

TDK-DOLGOZAT



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

ARCANIMÁCIÓ MOZGÁSRÖGZÍTÉS SEGÍTSÉGÉVEL

Szerző: **Könyvesi Gábor**
mérnök informatikus szak, III. évf.

Konzulens: **Vámossy Zoltán**
egyetemi docens

Budapest, 2012.

Tartalomjegyzék

I. Bevezető.....	7
1. Célkitűzések.....	7
I. Irodalomkutatás.....	8
1. Mozgásrögzítés (motion capture).....	8
1.1 Teljes test mozgásának rögzítése markerek segítségével.....	8
1.2 Arcmozgás-rögzítés markerekkel.....	9
1.3 Marker-mentes arcmozgás rögzítés.....	9
1.4 Valós idejű marker-mentes arcmozgás rögzítés.....	9
1.5 Ajak- és szemkörnyék követése marker-mentes módszerekkel.....	9
2. A mozgáskövetés kihívásai.....	10
II. A rendszer működésének elvi háttere.....	11
1. Kalibráció.....	11
1.1 Pinhole kamera modell.....	11
1.2 A kamera kalibrációs mátrixa.....	12
1.3 Torzítások.....	13
1.4 Külső paraméterek.....	13
2. Markerek szegmentálása.....	14
2.1 Sötét markerek.....	14
2.2 Világos markerek.....	14
2.3 Intenzitás-alapú markerek küszöbölése.....	14
2.4 Élénk színű markerek.....	15
2.5 Kapcsolódó komponens analízis.....	16
3. Sztereó megfeleltetés.....	17
3.1 Markerpárok keresése.....	17
3.2 Epipoláris geometria.....	17
3.3 Koordináták pontatlanságának kezelése.....	18
4. Háromszögelés.....	18
4.1 Projekciós mátrixok.....	18
4.2 Problémafelvetés.....	19
4.3 Lineáris háromszögelés.....	19
4.3.1 Homogén megoldás.....	19
5. Pontkövetés.....	20
6. Deformálás.....	20
6.1 Virtuális arc megalkotása.....	20
6.1.1 Alap mesh.....	20
6.1.2 Származtatott felületek.....	20
6.2 Animációs módszerek.....	20
6.3 Korreláció.....	21
6.3.1 Dimenziószám csökkentése.....	21
6.3.2 Interpolációs függvények.....	21
III. Megvalósítás.....	22
1. Hardver- és szoftverkövetelmények.....	22
2. A rendszer szerkezete futtatási egységek szempontjából.....	23

2.1.1 Kalibráció.....	23
2.1.2 Mozgásrögzítés.....	23
2.1.3 Beépülő modul.....	23
2.1.4 Modulok közti kommunikáció.....	23
3. A virtuális arc modellje.....	24
4. Működés.....	25
4.1 Kalibráció.....	25
4.2 Mozgásrögzítés.....	25
4.2.1 Markerek megtalálása.....	25
4.2.2 Térbeli rekonstrukció.....	25
5. Eredmények értékelése.....	26
IV. Irodalomjegyzék.....	27

I. Bevezető

Számítógépes grafikában a háromdimenziós alakok mozgatása nem könnyű feladat. Az emberi látás különösen érzékeny az emberi mozdulatokra, az arcok apró rezzenéseire. Észlelésük, még ha nem is tudatosan az emberben, megváltoztatja a néző elgondolását a vizsgált személyről. Ezeknek a mozdulatoknak kézi megvalósítása időigényes, sok tapasztalatot igényel, és az eredmény minősége gyakran elmarad a valóságghű animációtól.

Ennek a problémának a megoldására találták ki a *motion capture* (mozgásrögzítés, röviden: MoCap) technológiát. Alkalmazásával lehetővé vált, hogy a valós mozgást rögzítsék és digitalizálják, majd később ráültessék virtuális alakokra. A mai filmekben és videojátékokban látható minőségi animáció részben ennek a technológiának köszönhető.

Az arc és a test mozgásának jellegi különbségei miatt egyes módszerek előnyösebbek az egyik, kevésbé előnyösek a másik esetben. Az arc mozgása nehezebben írható le mereven mozgó (*rigid*) tagokkal, ugyanis a fej egyetlen ténylegesen ilyen jellegű eleme az állkapocs (rögzített koponya esetén). Az arc további mozdulatait az izomközegek összehúzódása és a bőrfelület deformációja, redőződése adja.

E projekt tárgya egy arcmozgásra specializálódott mozgásrögzítő rendszer megalkotása, amely könnyen hozzáférhető és intuitív a végfelhasználó számára. A projekt megvalósítása szorosan együttműködik egy háromdimenziós modellező és animációs szoftverrel annak érdekében, hogy a projekt eredményei a gyakorlatban közvetlenül alkalmazható megoldást nyújthasson.

1. Célkitűzések

A lehetőségek közötti mérlegelés során a következő szempontokat igyekeztem szem előtt tartani. Ezek tekinthetők a projekt célkitűzéseinek:

- Valós idejű vagy azt minél jobban megközelítő működés: A mozgás rögzítésének folyamatát sokkal intuitívabbá teheti, ha a azonnali visszajelzést kapunk. Az általános és a valós időben működő rendszerek közötti teljesítménykülönbség akár nagyságrendi különbség is lehet, így egy ténylegesen valós időben működő rendszert nehéz felépíteni.
- Általánosság: Olyan rendszer megalkotása, ahol a vizsgált arc (aminek a mozgását rögzítjük) és a mozgatott virtuális arc között nem kell megkövetelni a szoros kapcsolatot. Így egyetlen színész sokféle virtuális arcot meg tud mozgatni.
- Alacsony hardverkövetelmény: Azokat a megoldásokat részesítem előnyben, amelyek nem követelnek specializált hardverelemeket és a szükséges elemek bárki számára elérhetőek és könnyen beszerezhetőek.
- Szoftverfüggségek minimalizálása: A megvalósítás során azokat a szoftverkönyvtárakat, alkalmazásokat preferálom amik minél jobban lefedik az adott feladatkört, hogy a függőségek száma minimális legyen. Emellett az általános, illetve több operációs rendszeren is működő megoldásokat részesítem előnyben.
- Nyitottság a továbbfejlesztés felé: Általános nyitottság a rugalmas, bővíthető, (platformok között) hordozható megoldások felé.

I. Irodalomkutatás

1. Mozcásrögzítés (motion capture)

A mozgásrögzítés egyik válfaja a jelölőpontokat (*marker*) alkalmazó mozgásrögzítés. A markerek segítenek abban, hogy az arcot, mint bőrfelületet, elemezni tudjuk optikai rendszerek segítségével. Itt a bőrfelület deformálódását segítenek érzékelni. Teljes test esetén a markereket a végtagok forgáspontjai környékén helyezik el így a pontok helyzetéből következtetni tudunk a tagok orientációjára és egymással bezárt szögére. Az optikai mozgásrögzítésben passzívnek nevezzük azokat a markereket amik nem bocsájtanak ki saját fényt.

