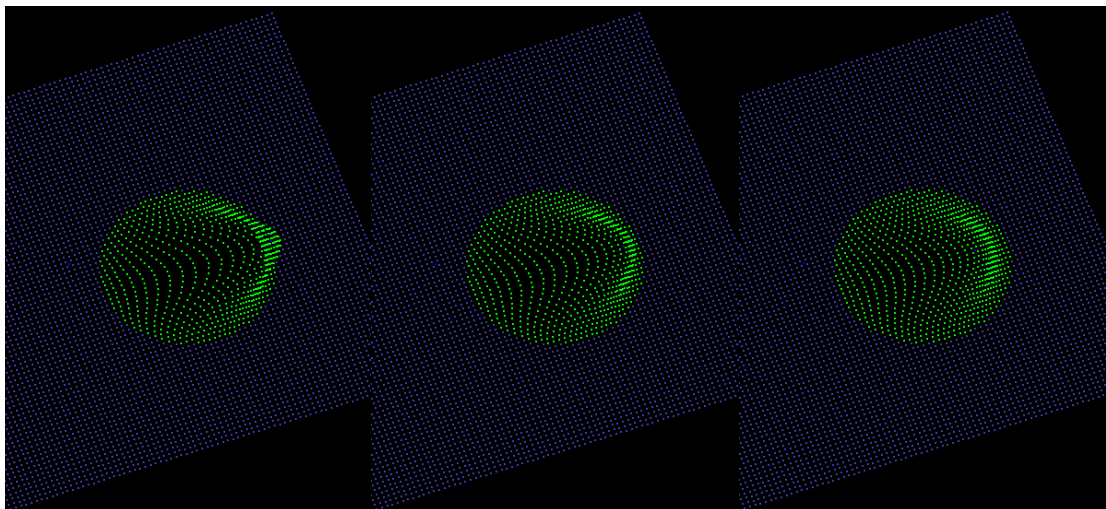
Ez a Luciában sok helyütt nincs jól kezelve, nincsenek a peremfeltételek helyesen meghatározva. Például sok esetben egy FP-t a hatósugarán kívülre is el lehet mozgatni, aminek nincs semmi értelme (valószínűleg csak figyelmetlenségből maradt benne a Lucia kódjában), viszont teljesen eltorzítja az arcot, széthúzza a textúrát, vagy szét is szakítja azt.

A projekt előrehaladtával hamar világossá vált, hogy a Lucia modell minden egyes FP-jét és a hozzájuk tartozó hatókörökön belüli elmozdulásuknak a határértékeit ellenőrizni kell. A kísérleteim alapján arra a következtetésre jutottam, hogy nem elegendő, ha megállási feltételnek a hatókör sugarát adom meg (peremfeltételt 100%-ra állítom; 17. ábra) az egyes FP-k elmozdulásakor.

Ennél erősebb feltételt kell megszabni, mert különben túlságosan torzul a pont hatóköre, mozgatkor a kör széléhez közelítve az FP-t. Ez a torzulás a modell textúrájában egy igen szembetűnő és természetellenes gyűrődést eredményezne az arcon. Ha a hatókör sugarának a 80 százalékát vettem megállási feltételnek, még az

is kevésnek bizonyult. Ekkor ugyan nem gyűri ki az FP a hatókörön kívülre a szélén elhelyezkedő pontokat, tehát a hatókör körvonala nem torzul, de még mindig túl nagy lesz a pontok sűrűsödése a kör egyik szélénél, míg túlságosan széthúzódnak a pontok a hatókör másik szélénél. (17. ábra)

Ezért választottam az elmozdulás peremfeltételének a hatósugár 60 százalékát. Természetesen lehetett volna még tovább finomítani a dolgon, hisz a 60 és 80 százalék között nagy különbség van, de ahhoz túlságosan sok időt kellett volna befektetni és a várható eredmény nem hozott volna akkora fejlődést, hogy ez a többlet munka kifizetődő legyen.

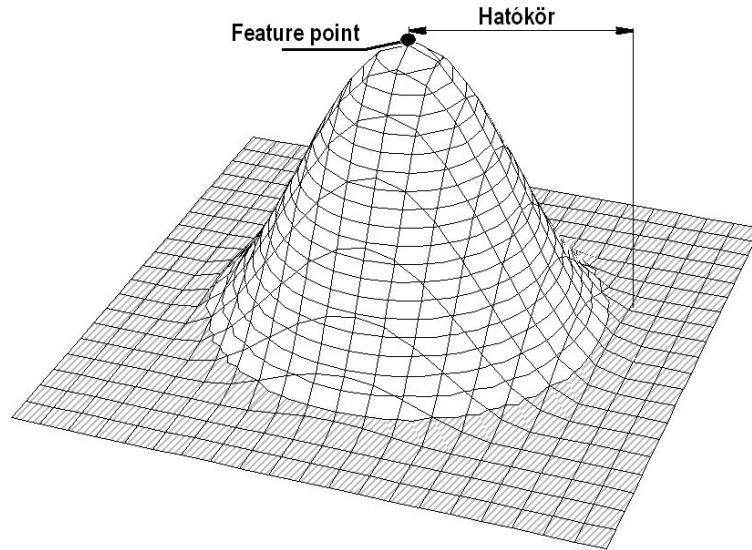


ábra 17: FP elmozdulása 100%-os, 80%-os, 60%-os peremfeltétel mellett

8.3.Két, vagy több pont 3D-s statikus megfigyelése

A program továbbfejlesztésében a következő lépés az volt, hogy az eddigi egy pont mellé felvettem még kettő tetszőleges pontot a síkon és megoldottam, hogy az FP-eket egy vectorban tárolja a program, így a későbbiekben könnyebben lesz kezelhető ezeknek a pontoknak a halmaza. Ezenkívül kibővítettem az általam írt FDP osztályt és így már az osztályhoz tartozó pontok és az azok hatósugarában elhelyezkedő pontok háromdimenziósak lettek, míg a sík egyéb pontjai maradtak kétdimenziósak.