1.1 Teljes test mozgásának rögzítése markerek segítségével

Egy marker alapú, teljes test mozgását rögzítő rendszer vázlatos működése Guerra-Filho [Gue05] alapján:

- Felszerelés: Körben elhelyezett triggerelhető kamerák, egyszínű öltözet a színésznek (a továbbiakban a színész utal arra a személyre akinek a mozgását rögzítjük), egyszínű terem, egyenletesen, lehetőleg szórt fénnel megvilágítva.
- Előkészületek:
 - Kamera kalibrálás: A kamerák összehangolása érdekében egy adott mintájú táblát tartanak a kamerák felé. Ha nem lehet a táblát úgy tartani hogy minden kamera által látható legyen, akkor több részletben hangolják össze a kamerákat.
- Markerek érzékelése: A kameraképen többféle módszerrel megkeressük azokat a pontokat ahol feltételezhetően marker található. A különböző algoritmusok eredményét összevetjük, ezzel csökkentve a hibás találatok számát.
- Térbeli megfeleltetés: A kameraképéből kinyert markerpontokat megfeleltetjük a többi kamera képen találhatóakkal. A folyamat során figyelembe kell vennünk, hogy kisebb pontatlanságok lesznek a kameraképek között, ezért nem várhatunk el tökéletes egyezéseket. A pontok projekcióiból kiszámítjuk a pontok térbeli helyzetét. A művelet eredménye egy háromdimenziós ponthalmaz.
- Időbeli megfeleltetés: Az adott időpillanatokban kinyert háromdimenziós ponthalmazok közötti megfeleltetés. Markerpontok az idő során eltűnhetnek, majd újra felbukkanhatnak. A művelet eredménye az elmozdulási (*translational*) adathalmaz.
- Utólagos feldolgozás (post-processing):
 - Címkézés: Megadjuk hogy az egyes markerek a virtuális embermodell melyik pontjaihoz tartoznak. Ez történhet kézzel, vagy automatikusan, ha rendelkezünk a megfelelő antropometriai adatokkal. Utóbbi esetben sokat segít, ha színész a felvétel kezdetekor egy alap testtartást vesz fel.
 - Adatok kiegészítése: Ha egy marker mozgási adata hiányos (pl. a felvétel során egy ideig takarásba került), akkor a hiányzó adatok kiegészíthetők a meglévő adatok és szomszédos markerek adatainak segítségével.

Ez a vázlat csak a projekt szempontjából fontos részeket emeli ki. A rendszerben szűrkeskálás kamerákat használtak, viszont felmerült, hogy színes kamera esetén javíthatja a pontosságot és könnyítheti a címkézést, ha többféle színű markert használunk.

1.2 Arcmozgás-rögzítés markerekkel

Többek között ezt a lehetőséget is kihasználta Guenter et al. rendszere [Gue98], ami már arcmozgás rögzítésére szolgált és szintén marker alapú. Itt 6-féle színű, összesen 182 jelölőpontot ragasztottak a színész arcára, ügyelve, hogy az azonos színű pöttyök a lehető legtávolabb kerüljenek egymástól. A markerek fluoreszcens festést kaptak és a színész arcát látható fény mellett UV tartományú fénnel is megvilágították, így biztosítva a megfelelő környezetet a jelölőpontok érzékeléséhez.

A rendszerben a színész fejét többé-kevésbé rögzítették, mivel úgy vélték, hogy a nyak mozgása ebben az esetben rontana az eredményen. Emellett javasolták, hogy a nyak mozgását egy független testmozgás-rögzítési munkafázisban végezzék. A nyakmozgás kiküszöbölésének problémájával későbbi munkákban is találkozhatunk. Jones et. al. rendszerében [Jon10] a kamerát a fejhez rögzítették.

1.3 Marker-mentes arcmozgás rögzítés

Alexander et al. [Ale09] projektjének célja az volt, hogy képesek legyenek egy valós arcot és mozgását újraalkotni virtuális formában úgy, hogy az teljesen élethű hatást keltsen. Az kutatást végző egyetem munkatársai főként a színész arcának digitalizálásában ténykedett, az animációt egy külső cégre, az Image Metrics-re bízta.

A motion capture rendszer működésének részleteiről keveset tudni, de Alexander et al. leírása alapján ez egy markermentes rendszer amely a feldolgozás során jelentős kézi beavatkozást igényel. Ez a technológia szoftvertermékként kevésbé, inkább szolgáltatásként jellemezhető. Úgy jellemezte, hogy a rendszerben egyfajta egyensúly van az automatizált és a kézi feldolgozásban.

1.4 Valós idejű marker-mentes arcmozgás rögzítés

Weise et. al. rendszerében [Wei09] egy strukturált fény alapú háromdimenziós szkennel segítségével valós idejű marker-mentes rendszert tudtak létrehozni. Itt a színészek arcáról létrehoztak a szkennel segítségével egy kezdeti adathalmazt a különböző arckifejezésekből. (Alexander et. al. munkájának is fontos momentum a egy hasonló adathalmaz elkészítése volt) Ez egyrészt lehetővé tette a jó minőségű mozgásrögzítést, másrészt segít párhuzamot vonni a különböző arcok között.

A már valós idejű feldolgozásnál ezt az adatbázist veszik alapul, és az éppen észlelt felületet igyekeznek megfeleltetni az adatbázis arcmodelljével. A folyamat során külön figyelmet fordítanak a szemhéjra és a az ajkak belső körvonalára. (További kényes területek a szemek, és a nyelv. Ezeket marker alapú módszerekkel nem lehet követni. Ez a rendszer ezeket a részeket figyelmen kívül hagyja.)

1.5 Ajak- és szemkörnyék követése marker-mentes módszerekkel

Chuang et. al. [Chu02] a szemöldököket, a szemek kontúrját, a pupillákat és az ajkak belső és külső ívét követte és ezen felül három markerpontot használt a fej orientációjának meghatározásához.

2. A mozgáskövetés kihívásai

Mivel az adatokat animálásra használjuk fel több kihívással is szembe kell néznünk [Gle02]:

- Az arckövetéssel és arcfelismeréssel ellentétben a részletek is fontosak.
- Apró remegések és rángások váratlan helyeken is előfordulhatnak. Gyakran a rendszer pontatlanságából, a képkockák közötti folytonosság hiányából fakadnak. Ezek a néző számára könnyen észlelhetők és zavaróak.
- A forráskép szűrése csak korlátozottan lehetséges, mert meg kell őriznünk a nagy frekvenciás adatokat.
- A furcsa és gyakran kiszámíthatatlan mozdulatok nehezítik egy robusztus módszer kialakítását.

II. A rendszer működésének elvi háttere

E fejezet részletezi a feladat megoldásának eszközeit és igyekszik minél tisztább képet adni a felhasznált algoritmusok működéséről, tulajdonságairól. Emellett megemlíti a szakirodalomban fellelhető jelentősebb alternatívákat.

1. Kalibráció

Ha térbeli rekonstrukcióhoz sztereó kamerás rendszert alkalmazunk ismernünk kell a kamerák belső és külső paramétereit, és figyelembe kell vennünk a kamerák torzításait. A paraméterek (külső, belső, torzítási) ismeretének hiányában, azokat a kamerák kalibrációja során meg kell becsülnünk. [Zha99]

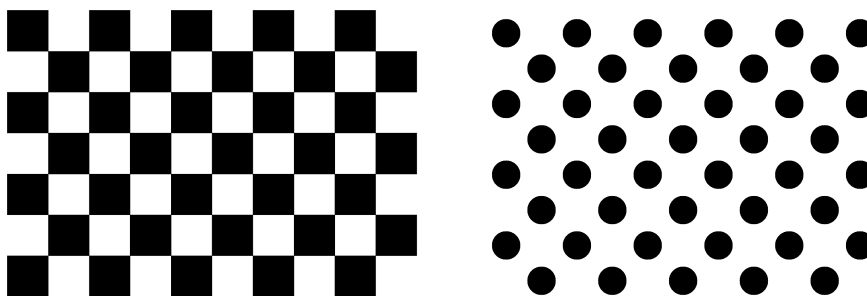


Fig. 1: Kettő gyakran alkalmazott kalibrációs mintatípus.

A kalibráció viszonyítási alapjául a kalibrációs minta szolgál. A minta általában egy sík lap, amelyen jól és pontosan felismerhető jellegzetes pontok találhatóak, előre ismert alakzatban. A leggyakrabban alkalmazott minta sakktábla jellegű, és a sötét négyzetek találkozásánál fellelhető sarokpontokat használjuk fel a kalibrációhoz. (Fig. 1)

1.1 Pinhole kamera modell

A kamerák modellezésének alapjául a pinhole kamera modell szolgál [Har04 153. o.]. A modell szerint a kamera szenzorára érkező fénysugarak egy közös ponton haladtak át, a kamera vetítési középpontján. A képsíkon (szenzoron) a tárgy fordított képe jelenik meg. A képsíkot képzeletben a kamera középpontja elé szoktuk képzelni, így a képzeletbeli képsíkon már az egyenes állású kép jelenik meg.

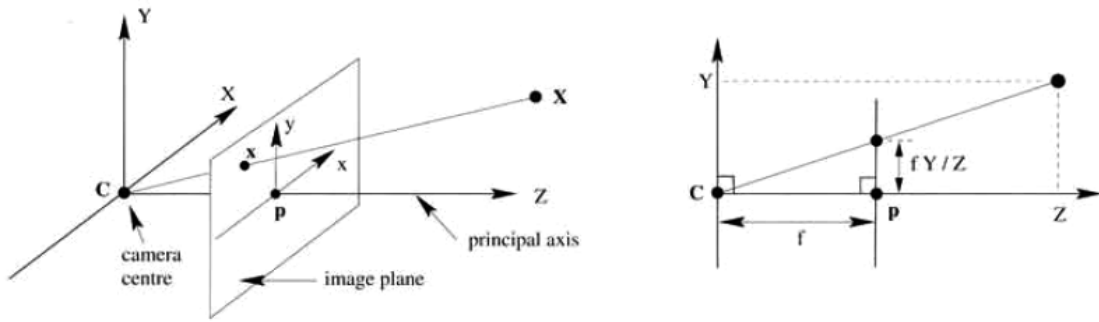


Fig. 2: A pinhole kamera modell. Forrás: [Har04]

A Fig. 2 ábrán C a kamera vetítési középpontját, p a képsík középpontját (*principal point*) jelzi. A C és p pontokat összekötő tengely az optikai tengely (merőleges a képsíkra), és f a C és p pontok távolsága (fókusz távolság). Ebben az idealizált környezetben, a fókusz távolságot felhasználva megalkotható egy, a kamerához tartozó projekciós mátrix, amely egy X térbeli pontot levetít a képsíkra.

$$x = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} X$$

Ahol X egy háromdimenziós térbeli pont és x a pont képsíkra vetített pozíciója. A képletben x is háromdimenziós, ahol a harmadik koordináta a mélységet jelöli. A mélységről rekonstrukció során nincs tudomásunk ezért az x pontot egy két dimenziós pontnak tekintjük és a harmadik koordinátát 1-re rögzítjük.

1.2 A kamera kalibrációs mátrixa

A gyakorlatban további paraméterekkel is számolnunk kell az általunk használt szenzorok eltérései és jellegzetességei miatt. A leggyakoribb ilyen paraméterek a képsík középpontjának helyzete a képen, egy képpont (*pixel*) mérete, és a képpont szélességének és magasságának aránya. Ezek alapján bővíthetjük a projekciós mátrixot:

$$x = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} X$$

Ahol x_0 és y_0 a képsík középpontjának x és y irányú pixelkoordinátája, α_x és α_y a fókusz távolság pixelekből kifejezve. A pixel magassága és szélessége eltérhet, ezért van szükség a két különböző α paraméterre. A f és az α paraméterek viszonyát a következőképpen írhatjuk le:

$$\alpha_x = f \cdot m_x, \quad \alpha_y = f \cdot m_y$$

Ahol m_x és m_y egy pixel képsíkon vett méretének reciproka az adott irányban. Hasonló egyenletek írhatóak fel az x_0 és y_0 paraméterekhez is:

$$p = (p_x, p_y) \\ x_0 = p_x \cdot m_x, \quad y_0 = p_y \cdot m_y$$

Ezt a bővített mátrixot a továbbiakban **K**-val jelölöm. A mátrixnak 10 szabadsági foka van.

Van lehetőség a mátrix további bővítésére is, de erre csak nagyon ritka és speciális esetekben van szükség. A legtöbb kameránál elegendő az eddigiek alapján kialakított mátrix.

1.3 Torzítások

Bár építhető pinhole jellegű kamera, a gyakorlatban használt kameráknál legtöbbször valamilyen lencsén vagy lencserendszeren keresztülhaladva érkeznek a fénysugarak a szenzorra. Ennek következtében sugárirányú (radiális) torzítás lép fel. (Fig. 3) [Har04 189.o., Bra08 375.o.]

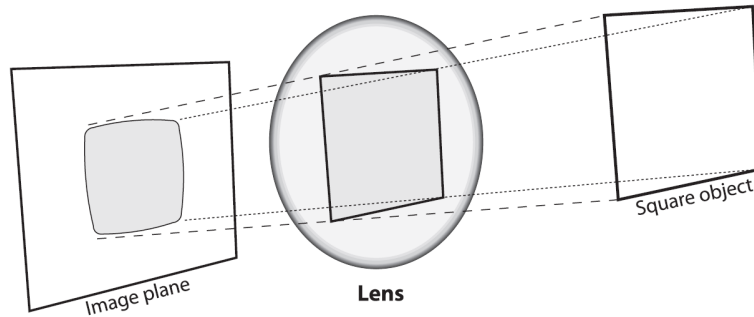


Fig. 3: A radiális torzítás hatása. Forrás: [Bra08]

A képsík középpontjában nincs torzítás, és a középponttól távolodva növekszik a torzítás mértéke. A torzítás mértékét leírhatjuk egy $L(r)$ függvénnyel, amelynek egyetlen paramétere a középponttól való távolság. A függvény jellemzésére a függvény Taylor sorának együtthatóit szokták használni. Enyhébb radiális torzítású optikák esetében az első két együttható is elegendő. Jellemző jelölésük k_1 és k_2 . Halszem optikák esetén további együtthatókat is felhasználnak.

A torzítások egy másik típusa a tangenciális torzítás. Ez gyártási hiba eredménye, melynek következtében a szenzor lapkájának síkja nem teljesen párhuzamos a képsíkkal. [Bra08 376.o.]

1.4 Külső paraméterek

Az eddigi paramétereket összefoglaló néven belső paramétereknek szokták jellemezni és egy kamera jellemzőit írják le. A külső paraméterek ezzel szemben két kamera viszonyát írják le, illetve a kamerák térbeli elhelyezkedését.

A két kamera relatív térbeli viszonyát jellemezhetjük egy eltolási és forgatási transzformációval. Az eltolás egy t vektorral, a forgatás egy $\mathbf{R}$ mátrixszal leírható. A világ koordináta-rendszer és egy kamera koordináta-rendszere közötti viszonyt az alábbi módon adhatjuk meg [Bra08 427. o.]:

$$\mathbf{X}_{tr} = \mathbf{R} \mathbf{X} + t$$

Ahol $\mathbf{X}_{tr}$ az $\mathbf{X}$ pont helyzete az új koordinátarendszerben. Ha $\mathbf{X}$ -et homogén koordinátákkal fejezzük ki, akkor a forgatást és eltolást kifejezhetjük egyetlen mátrixszal [Har04 156. o.]:

$$\mathbf{X}_{tr} = \begin{bmatrix} \mathbf{R} & t \end{bmatrix} \mathbf{X}$$

Ez alapján könnyen kifejezhető az $\mathbf{X}$ pont helyzete a kamerákhoz tartozó koordináta-rendszerekben:

$$\mathbf{X}_l = \begin{bmatrix} \mathbf{I} & \mathbf{o} \end{bmatrix} \mathbf{X}, \quad \mathbf{X}_r = \begin{bmatrix} \mathbf{R} & t \end{bmatrix} \mathbf{X}$$

Ahol X_l és X_r az X pont helyzetét jelölik a bal és jobb kamerák koordináta-rendszereiben. $\mathbf{R}$ és t a két kamera közötti relatív elforgatást és eltolást jelölik. $\mathbf{I}$ egy ($\mathbf{R}$ -el megegyező méretű egységmátrix), $\mathbf{0}$ pedig egy (t -vel megegyező méretű) nullvektor. X itt is homogén koordinátás térbeli pontot jelöl.