Két vagy annál több pont esetén a legfontosabb jelenség amit vizsgálni kell, az ezeknek a pontoknak az egymásra hatása. Mivel a fenti esetben olyannak választottam meg a síkban mozgó FP-nek a peremfeltételeit, hogy annak mozgatása során ne deformálja a hatókörének a területét (hisz a szabvány alapján, egy FP csak a saját hatókörén belüli vertexekre hat (18. ábra)), csak abban az esetben van értelme két pont egymásra hatásáról beszélni, ha azok hatókörei valamilyen mértékben fedik egymást.



ábra 18: FP és a hatókörön belül elhelyezkedő pontokra kifejtett hatása

A Lucia modellben a távolság arányában egy bizonyos súlytényezőt számolnak, ugyanis e nélkül a hatókör szélén elhelyezkedő pontok a szomszédoktól igen eltérő mozgáshoz vezetnek volna, ami a modell még szakadozottabb mozgáshoz vezetett volna. Viszont nem lehet a távolság arányában a súlytényezőt egy lineáris függvény szerint változtatni, hisz az hasonló szakadozott animációhoz vezetett volna. Éppen ezért az FP-k súlytényezői a következő képlet szerint alakulnak:

$$w_{ij} = \frac{1 - \cos \left(\left(1 - \frac{s_i}{r_j} \right) * \pi \right)}{2} \quad i=1 \dots n, \quad j=1 \dots 86$$

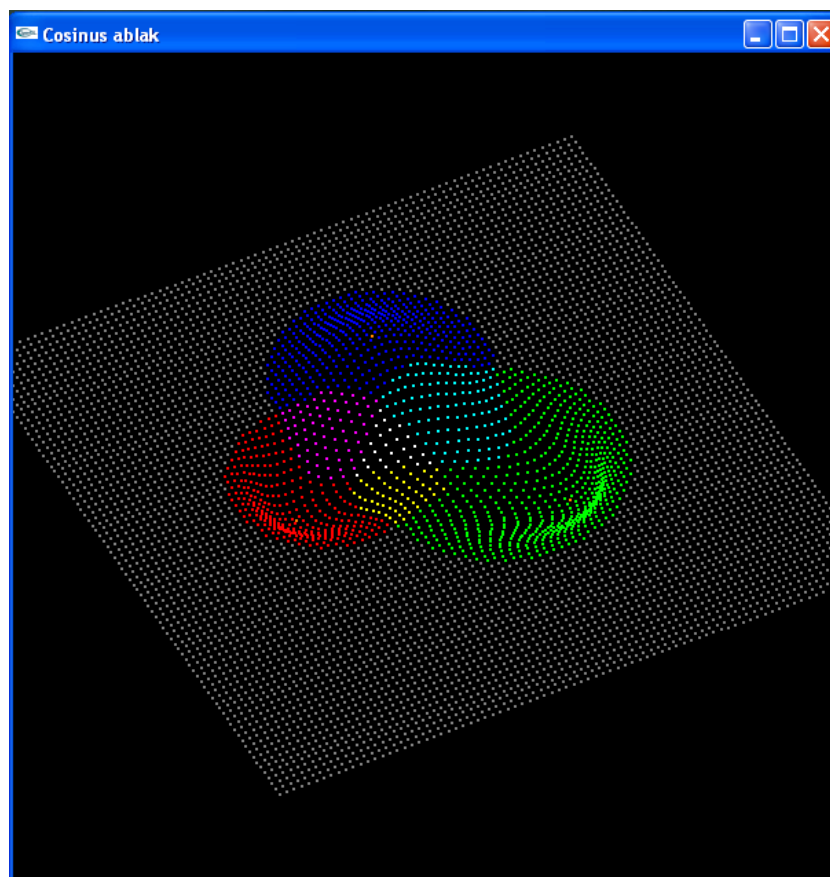
Ahol w_{ij} a j-edik FP által mozgatott, a hatókörben az i-edik vertex súlytényezője, s_i az FP és annak hatókörében az i-edik vertex Descartes távolsága, r_j pedig a j-edik FP hatókörének sugara.

Az s_i , vagyis az FP-től vett távolság háromdimenziós esetben a következőképpen alakul:

$$s_i = \sqrt{(x_{FDP} - x_i)^2 + (y_{FDP} - y_i)^2 + (z_{FDP} - z_i)^2}$$

Ennek a koszinusos megközelítésnek az a hátránya, hogy némely esetben nem működik megfelelően pl.: a szájnál és annak környékén, de erre majd a későbbiekben térek ki bővebben.

A nem a száj környékén lévő FP-kre ez a képlet helyesen működik, úgyhogy mi sem tértünk el ettől, mi is ezt használtuk a projektünkben. Két vagy három FP modellezésére tökéletesen alkalmas, így én is ezzel a képlettel vizsgáltam a programomban a pontok egymásra hatását. Ha a súlytényezőket eszerint állítjuk be, az azt eredményezi, hogy az FP mozgásakor, a hatókörön belül elhelyezkedő pontokra egyre kevésbé hat az FP, a távolság arányában. Így a hatókör széle felé kis elmozdulást kapunk, míg az FP tözsomszédságában az elmozdulás közel azonos az FP elmozdulásával.