Bár a kamerák egymás felett is elhelyezkedhetnének, a projekt két egymás melletti kamerát használ, így a jobb érthetőség érdekében a kamerákra úgy hivatkozok, hogy jobb és bal kamera. A jobb oldali kamerához tartozó kifejezéseket a továbbiakban vesszővel jelölöm. (Például: x')

2. Markerek szegmentálása

A projekthez létfontosságú, hogy a markerek szegmentálása robusztus és stabil legyen, miközben azok közelednek, távolodnak és gyorsan mozognak a képen. A markerek típusát úgy választjuk meg, hogy jellemzői segítsék a szegmentálást, vagyis a pontok megtalálását. A markerek jellege változó lehet a felhasználástól függően.

A projekt passzív optikai markereket használ. Az ilyen markerek színe elüt a környezetétől, jelen esetben a bőrszíntől. Alakjuk általában gömb vagy korong. Ezek a markerek a képen összefüggő, megközelítőleg homogén, ovális vagy kör alakú foltokként jelennek meg. Emellett felhasználhatjuk, hogy viszonylag kicsik, és ideális esetben egymástól szeparáltak.

A továbbiakban felvázolom az általam kipróbált markertípusokat és a megtalálásukhoz használt módszereket.

2.1 Sötét markerek

Erre a célra szemceruzát használtam. Előnye volt, hogy könnyen beszerezhető és könnyen kezelhető. A kép feldolgozásakor további előny, hogy a szürke-skálás kamerakép is elegendő. A módszer érzékeny a fényviszonyokra, és nehéz olyan küszöbértéket találni (legyen az globális vagy lokális) ami stabilan jó eredményt adna. Emellett nehéz megkülönböztetni a markereket az arc egyéb sötét elemeitől (például a pupillától).

2.2 Világos markerek

Erre a célra fényvisszaverő fóliát használtam. Nehezebb beszerezni, mint az előbbi, viszont kevésbé érzékeny a fényviszonyokra. Feldolgozása hasonló az előzőhöz, viszont itt nagy segítség, hogy alacsony expozíciós időknél is jól láthatóak maradnak a markerek. A szem itt is zavaró tényezőként jelenik meg, a szem fehér részeit nehéz megkülönböztetni a markerektől.

2.3 Intenzitás-alapú markerek küszöbölése

Az eddigi markerek megtalálására a legjobb eredményt lokális küszöböléssel értem el. A medián alapú lokális küszöbölési módszer lényege az, hogy az eredeti kép és a kép medián-szűréssel módosított változatának különbségére alkalmazunk globális küszöbölést. Azok a részletek amelyek egy adott méretnél kisebbek és intenzitásuk eltér a környezetüktől, kiugróan magas (vagy alacsony) értéket kapnak. Az eredmény minőségét rontották a kontrasztos, hosszú körvonalak. (Fig. 4)



Fig. 4: Az eredeti kép és a kép medián-szűrt változatának különbsége. Bal oldalt az eredeti kép, középen a pozitív előjelű különbség, jobb oldalt a negatív előjelű különbség (minél sötétebb, annál nagyobb).

A probléma megoldására egy pixelen, különböző irányú mediánok abszolút érték szerinti minimumát vettem, majd az így kapott értéket vontam ki az eredeti kép pixeléből (Fig. 5). Az eredményképen azok a részletek kaptak magas értéket amelyek minden irányban „eltűntek” a medián-szűrt képekről.

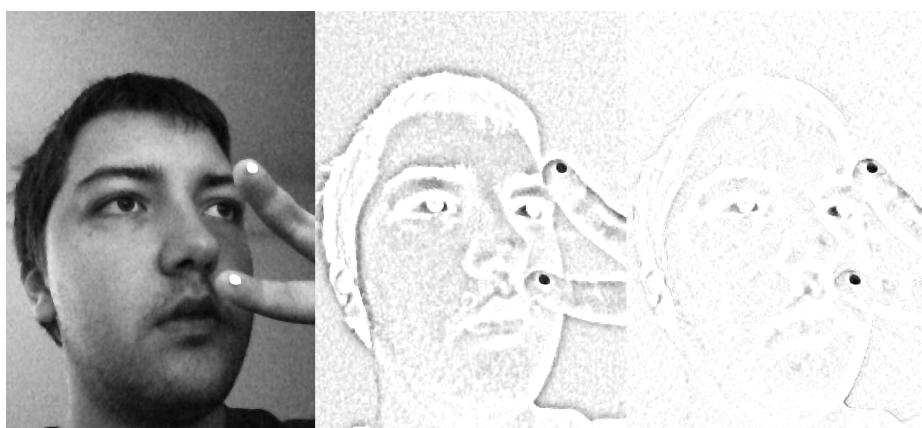


Fig. 5: Az eredeti kép és a medián-szűrt kép különbsége. Bal oldalt az eredeti kép, középen a különbség átlagos medián szűrést alkalmazva, jobb oldalt a különbség két, egymásra merőleges irányú medián felhasználásával (minél sötétebb, annál nagyobb a különbség).

A módszer a sötét és világos markerekre is működik. Emellett egy, ha sötét és világos markereket is használunk egy adott konfigurációban a különbség előjeléből megállapítható, hogy egy adott marker melyik csoportba tartozik (Fig. 4).

A medián alapú küszöbölés további előnye, hogy egy kép medián szűrése viszonylag hatékonyan megvalósítható úgy, hogy a számítási komplexitása nem függ a szűrőkernel méretéről.

2.4 Élénk színű markerek

Az élénk színű, esetleg enyhén fluoreszcens markerek használata valamivel egyszerűbb az előbbieknél, ugyanis itt a szín erős támpontot ad a szegmentáláshoz. (Fig. 6) Természetesen a szürke-skálás kép itt már nem elegendő. A képet HSV színtérbe konvertáltam, mivel ebben a színtérben a színeket árnyalat, élénkség és intenzitás alapján lehet értékelni, ezáltal használata sokkal intuitívabb. A szín szerinti globális küszöbölés is elegendőnek bizonyult.



Fig. 6: Színes markerek

2.5 Kapcsolódó komponens analízis

A küszöbölés során a kép pixeljeit osztályokba soroljuk. Egyetlen küszöb esetén előtér- és háttérpíxelek osztályokba. Ebben az esetben az előtérpíxelek jelentik a markereket. A kapcsolódó komponens analízis célja, hogy megtalálja az azonos osztályba tartozó összefüggő foltokat majd megállapítsa a foltok jellemzőit.

Az első lépés (címkézés) során egy adott folt pixeljeinek egy címkét (a foltot azonosító értéket) ad. A jellemzők kiszámítása algoritmustól függően történhet a címkézés során, vagy külön menetben. A projekt szempontjából a legfontosabb jellemző a súlypont, viszont számos más jellemző is kiszámolható úgy, mint terület, kerület, főtengely iránya, kompaktság, nyomatékok (Fig. 7). Maga a címkézés is történhet egy vagy több menetben.



Fig. 7: Markerek a kapcsolódó komponens analízis után. A markereket piros körök jelzik. A körök középpontjai a markerek súlypontjai, a körök sugarát a markerek területe határozza meg. A megtalált markereket piros nyilakkal emeltem ki.

A kapcsolódó komponensek címkézését nagyon lelassítja, ha labirintusszerű foltokat kell címkéznie. A probléma megoldására több algoritmus is született, azonban markerek esetén abban a szerencsés helyzetben vagyunk, hogy a címkézendő területek összefüggő és viszonylag kompakt foltok, így egyszerűbb algoritmusokat is hatékonyan tudunk alkalmazni.

3. Sztereó megfeleltetés

3.1 Markerpárok keresése

Miután mindkét kamera képét szegmentáltuk, két ponthalmazt kapunk amely tartalmazza a megtalált markerek pozícióját miután a pontokat az adott képsíkra vetítettük. A pozíciók pixelkoordinátákat jelölnek. A most következő lépésben, igyekszünk megtalálni az egyik kamera képen található pontokra megállapítani, hogy a másik kamera képen talált pontok közül melyikkel alkotnak párt. A párhoz tartozó pontok egy közös térbeli pontot képviselnek, egyetlen térbeli pont képsíkra vetítésével jöttek létre.