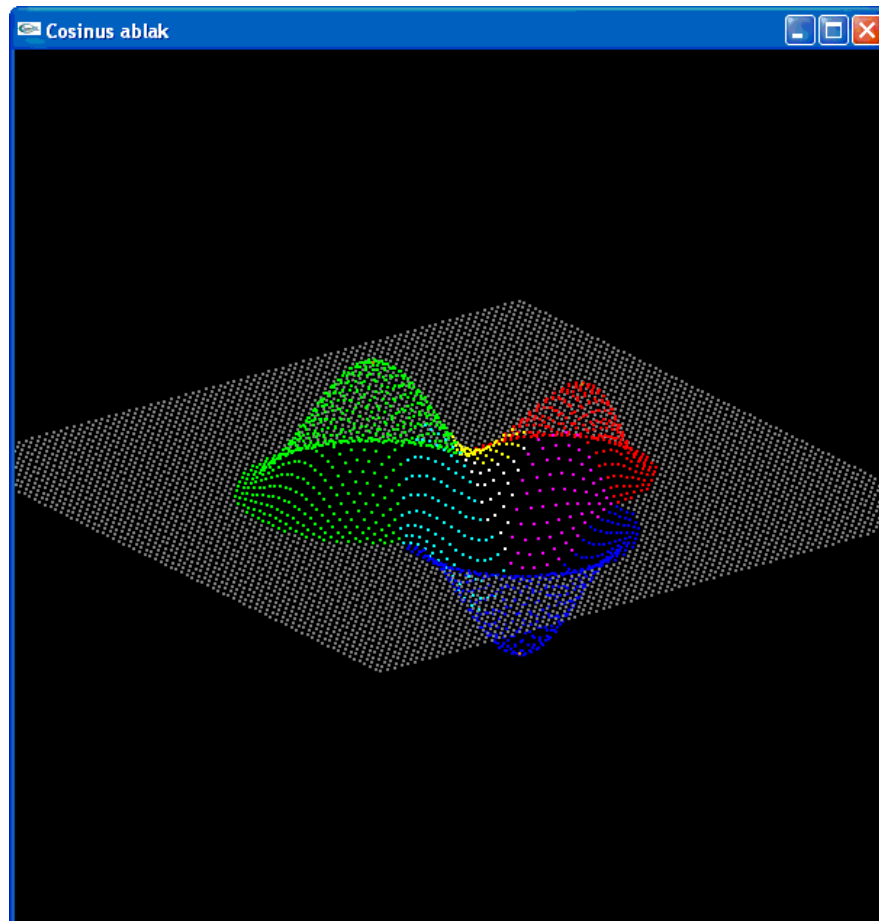


ábra 19: 3 FP pont mozgása a síkon, különböző irányokba.

Ezzel a technikával kiküszöbölhetők a hatókör kerülete mentén keletkező szakadások, ugyanis itt, a határon elhelyezkedő pontoknak már nem lesz nagy az elmozdulásuk, mint ahogy egy lineáris képlet alkalmazásával lett volna. Amikor a módosítást végeztem a programon belül az FDP osztályon, a jobb átláthatóság kedvéért megoldottam, hogy ha az FP-k különböző színű hatókörei fednék egymást, akkor a közös területnek a színe az eredeti két szín keverékével egyezzen. A színek segítségével könnyen lekövethető, hogy hogyan alakulnak az FP hatókörén belül elhelyezkedő pontok a mozgás során (6. ábra). Láthatjuk, hogyan deformálódik a három „erő” hatására a hatóköröknek a metszete, és az is megfigyelhető, hogy sehol

sincs szakadás a pontok között, illetve a pontrács módosulásában sem fedezhető fel semmilyen törés, mindenhol szép egyenletes átmeneteket találunk.

Egyedül arra kell figyelni ebben az esetben is, mint ahogy azt korábban már tárgyaltam, hogy az FP-k hatóköreinek peremfeltételeiről gondoskodjunk. Amit az előző pontban bevezettem, az ebben az esetben is megállja a helyét, nem kell a 60 %-os határon módosítani.



ábra 20: FP-k mozgása a térben

A program továbbfejlesztésében, a következő lépésben a harmadik dimenziót vezettem be (7. ábra). Az FP-k és a hatóköreiken belül elhelyezkedő pontok lettek háromdimenziósak, míg a sík többi pontját, amelyeket nem mozgatok a kísérletezés folyamán, nem változtattam meg, maradtak kétdimenziósak.

A harmadik dimenzió bevezetésének több oka is volt. Az első, hogy így a modellemmel általánosabb feladatokat is képes vagyok megoldani. A második, hogy ha nagy ritkán is, de némely vizéma megértését sokkal könnyebbé teszi a siketek számára a térbeli kép. Ezt egyébként ügyes és jól átgondolt árnyékolással is ki lehetne váltani a mozgatott FP-k körül, ez azonban nem része az én feladatomnak.