Két kamera ilyen jellegű pontjai között a fundamentális mátrix teremti meg a kapcsolatot. Ha a térbeli pontot hiba- és torzítás mentesen vetítjük le a képsíkokra akkor a fundamentális mátrix ismeretében a következő egyenlet írható fel [Har04 245. o.]:

$$x'^T F x = 0$$

Ahol x a bal kamera képen található pont koordinátája, x'^T a jobb kamera képen található pont koordinátájának transzponáltja és F a fundamentális mátrix. Az egyenletet szokás korrelációs kritériumnak is nevezni, mivel a két pont csak akkor alkothat párt, ha az egyenlet teljesül.

Ha pixelkoordináták helyett normalizált koordinátákat használunk, a fundamentális mátrix feladatát az esszenciális mátrix látja el. A két mátrix viszonyát a következő egyenlet írja le.

$$E = K'^T F K$$

Ahol E az esszenciális mátrix, K és K' a bal és jobb kamerák belső paramétereit magukba foglaló kalibrációs mátrixok.

3.2 Epipoláris geometria

Az epipoláris geometria a kamerák közötti belső vetítési viszonyt írja le. A viszonyt a korábban említett fundamentális mátrix összegzi. [Har03 239.o.]

A fundamentális mátrix az egyik kamera képen található pontot egy egyenesre képzi le. Ezen az egyenesen fekszik a pont párja a másik kamera képen. Ezt az egyenest epipoláris egyenesnek szokták nevezni.

$$l' = F x, \quad l = F'^T x'$$

Ahol l' a jobb képen látható, x -hez tartozó epipoláris egyenes és l a bal látható, x' -höz tartozó epipoláris egyenes.

Minden, a képsíkon található ponthoz felírható egy ilyen epipoláris egyenes és ezek az egyenesek egy közös pontban találkoznak. Ezt pontot epipólusnak nevezik. Az epipólus térbeli megfelelője a kamera vetítési középpontja (origója). Ez epipólusokra a következő egyenletek érvényesek:

$$F e = 0, \quad F'^T e' = 0$$

Ahol e és e' az epipólusok koordinátái a bal illetve a jobb képen, és 0 egy nullvektor.

A több pontpár esetén a megfeleltetés nem egyértelmű, ha a térbeli pontok a kamerák origóival egy síkot alkotnak. Ebben az esetben az epipoláris egyenes több ponton is keresztül halad.

Továbbá a megfeleltetés nem egyértelmű, ha a egy térbeli pont az origókat összekötő egyenesre esik. Itt a rekonstrukció sem lehetséges, ugyanis nem tudjuk megállapítani, hogy az egyenesen hol helyezkedik el a pont. Ebben az esetben a pont a képsíkokra vetítve az epipóluson helyezkedik el mindkét kamerán.

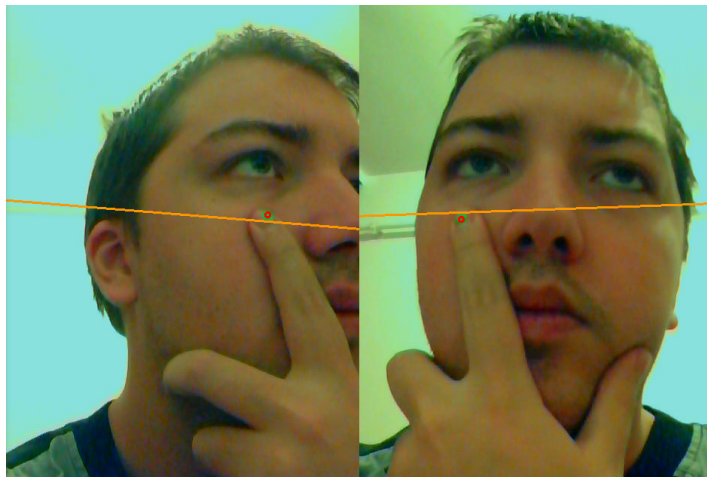


Fig. 8: Egy ponthoz tartozó epipoláris vonalak (narancssárga).

3.3 Koordináták pontatlanságának kezelése

A gyakorlatban nem várható el, hogy a korrelációs kritérium tökéletesen teljesüljön a képalkotás és feldolgozás során fellépő zajok és hibák miatt. Valós esetben azt vizsgáljuk, hogy a $x^T F x$ kifejezés abszolút értéke egy bizonyos határérték alatt van-e vagy sem. Ennek a geometriai értelmezése, hogy vizsgáljuk, hogy a képen látható pont távolsága az epipoláris vonaltól az adott határérték alatt van-e. Ez a határérték nagyban függ attól is, hogy a kalibráció során milyen pontosan sikerült megbecsülni a kamerák paramétereit. A továbbiakban feltételezni fogjuk, hogy a becsült paraméterek helyesek.

4. Háromszögelés

4.1 Projekciós mátrixok

A térbeli pont rekonstruálása lehetséges a pont kamerák képsíkjaira vetített képeinek és a kamerák projekciós mátrixainak ismeretében, ha a pont nem esik egybe egyik kivételes esettel sem (például a pont képei nem esnek egybe az epipólusokkal). A projekciós mátrixok kiszámíthatóak a kamerák belső és külső paramétereinek ismeretében. A projekciós mátrix segítségével kiszámíthatjuk egy térbeli pont adott kamera síkjára vetített képét:

$$x = P X$$

Ahol x a képsíkon található kétdimenziós pont, X a térbeli háromdimenziós pont, P a kamera projekciós mátrixa. A képletben X homogén koordinátájú térbeli pont, x a pont képsíkra vetített képe. A projekciós mátrixot felírhatjuk a két kamerára az alábbi módon [Har04 244. o.]:

$$P = K \begin{bmatrix} I & 0 \end{bmatrix}, \quad P' = K' \begin{bmatrix} R & t \end{bmatrix}$$

Ahol P és P' a bal és jobb kamerák projekciós mátrixai, I egy egységmátrix, t és R a két kamera közötti eltolás és elforgatás.

4.2 Problémafelvetés

A háromszögelést nehezíti, hogy koordináták, amik a rekonstrukció alapját képezik, pontatlanok. Ideális esetben a kamerák origójából a képsíkon található pontok felé indított sugarak egy közös pontban találkoznak, viszont a gyakorlatban ez a helyzet ritkán fordul elő.

Ezek alapján a célunk az, hogy meghatározzuk az X térbeli koordinátát, minél nagyobb pontossággal az x és x' pontok alapján, úgy hogy az $\hat{x}$ és $\hat{x}'$ idealizált koordinátákról nincs tudomásunk. Az $\hat{x}$ és $\hat{x}'$ koordináták az X térbeli pont tökéletes projekciói és az x és x' pontok ezen idealizált pontok zaj által befolyásolt változatai.

4.3 Lineáris háromszögelés

Az $x = PX$ egyenlet alapján felírható az alábbi egyenletrendszer [Har04 312. o.]:

$$\begin{aligned}x(p^{3T}X) - (p^{1T}X) &= 0 \\y(p^{3T}X) - (p^{2T}X) &= 0 \\x(p^{2T}X) - y(p^{1T}X) &= 0\end{aligned}$$

Ahol p^{iT} a P mátrix i -edik sorának transzponáltja ($i = 1, 2, 3$) és $x = (x, y, 1)$. Az egyenletrendszer felírható mindegyik kamerára, és az egyenletrendszerek alapján az $x = PX$ és $x' = P'X$ egyenletek kombinálhatóak $AX = 0$ alakban, ahol:

$$A = \begin{bmatrix} x p^{3T} - p^{1T} \\ y p^{3T} - p^{2T} \\ x' p'^{3T} - p'^{1T} \\ y' p'^{3T} - p'^{2T} \end{bmatrix}$$

Ebben az esetben az A mátrixban az egyenletrendszer első két egyenletét használtuk fel, mivel a három egyenlet közül csak kettő lineárisan független.