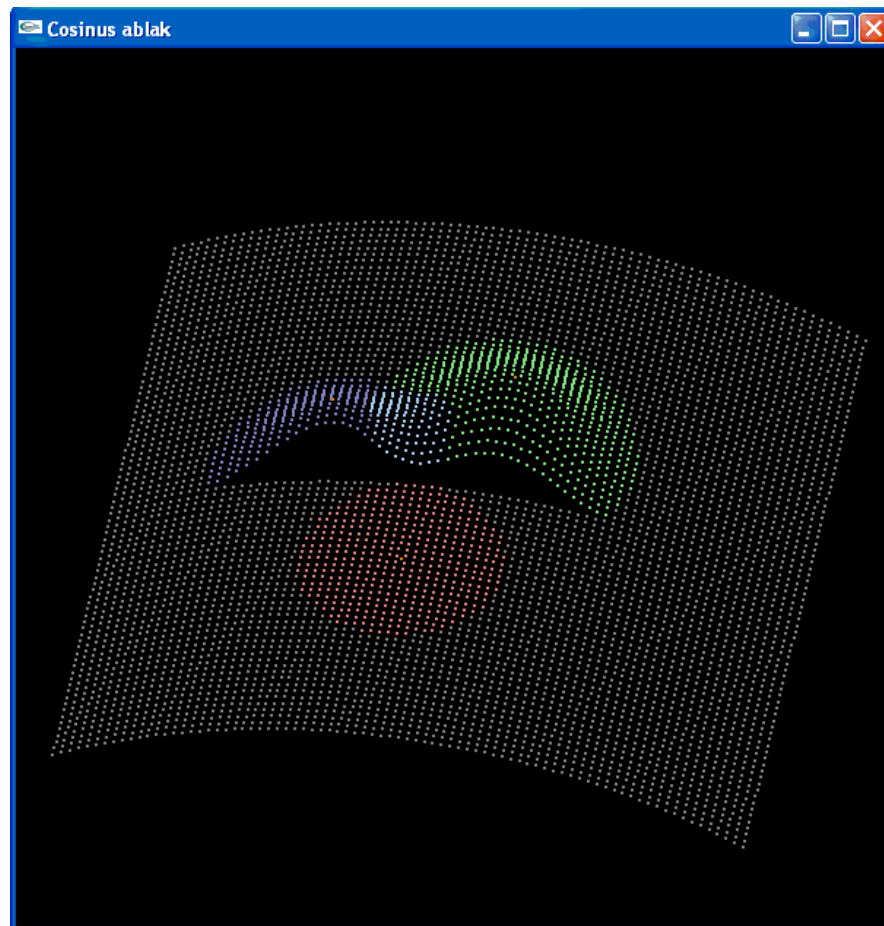
És hogy mi jelent a siketek számára segítséget? Elsősorban azoknak a hangoknak a kiejtésére gondolok, amelyeknél levegővel telítődik, felfújódik az arcunk, mint például némely zárhang esetén (b - p), vagy amikor csücsörítünk. Ebben a két esetben vannak olyan pontok, amelyek kiemelkednek az arc síkjából, ezért van szükség a térbeli ábrázolásra. Ezen a ponton még a felvételeinken is módosítani kellett, ugyanis kezdetben mi nem festettük meg a bemondóink arcán az 5.3 és 5.4 pontokat (MPEG4 ábra), amelyek az arc felfújódását jelzik, hanem csak a szájra koncentráltunk. A siketektől származó visszajelzések azonban egybehangzóan kimutatták felvételeink hiányosságát. A felvételek módosításából adódik, hogy a neurális hálózatot is újra kellett tanítani.

A háromdimenziós pontok esetén nem engedhető meg, hogy a 'z' tengely mentén az elmozdulás túl nagy mértékű legyen, mert akkor a hatókör határa mentén túl durva lesz az átmenet, ami a textúra megtöréséhez vezethet. Ezért ebben az esetben is, a már bevált 60 %-os peremfeltételt állítottam be, így az elmozdított FP és a maga után vont hatókör egy lekerekített kúp alakzatot hoz létre.

Végül még egy fontos fejlesztést hajtottam végre a programomban, amihez szintén a Glut könyvtárat használtam. Megvalósítottam azt, hogy a síkot mind az egér, mind a cursor billentyűk segítségével forgatni lehessen három dimenzióban és beépítettem a nagyítás és kicsinyítés funkciókat is. Ezek az új opciók, amelyeknek a segítségével körbe lehet járni a síkot, meg lehet minden irányból vizsgálni az elmozdított FP-ket, be lehet nézni a sík alá, zoomolásra is van lehetőség, nagyban segítettek az immáron háromdimenziós modell elemzését a kísérleteim során.

8.4. Több FP statikus vizsgálata egy szakadással, a meghajlított síkon

Ebben a továbbfejlesztési szakaszban két dolgot valósítottam meg a programomban. Az első, hogy meghajlítottam a síkot, így az úgy néz ki mintha egy hengerből lenne egy részlet. Ezáltal kicsit jobban hasonlít az arc sematikus ábrázolásához. A másik, hogy vízszintes irányban csináltam egy szakadást a síkon, ami a szájat hivatott modellezni az én leegyszerűsített rendszeremben.



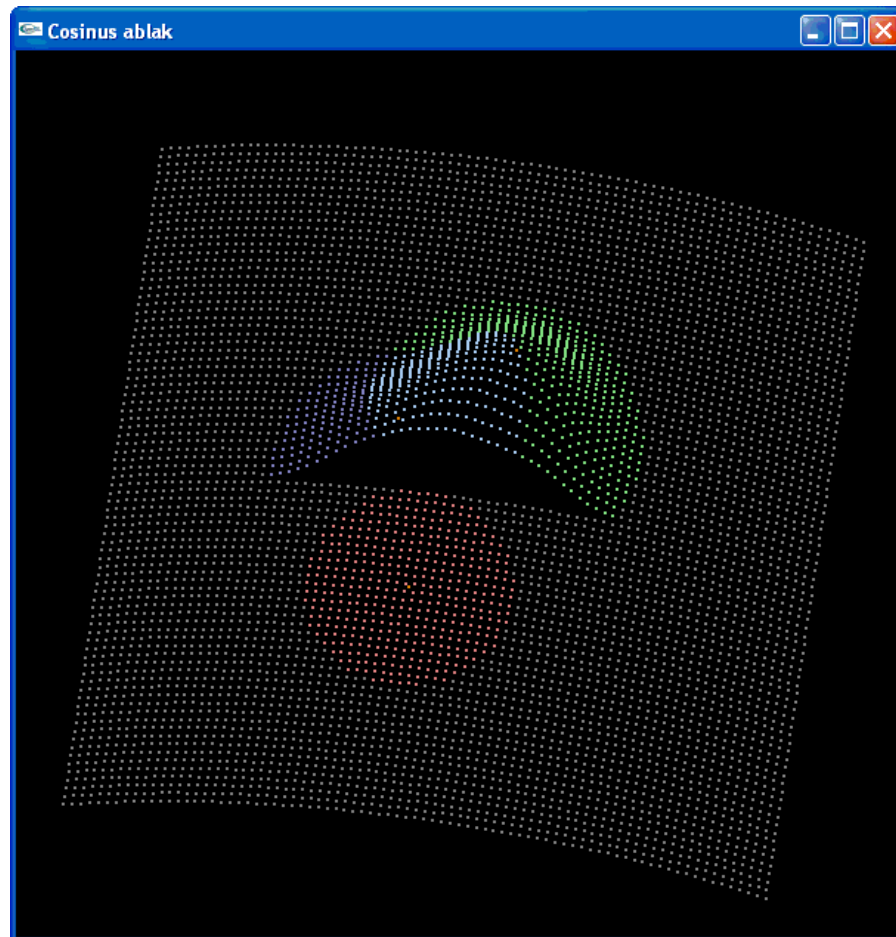
ábra 21: Nyitott szájnak a modellezése. A hajlított síkon mozgatott FP-k hatására kialakuló felső ajknak hullámvonal a rajzolata

Az ajkak környékén nem működhet jól az előző fejezetben bemutatott koszinusos képlet, mert nem veszi azt figyelembe, hogy a száj miatt van egy vízszintesen végigfutó szakadás a textúrában. Egy az alsó ajkon elhelyezkedő FP, a mozgatáskor nem húzhatja magával a felső ajkon lévő FP-t, még akkor sem ha az a hatókörén belül helyezkedik el (és a száj környékén szép számmal találunk ilyen eseteket, az összes 2-es és 8-as családba tartozó FP ilyen (lásd 7. ábra). Itt nem alkalmas pusztán Descartes távolságot számolni, ugyanis az alsó és felső ajkak vertexeit teljesen külön kell mozgatni. Ezt a problémát a Lucia készítői úgy küszöbölték ki egy elég kezdetleges megoldással, hogy ezeken a területeken nem hatókört alkalmaznak, hanem felsorolják az FP által mozgatott vertexeket és egyszerűen kihagyják a nem oda tartozókat.

A „szájat” én úgy valósítottam meg, hogy a sík pontjait, egy általam kijelölt szakaszon megdupláztam és a környéken elhelyezkedő FP-k, hatóköreinek (ha átlógott a száj vonalának a túloldalára) túlnyúló részét levágtam. Tehát megmaradt a hatókörnek minimum a fele – nem úgy mint a Luciánál alkalmazott megoldásnál –, és így

kikerülhető az az igencsak fáradságos és időigényes procedúra, hogy külön-külön, manuálisan kell megadni az összes ilyen kényes helyen szereplő FP-nek, hogy mely vertexeket és milyen súlytényezőkkel mozgasson, és melyeket ne.

Tehát a helyenként rosszul működő koszinusos képlet ismét alkalmazhatóvá vált és könnyen tudtam a saját modelletem használni a száj körül fellépő megjelenítési problémák vizsgálatára.

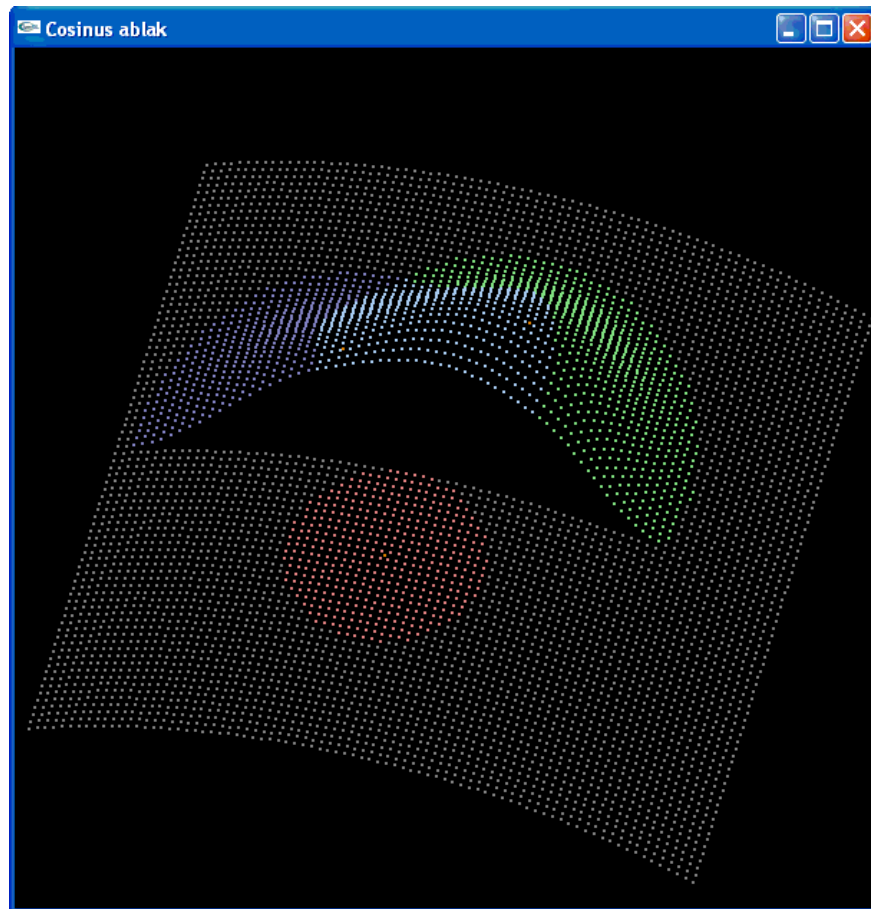


ábra 22: : A közelebb vitt FP-k

Azonban azt vettem észre, hogy az általam véletlenszerűen felvett pontok nem alkalmasak a száj modellezésére. Ugyanis, ha a két FP-vel megpróbálom kinyitni a száját, a képletben szereplő koszinus függvény hatására a két pont között, a hatókörük átfedésénél, ahol mindkét FP-nek már kisebbek a súlytényezői, a 'száj' hullámvonalban nyílik ki. Ezáltal nem hasonlít egy szájra. (21. ábra)