4.3.1 Homogén megoldás

Az $AX = 0$ triviális megoldása lenne $X = 0$, viszont számunkra ez a megoldás nem hasznos. Ezt az esetet kizárhatjuk egy új kritérium bevezetésével. Egyik lehetőségünk a $\|X\| = 1$ [Har04 312. o., 588. o.]. Általános esetben ennek az egyenletrendszernek nem lesz pontos megoldása, így a legkisebb négyzetek módszere (*least squares*) szerinti megoldást keressük, ahol $\|AX\|$ értékét igyekszünk minimalizálni. A megoldáshoz felhasználjuk A szinguláris érték szerinti felbontását (*singular value decomposition*, SVD):

$$A = UDV^T$$

Ahol U és V ortogonális mátrixok, D egy diagonális mátrix, ahol D főátlójában elhelyezkedő értékeket szinguláris értékeknek szokták nevezni [Fog02]. A legtöbb implementáció olyan felbontást végez ahol a szinguláris értékek csökkenő sorrendben helyezkednek el a D mátrix főátlójában [Har04 585. o.].

A keresett megoldás az A mátrix szinguláris felbontása szerinti V legkisebb szinguláris értékhez tartozó oszlopa. Ez a V mátrix utolsó oszlopát jelenti (ha a szinguláris értékek csökkenő sorrendben vannak).

[Har03] értékelése szerint ez a háromszögelési módszer a legtöbb esetben elegendően jó minőségű eredményt ad és könnyen általánosítható kettőnél több kamera esetére is. Hiányossága, hogy nem invariáns a projekciós mátrixok affin transzformációjára, holott létezik olyan háromszögelési módszer invariáns az ilyen jellegű transzformációkra [Har03 313. o.].

5. Pontkövetés

A rekonstruált térbeli pontok mozgását szeretnénk követni az időben. A feladat egyik legegyszerűbb megoldása, hogy egy ponthoz megkeressük a következő időpillanatban hozzá legközelebb eső pontot [Woo07]. Magas mintavételezési ráta esetén két időpillanat között a pont elmozdulása viszonylag kicsi lesz. Abban az esetben, ha egy adott távolságon belül nem találunk pontot, akkor pont helyzetét nem változtatjuk meg.

6. Deformálás

6.1 Virtuális arc megalkotása

6.1.1 Alap mesh

A deformálandó alak többféle módon is kialakítható, a mi esetünkben ez egy háromdimenziós pontokból (*vertex*) és az általuk megadott sokszögekből (*polygon*) álló rácsháló (*mesh*) lesz.

6.1.2 Származtatott felületek

A nagyon magas poligonszámú rácshálókat nehéz kezelni és a feldolgozásukhoz szükséges számítási teljesítmény és memóriaigény is magas. Ezért használják széles körben a származtatott felületeket. A módszer alapötlete, hogy egy kezelhető poligonszámú rácshálót alkotunk meg, és ez alapján generálunk egy sokkal magasabb felbontású, simább felületet. További előny, hogy a kész rácsháló felbontása csökkenthető, illetve növelhető igény szerint a generáló algoritmus paraméterezésével, vagy az egész lépés kihagyható. Ezáltal a célnak megfelelően beállíthatjuk a minőség és az erőforrás-igény közti egyensúlyt.

A modellező szoftverekben az organikus jellegű alakok megformálásához leggyakrabban a *subdivision surface* (továbbosztott felület) módszert alkalmazzák. A subdivision surface valójában nem egy algoritmus, hanem a mesh rekurzív felosztásán alapuló algoritmusok gyűjtőneve. A legnépszerűbb ilyen algoritmus a Catmull-Clark subdivision. [Cat78]

6.2 Animációs módszerek

Az arcok animálására többféle módszer is kialakult [Che09]. Az alakok deformálására néhány népszerű lehetőség:

- Rácspontok közvetlen mozgatása: a markerek a hozzájuk rendelt rácspontokat közvetlenül mozgatják.
- Blendshape használata: a háromdimenziós alaknak több változatát is megadjuk, mindegyik változat egy jellegzetes formát jelöl (például alaphelyzet, összehúzott szemöldök, nyitott száj, csukott szem, stb.). A markerek vezérlik, hogy a végső alakban ezek a deformációk milyen arányban vannak jelen. Az alakok között topológiai eltérés nem lehet.
- Skeletal Animation (csontváz animáció): megadjuk a forma vázát és a markerekkel a „csontokat” mozgatjuk. Arc mozgatásánál sokszor az izmokat helyettesítik ezzel a

rendszerrel és a csontok végpontjait vezérlik a markerek.

- Muscle-Based Animation (izomalapú animáció): A csontváz animáció egy fejlettebb változata, itt az izomkötegeknek kiterjedésük is van, mozgásuk során alakjuk változhat. Segítenek megőrizni a deformált alak térfogat-érzetét.

Kézi animáció esetén népszerű módszer a blendshape-ek használata. Megvalósítása technikai szempontból viszonylag egyszerű, és művészi szempontból intuitív. Használatakor eltároljuk a mesh alaphelyzetbeli alakját, és a deformációk deltáját (a mesh pontjainak elmozdulását az alaphelyzethez képest). A deltákhoz súlyokat rendelhetünk így megadhatjuk az adott blendshape hatásának mértékét, illetve a súlyok segítségével „keverhetjük” a különböző alakokat. Az így kialakult súlyokat paraméterlistaként is felfoghatjuk.

6.3 Korreláció

Az arc animálásához meg kell állapítanunk az adott markerkonfiguráció és paraméterkonfiguráció közötti viselkedésbeli összefüggéseket [Den06]. A korreláció megállapításának alapjául a $\langle M_i, W_i \rangle$ párok szolgálnak ahol M_i a markerek pontkoordinátáinak konkatenációja, W_i a paraméterek súlyaiból álló vektor és n darab ilyen párunk van ($i = 1, 2, \dots, n$). Ezek alapján a párok alapján szeretnénk meghatározni a W súlyvektort tetszőleges M -re. A feladat értelmezhető szórványos adatok interpolációjaként (*scattered data interpolation*), ahol minden paraméterre meg szeretnénk határozni egy $F_j(M)$ interpolációs függvényt amely bemeneteként a markerekhez tartozó konkatenált vektort kapja és értéke az adott paraméter súlya ($j = 1, 2, \dots, m$; ahol m a paraméterek száma).

6.3.1 Dimenziószám csökkentése

Mozgásrögzítéskor az arcra akár több tíz markert is helyezhetnek és ezáltal az M_i dimenzióinak száma nagyon magas lehet. A dimenziószám csökkentése érdekében alkalmazhatunk főkomponens analízist (*principal component analysis*, PCA). A redukció során a kellő minőségű interpoláció érdekében egyensúlyt kell teremtenünk a dimenziók száma és a rendelkezésre álló korrelációs mintapárok száma között [Den06].

6.3.2 Interpolációs függvények

A függvények felépítésére több lehetőségünk is van. Ezek közül néhány: B-spline alapú interpoláció [Lee97], radiális bázisfüggvények hálózata (*Radial Basis Function Network*) [Den06].