Három dolgot próbáltam ki ennek a problémának a kiküszöbölésére. Az első megoldási javaslatom, hogy az FP-ket közelebb vittem egymáshoz. Ezt én a saját modelletemen megtehetem, de egy MPEG-4 szabványos fejmodellel, pl.: a Lucián, ez a módszer nem alkalmazható. Ha közelebb viszem a pontokat, akkor a hatókörökön

belül úgy alakulnak a súlyok, hogy a kirajzolásnál száz pontjai egy íven helyezkednek el (22. ábra).



ábra 23: A megnagyobbított hatókörök

A másik megoldás, ami az előzővel szinte egyenértékű, hogy a hatóköröket növelem meg, akkorára, hogy azok nagyobb mértékben fedjék egymást. Így közelebb kerül egy hatókörnek a széle egy másik hatókörnek a középpontjához, azaz az FP-hez, ahol erősebb súlyok hatnak az egyes pontokra (23. ábra). Ez a megoldás látszólag nem sokban különbözik az előzőtől, mégis nagy az eltérés, hiszen ezt „legálisan” használhatom a szabványos fejmodelleken is, ugyanis a szabványban csak az FDP-k pontos helyei vannak rögzítve, a hatókörök nagysága nem, így azok szabadon variálhatóak.

Ennél a pontnál tartom fontosnak megjegyezni azt a tényt, hogy egy „hatókör” nem csak kör alakú lehet, hanem más alakzatot is felvehet, mint pl.: ellipszis, vagy egy manuálisan körbehatárolt tetszőleges alakzat, így elkerülhető a méretváltozás, ami az én modellemben bekövetkezett. Például egy tojás alakú, lapos 'hatókör' segítségével megvalósítható egy olyan elrendezés, ami a pozitív 'x' irányba

megnöveli a hatókört, de az 'x' tengely mentén negatív irányban és az 'y' tengely mentén változatlanul hagyja azt.

A harmadik megoldás, hogy ebben az esetben egy másik képletet alkalmazunk és akkor nincs szükség sem az FP elmozdítására a helyükről, sem nagyobb hatókört rendelni hozzájuk. Az erre vonatkozó szakirodalom áttanulmányozása után azonban maradtam a jól bevált koszinuszos képletnél.

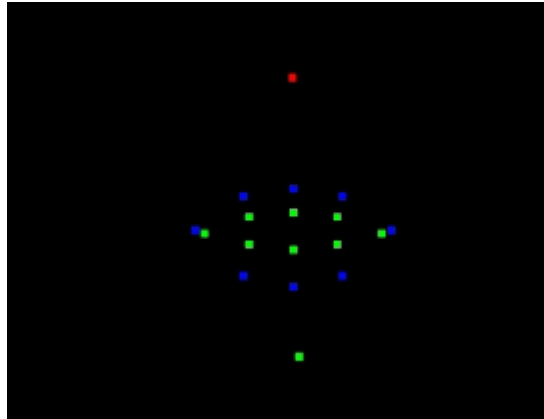
A Lucia programban a koszinuszos képleten kívül használnak egy másik képletet is a súlyok kiszámítására, ami a következő:

$$w_{ij} = e^{\frac{-d_i^2}{r_j^2}} \quad i = 1 \dots n; \quad j = 1 \dots 86;$$

Ahol w_{ij} a j-edik FP által mozgatott, a hatókörben i-edik vertex súlytényezője, d_i az FP és a hatókörben i-edik vertex éltávolsága, r_j pedig a j-edik FP hatókörének sugara. Ezt a képletet én nem használtam, mert ez az alkalmazás sokkal időigényesebb, hiszen akkor is számol a száj környékén súlytényezőket, amikor arra az adott vertexre nem is hat az FP.

8.5. Pontok áthelyezése a saját modellembe

Egy jeltolmács által bemondott szöveg videójának elemzése és a pontok automatikus kinyerése után, a projektnek ezzel a részével foglalkozó társaimtól megkaptam a száj környéki pontoknak (24. ábra) az eredeti és a beszéd közbeni elmozdulásnak a koordinátáit. Ezáltal a 2.2-től 2.8-ig (száj belső), illetve a 8.2-től 8.10-ig (száj külső) rendelkezésünkre állnak az MPEG4 szabványos pontok egy abszolút koordináta rendszerben (lásd 7. ábra). Ezek után a bemondó jeltolmács FAPU-it már igazán gyerekjáték meghatározni, hiszen csak egy olyan képet kell keresni a videón, amelyen a bemondó szája alaphelyzetben van, például a beszéd egyik szünetében, és lemérni a szájszélességet és a száj-orr közti távolságot. Így megkapom a számunkra érdekes FAPU-kat: MW-t és MNS-t (lásd 8. ábra).