III. Megvalósítás

1. Hardver- és szoftverkövetelmények

Az egyik legfontosabb hardverelem a kamerapár ami a videojelek rögzítését végzi. A kameráknak több kritériumnak meg kell felelniük:

- Állandó fókusz távolság: Ahhoz, hogy a kamera kalibrációjával becsült paraméterek felhasználhatóak legyenek, a kamera paramétereinek állandónak kell lennie. Ezek közé tartozik a fókusz távolság is. Ennek érdekében a kamera fókusz távolságának rögzítettnek, vagy kézzel állíthatónak kell lennie. Rögzített esetben általában olyan optikát alkalmaznak amely képe a tipikus felhasználási távolságban és akörül viszonylag széles tartományban éles marad.
- Állítható expozíciós idő: Az expozíciós idő állításával a rendszer alkalmazkodni tud a különféle fényviszonyokhoz. A kézzel beállított expozíciós idő elengedhetetlen a stabil másodpercenkénti képkockaszám tartásához. Az alacsony expozíciós idő csökkenti a gyors mozgás okozta elmosódást, a magasabb expozíciós idő csökkenti a zaj mértékét.
- Szinkron működés (triggerelhetőség): Egyetlen kamera esetén kevésbé fontos kritérium, de kettő vagy több kamera esetén fontos, hogy a képkockák adott időpillanatokban készüljenek, több kamera esetén mindegyik kamera képe egyazon időpontban készüljön.

A projekthez két darab Logitech C160-as webkamerát használtam. A kamerák maximálisan 30 színes képkockát biztosítanak másodpercenként, VGA felbontásban. Rendelkeznek USB Video Class [USB] támogatással, ezáltal probléma nélkül használhatóak a főbb asztali operációs rendszereken.

A kamerák nagy hiányossága, hogy nem triggerelhetőek. Emiatt nincs garancia arra, hogy a kamerákból érkező képek egy időben készültek. Azonban kellően magas mintavételezési sebesség mellett feltételezhetjük, hogy a tárgy nem mozdul el nagy mértékben [Haj05]. Ebben az esetben a maximális mintavételezési sebesség 30 képkocka másodpercenként. Ha feltételezzük, hogy ezt mindkét kamera ezt stabilan és egyenletesen képes tartani, akkor a két kamera képe közötti időeltolódás maximálisan $1/30$ másodperc. (Ez az az eset amikor az egyik kép éppen frissült, a másik kép éppen frissítés előtt áll.)

A kamerák paramétereinek szoftveres beállítására Linux operációs rendszer alatt a gview helper segédprogramot alkalmaztam. A program segítségével beállítható többek között az expozíciós idő és a fehéregyensúly. Windows operációs rendszer alatt hasonló lehetőségeket nyújt a kamerákhoz mellékelt hivatalos segédprogram. A segédprogram nélküli beállítás is lehetséges Linux alatt a Video4Linux és libwebcam, Windows alatt a DirectShow programozói felületeken keresztül.

Mivel a projekt célja egy gyakorlatban is jól alkalmazható szoftver elkészítése, ezért a rendszer szorosan együttműködik egy már létező, háromdimenziós grafikai programmal, a Blenderrel [Blender]. Ez egy többplatformos, nyílt forráskódú, ingyenes alkalmazás. Funkcionalitása Python nyelven írt kiegészítőkkel bővíthető, illetve a forráskód alapján a teljes alkalmazás a személyes igények szerint átalakítható.

Bár lehetőleg többplatformos eszközöket alkalmaz, a projekt elsődleges platformja a Linux operációs rendszer. Ennek egyik oka, hogy a Blender egyik kiemelten támogatott operációs rendszere a Linux. Másik ok, hogy az implementáláshoz szükséges időt nagyban lecsökkenti, ha csak egyetlen platformra kell fejleszteni.

Az implementáció során felhasználtam az OpenCV [OCV] és Boost [Boost] szoftverkönyvtárak egyes elemeit. Az előbbi főként a kalibráció implementálásához. Az utóbbit a szálkezelés, referenciaszámláló mutatók jól kidolgozott megvalósítása és a Python kiegészítők implementálását segítő elemek miatt.

2. A rendszer szerkezete futtatási egységek szempontjából

A rendszer több egységre tagolódik futtatási egységek szempontjából. A kamerarendszer kalibrációját végző modul ezek közül az első.

2.1.1 Kalibráció

A kamerák egymáshoz viszonyított távolságának és orientációjának állandónak kell lennie annak érdekében, hogy a kalibrációval becsült paraméterek hitelesek és minél pontosabbak maradjanak. Viszont egy ilyen helyzetben a kamerarendszer paramétereit elegendő egyszer kiszámítani, és az adatok újra felhasználhatóak lesznek. Emellett a kalibráció ismételhető, és az ismételt kalibrációk eredményei közül kiválasztható a legpontosabbnak feltételezett. Az eredményképp létrejött kalibrációs paramétereket és mátrixokat a modul szöveges fájlba menti.

2.1.2 Mozgásrögzítés

A modul feladata kettős. Első feladata az adott felvétel paramétereinek beállítása (például a marker referenciaszíne), és a tényleges felvétel előkészítése (például a korrelációs paraméterek kiszámítása). Második feladata a tényleges mozgásrögzítés. Ez a modul számítja ki az aktuális paramétervektort és jeleníti meg a kamerák képeit.

2.1.3 Beépülő modul

Egy Python nyelven írt modul, ami beépül a Blender alkalmazásba. A modul hozzáfér a virtuális alak összes adatához és képes azokat módosítani, az adathalmazt új adatokkal bővíteni. Továbbá képes a felhasználói felülethez új vezérlőelemeket is hozzáadni.

A modul egy alternatív megvalósítási lehetősége, ha a Blender funkcionalitását úgy bővítjük, hogy a módosításokat közvetlenül az alkalmazás forráskódjában tesszük. Ennek előnye a teljes fejlesztői szabadság és esetlegesen a teljesítmény optimalizálásának lehetősége. Viszont a fejlesztéshez szükséges idő szempontjából előnytelen ez a megoldás. Továbbá előnytelen a végfelhasználó számára is, ugyanis ebben az esetben egy specializált változatát kell használnia a programnak, nem csak egy modult betölteni.

2.1.4 Modulok közti kommunikáció

Az utóbbi két modul közti kommunikációra több lehetőség is kínálkozik. Az egyik lehetőség, hogy a mozgásrögzítés modult a beépülő modul szoftverkönyvtárként, dinamikusan tölti be. Ebben az esetben a modulokat egy C nyelvű interfész köti össze.

Másik lehetőség, hogy a két modul hálózaton keresztül kommunikál. Az előnyök közé tartozik a jó tesztelhetőség, rugalmasság, folyamatszintű függetlenség. Cserében a módszer növeli a komplexitást az előző megoldáshoz képest.

3. A virtuális arc modellje

A projekthez szükséges fejmodellt a Blender alkalmazásban készítettem. A készítésekor figyelembe kell venni, hogy a modell animálásra készül, így szem előtt kell tartani, hogy a pontokat összekötő élek ívei nagy vonalakban kövessék az arc izomzatának íveit (Fig. 9).

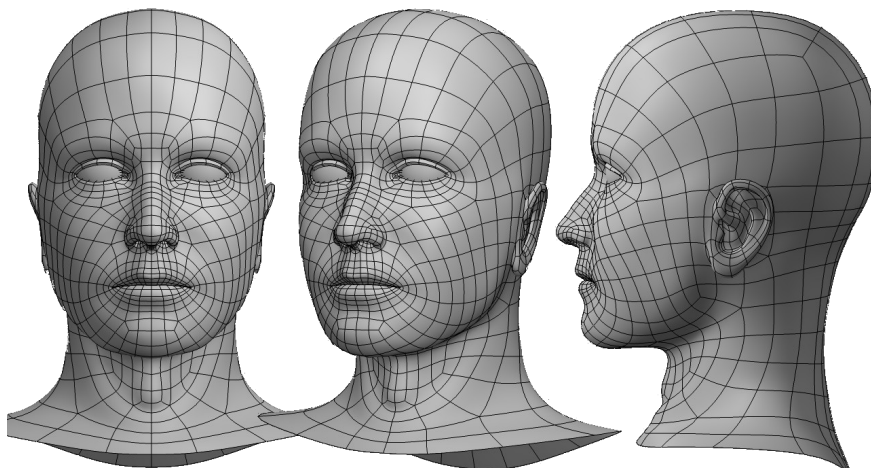


Fig. 9: A projekt számára készített fejmodell

További cél volt, hogy a kialakított topológiával többféle arcot is meg lehessen valósítani (Fig. 10), így kevésbé lehet kihasználni egy adott arc adta lehetőségeket a topológia egyszerűsítésére.