ábra 24: A videóból levett pontok

A saját modellem FAPU-ját – amiből csak egy van (MW), mert csak a száját modellezem –, ugyanezzel a módszerrel határozom meg, azaz lemérem a száj csukott állapotában a két széle közti különbséget. Ha már megvannak mindkét rendszer FAPU-i, pontosabban a két rendszer FAPU-inak az aránya, és a koordináta rendszerek közti transzformációhoz szükséges méretbeli különbségek, akkor a videóból kinyert pontokat bele tudom helyezni a saját szájmozgást modellező rendszerembe.

Az MPEG4 szabványban minden egyes FAP-hoz definiálva van a megfelelő FAPU, a csoport amihez tartozik, a pozitív elmozdulás iránya, hogy vajon az FP egy, vagy két irányba mozgatható-e és az elmozdulásának a maximuma.

Ezáltal teljes útmutató van a kezünkben és a saját rendszerünkben is ezeket a paramétereket kell alkalmazni. A már meghatározott FAPU-kkal elő tudom állítani a szabványban megadott maximumokat a saját FP-im elmozdulásához. A videofelvételből kinyert koordináták segítségével pedig be tudom állítani a az FP-khez tartozó hatókörök nagyságát. Ha megnézem az összes FP-re az elmozdulásakor mért legnagyobb távolságot, akkor a hatókör sugarát úgy tudom beállítani, hogy ez az elmozdulás legyen a sugár 60%-a, ahogy azt az előző pontban tárgyaltam. Tehát ezeket a maximális távolságokat be kell szorozni a 0,6-nek vett peremfeltétel reciprokával, ami 1,667, hogy megkapjuk az általam készített szájmozgást modellező rendszerre jellemző valós hatókört az összes FP-hez.

Így létre tudok hozni egy új modellt ami a száj mozgásának a vizsgálatára kiválóan alkalmas. Az általam létrehozott alkalmazással tudom tesztelni a pontok mozgását, azok egymásra hatását és tudok kísérletezni a hatókörök területével, hogy megtaláljam az optimumokat. A modell legnagyobb erősségei az egyszerűségében, átláthatóságában, a könnyen kezelhetőségében és a kis terjedelmében rejlenek.

A Lucia pontjai viszonylag egyszerűen kinyerhetők az eredeti programból (25. ábra), ugyanis egy külön erre a célra írt header fájlban vannak definiálva, a hatókörökkel és a FAPU-kkal együtt.

```
// száj belső bal széle
fdp[2][5][XCOORD]=-1.21;           // angolo interno dx della bocca
fdp[2][5][YCOORD]=-2.0;
fdp[2][5][ZCOORD]=8.4;
fFdpZona[2][5]=0.1;               // Torzítási Zóna kiterjedése
bFdpMand[2][5]=1;
bFapIndex[2][5]=TRUE;
```

ábra 25: Az FDP-k deklarációja

Áttéve az általam készített modellbe ezeket a pontokat a hozzájuk beállított hatókörökkel, és ugyanazt a szöveget elmondatra, hasonló hibákat tapasztaltam, mint az eredeti Luciánál, azaz a mozgítás hatására több helyen is szétesett, megtört a modellem. Ebből arra következtettem, hogy nem az én modellemmel van a baj, hanem a Lucia beállításával. Például a hatóköreinek a beállításával. Sok helyen megfigyelhető, hogy egy darabig szépen mozognak a Lucia pontjai, majd hirtelen szétesik a mozgása.

Tehát nem egyre rosszabb és egyre nagyobb mértékben esik szét a mozgása, hanem egy bizonyos pont után – amikor átlépi a hatókör határát – teljesen rossz lesz. Erre a megfigyelésre alapozom azt az állításomat, hogy a Luciának a szája környékén elhelyezkedő FP-khez túl kicsi hatóköröket rendeltek. Ezért készült az én programom, hogy ezt a problémát jól lehessen vizsgálni és modellezni.

9. Javaslat a rajzolás gyorsítására

Továbbá, észrevettem hogy jelentősen lehet javítani a kirajzolás minőségén abban az esetben, ha memóriahiánnyal küzd a PC, vagy a mobiltelefon. A rendszer jelen állapotában egymás után rajzolja ki a FAP-okat és ha elfogy az ideje, akkor darabos lesz a modell mozgása. Egy sorba rendezéssel azonban ez orvosolható. Ha a mozgató pontokat az elmozdulás mértéke szerint sorba rendezem, és a kirajzolás során a legnagyobb elmozdulású pontoktól halad a program a legkisebb elmozdulású pontok felé, akkor a darabos mozgás helyett egy finom mozgást kapunk. Ha idő vagy memóriahiányba kerül a processzor, akkor egy peremfeltétel megadásával arra kényszeríthetem, hogy csak a jelentős FAP-okat rajzolja ki, és az egy határértéknél kisebb elmozdulásokat hanyagolja el. Így sohasem fordulna elő

olyan eset, hogy jelentős elmozdulással bíró FAP-ok nem kerülnek kirajzolásra, és néhány % elmozdulású FP-k kirajzolására pazarolja a program az erőforrásait.

10.Következtetések

Az MPEG-4 szabványnak és más fejmodelleknek, más rendszereknek az elemzésének a tükrében, arra a következtetésre jutottam, hogy az általunk kidolgozott rendszer igen jól működik, hiszen csak pár százalék eltérést mutatott a siketekkel végeztetett teszt.

A Lucia korlátainak a felderítése után azt állapítottam meg, hogy ez a modell ilyen formában is a számunkra legmegfelelőbb modell. Számos hibája és hiányossága van, de nem kell új modellt keresnünk, az általam bemutatott szájmozgást modellező programmal elvégzett tesztek azt bizonyítják, hogy az összes pont elemzésével és a hatóköröknek az újragondolásával jelentős javulást érhetünk el a leolvashatóság terén. Az általam létrehozott egyszerű de hatékony új modellel elvégzett kísérleteim során elsősorban az extrémumokra koncentráltam, mert a határérték problémák ott lépnek fel a leggyakrabban.

A munkám alapján arra az eredményre jutottam, hogy az FP-k elmozdulását 60%-ban kell limitálni a hatókörön belül, különben a textúra szétszakadhat a hatókör szélén, ami nagymértékben rontja a modell megjelenését, és a leolvashatóságot.

Összefoglalva a videóból nyert pontok belehelyezését az én modellemben: kijelenthetem, hogy az általunk használt pontok (amelyek a videóból származnak), jól vannak megválasztva, és a mozgásukkal sincs probléma, hisz azok egy élő személynek a szájmozgását reprezentálják, ráadásul az illető jeltolmács, így a siketek számára a legmegfelelőbb bemondó. A Lucia pontjait áttéve az én modellemben és ugyanazt elmondatva vele, mint a jeltolmáccsal a videón, a modell ugyanúgy szétesik, mint az eredeti Lucia. Ezzel igazoltam, hogy az én modellem helyesen működik és a kezdeti feltevésünk is beigazolódott: a Luciában lévő pontoknak a hatókörei túl kicsire vannak beállítva. Ezeket a hatóköröket egyenként felül kell vizsgálni.

Javaslatot tettem egy igen egyszerű módosításra, mégpedig, ha az elmozdulás nagysága szerint sorba rendezzük a kirajzolandó FAP-okat, akkor sem lesz darabos és szétesett a fejmodell mozgása, ha a processzor kifutna az időből, vagy a memóriából.

Az általam felvetett módosítási javaslatokat még ellenőrizni kell. Ehhez újra a siketek segítségét kell kérni, hogy új tesztek megírásával, mérésekkel is alá tudjam támasztani az eredményeimet.

Köszönetnyilvánítás

Hálásan köszönöm a projektben résztvevő siketeknek és nagyothallóknak a tesztek és interjúk során az együttműködést. Külön köszönettel tartozom Takács György Professzor Úrnak az irányelvek meghatározásáért, az útmutatásokért, Tihanyi Attilának pedig a folyamatos konzultációért, valamint a sok, gyakorlatban igen jól használható ötletéért. Köszönöm Bárdi Tamásnak, Feldhoffer Gergelynek és a mobilos csoport összes tagjának az együttműködését.

11. Irodalomjegyzék

- [1] I. Pandzic and R. Forchheimer, „MPEG-4 Facial Animation: The Standard, Implementation and Applications”, Wiley, 2002.
- [2] Ekman, P. & Friesen, W. V. *Facial action coding system: A technique for the measurement of facial movement*. Palo Alto, Calif.: Consulting Psychologists Press. 1978
- [3] Ekman, P. and Friesen, W.V. (1977) *Manual for the Facial Action Coding System*, Palo Alto: Consulting Psychologists Press
- [4] Waters K., „A muscle model for animating three-dimensional facial expressions” *Computer Graphics*, 21: pp. 17-24, 1987.
- [5] Parke F. I., „Parametrized models for facial animation”, *IEEE Computer Graphics*, 2(9): pp. 61-68, 1982.
- [6] <http://tcc.itc.it/>
- [7] <http://xface.itc.it/index.htm>
- [8] Czap L., „Audiovizuális beszédfelismerés és szitézis”, PhD értekezés, 2004.
- [9] G. Takács, A. Tihanyi, T. Bárdi, G. Feldhoffer, B. Srancsik: “Speech to facial animation conversion for deaf applications” 14th European Signal Processing Conf., Florence, Italy, September 2006.
- [10] G. Takács, A. Tihanyi, T. Bárdi, G. Feldhoffer, B. Srancsik: “Database Construction for Speech to Lip-readable Animation Conversion” 48th Int. Symp. ELMAR-2006 on Multimedia Signal Processing and Communications, Zadar, Croatia, June 2006.
- [11] Takács Gy. *et al.*, „Siketek beszédkommunikációját mobiltelefonnal segítő eszközök fejlesztése”, Szakmai beszámoló (EMI – 0008 / 03 KPI), PPKE-ITK, 2005.
- [12] Feldhoffer G., Bárdi T., Jung G., Hegedűs I. M., „Mobiltelefon alkalmazások siket felhasználóknak”, *Híradástechnika*, 2005/10.

-
- [13] Segal, Mark; Arkley, Kurt: The OpenGL Graphics System: A specification, October 2004.
- [14] Cosi P., Fusaro A., Tisato G., „LUCIA a New Italian Talking-Head Based on a Modified Cohen-Massaro's Labial Coarticulation Model”, Proceedings of Eurospeech 2003, Geneva, Switzerland, September 1–4, 2003, Vol. III, pp. 2269-2272.
- [15] <http://www.turbosquid.com/3d>
- [16] Bárdi T., Feldhoffer G., Harczos T., Srancsik B., Szabó G. D., „Audiovizuális beszédatbázis és alkalmazásai”, Híradástechnika, 2005/10.
- [17] Ekman, P. & Friesen, W. V. (1975). *Unmasking the face. A guide to recognizing emotions from facial clues*. Englewood Cliffs, New Jersey: Prentice-Hall.
- [18] Paul Martz, „OpenGL röviden”, Kiskapu, 2007.
- [19] [Zsófia Ruttkay , Han Noot, Animated CharToon faces, Proceedings of the 1st international symposium on Non-photorealistic animation and rendering, p.91-100, June 05-07, 2000, Annecy, France](#)
- [20] Srancsik Bálint, Diplomaterv, „MPEG-4 paraméterekkel vezérelhető fej-animáció rendszer” , 2006