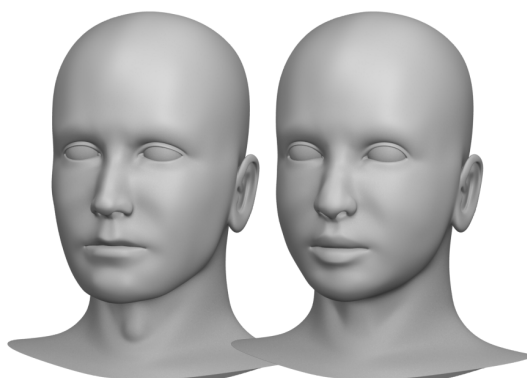


Fig. 10: Alternatív arcok azonos topológiával

Az arc deformálására a blendshape módszert alkalmaztam. A Fig. 11 képen a modell egy régebbi verziója látható. A három alak egyetlen blendshape-et mutat be különféle súlyértékekkel.



Fig. 11: Egy blendshape hatása különböző súlyértékekkel.

4. Működés

A teljes folyamat során a kamerákat oldalra döntve rögzítettem. Ezáltal megnövekedett az arc felvételére fordított pixelek aránya. A kamerák képét szoftveresen egyenes állásba forgattam. Az expozíciós időt és kamera képi beállításait a guvcview segédprogrammal állítottam be.

4.1 Kalibráció

A kalibrációhoz egy 7x6 belső sarokponttal rendelkező „sakktábla” mintát alkalmaztam, és a kalibrációt 15 képpár felhasználásával végeztem. A felhasználás jellegének megfelelően, a kalibrációs mintát kis méretűre készítettem. Két szomszédos pont közti távolság 15mm. A mintát kereső algoritmus segítése érdekében a mintát egy nagyjából 20mm széles, tisztán fehér keret veszi körül. [Bra08] egy 7x8-as tábla esetén legalább 10 képet javasol a kalibrációhoz [Bra08 388. o.]. Ha a tábla egyik irányú dimenziója páros, a másik páratlan, akkor a tábla orientációja egyértelműen megállapítható.

A kamerákat először külön-külön kalibráltam a belső paraméterek meghatározásához, majd elvégeztem a kamerák egymáshoz viszonyított kalibrációját a külső paraméterek felderítéséhez.

A kalibráció pontosságának meghatározásához a fundamentális mátrixot használtam fel. Először korrigáltam torzítások hatását mindkét kamera pontjain, majd megvizsgáltam, hogy a pontpároknál milyen mértékben teljesül a korrelációs kritérium, vagyis, hogy egy pont párja milyen messze van a ponthoz tartozó epipoláris egyenestől.

Az eredménymátrixokat és a torzítási paramétereket egy szöveges fájlba mentettem.

4.2 Mozgásrögzítés

Annak érdekében, hogy a két kamera képe között a legkisebb legyen az időbeli eltérés, a kamerák képének lekérését háttérszálon végzem. Így a feldolgozás során az algoritmus a legfrissebb képekkel dolgozik.

4.2.1 Markerek megtalálása

A feldolgozás egyik első lépése a referenciaszín kiválasztása a kameraképről. Miután ez megtörtént, minden képkockán HSV alapú globális küszöbölést végez a program. A küszöböléshez a legfontosabb feltétel a H (színárnyalat) csatorna küszöbölése, azonban ez önmagában nem elegendő, ugyanis a H csatorna akkor is felvesz egy értéket amikor a pixel színárnyalata nem meghatározható (például teljesen fekete). Emiatt a másik két csatornát is fel kell használni a küszöbölési feltétel összeállításakor. A küszöbölt képen eróziót hajtottam végre, így eltűntek a csak néhány pixelből álló hibás találatok.

A kapcsolódó komponens analízishez az előntéses kitöltés (*flood fill*) algoritmusát használtam fel. Az algoritmus végighalad a kép pixeljein és ha címkézetlen előtérpixellet talál, előtéssel megkeresi és felcímkézi az ahhoz kapcsolódó további előtérpixelet. A módszer egy menetben (*one-pass*) végzi a kapcsolódó komponens analízist és a kitöltés során megállapítja a komponens főbb paramétereit is. Ezek közül a legfontosabb a súlypont.

4.2.2 Térbeli rekonstrukció

A kameraképek pontjait a korrelációs kritérium alapján párosítottam össze. A pontpárok alapján a háromszögelést a II. fejezet 4.3. pontjában ismertetett lineáris háromszögeléssel valósítottam meg.

5. Eredmények értékelése

A különféle markerek tesztelése során sikerült találnom egy olyan markertípust, amely változatos fényviszonyok között is megállja a helyét, és a kezeléséhez a többi lehetőséggel szemben egyszerűbb algoritmusok is elegendőek. Emellett ezek az algoritmusok nem csak egyszerűbbek, de hatékonyak is. Az alacsony zajszint és jó találati arány érdekében továbbra is érdemes odafigyelni az egyenletes megvilágításra.

A projekt során sikerült megismerkednem a pontok térbeli rekonstrukciójával, amely két kamera képét használja fel. Az implementáció sok esetben kielégítő eredményt ad, a pontpárosítás és pontkövetés lépéseken érdemes további finomításokat végezni.

A virtuális arc modellezése során a kielégítő deformációkhoz elengedhetetlennek bizonyult a rácsháló él vonalvezetésének (*edge loop*) megfelelő kialakítása (Fig. 12). Erre különös figyelmet kellett fordítani. Emellett olykor meglepően nagy megszorításnak bizonyult a blendshape-ek gyakorlatilag egyetlen kritériuma, hogy a rácsháló topológiáján nem lehet változtatni. Ennek oka, hogy egyes topológiai hibákra már csak deformáláskor derül fény.

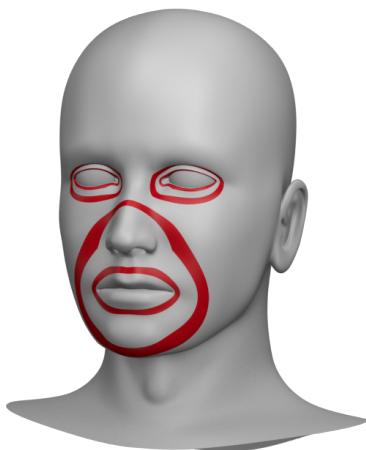


Fig. 12: Példák a topológiában felhasznált edge loop-okra

IV. Irodalomjegyzék

- [Ale09] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, P. Debevec:
Creating a Photoreal Digital Actor: The Digital Emily Project (2009)
IEEE European Conference on Visual Media Production (CVMP), 2009
<http://gl.ict.usc.edu/Research/DigitalEmily/> (Utoljára látogatva: 2010. Nov. 8.)
- [Blender] Blender 3D
<http://www.blender.org/>
- [Boost] Boost C++ Libraries
<http://www.boost.org/>
- [Bra08] Gary Bradski, Adrian Kaehler:
Learning OpenCV (2008)
O'Reilly Media, Inc., First Edition, ISBN: 978-0-596-51613-0
- [Cat78] E. Catmull, J. Clark:
Recursively generated B-spline surfaces on arbitrary topological meshes (1978)
Computer-Aided Design, Volume 10, Issue 6, November 1978, Pages 350-355
- [Che06] Chen Liu:
An analysis of the current and future state of 3D facial animation techniques and systems (2009)
School of Interactive Arts & Technology, Simon Fraser University
- [Chu02] E. Chuang, C. Bregler:
Performance Driven Facial Animation using Blendshape Interpolation (2002)
Computer Science Department, Stanford University
<http://graphics.stanford.edu/~echuang/face/> (Utoljára látogatva: 2010. Dec. 16.)
- [Fog02] Fogaras Dániel:
Singular Value Decomposition and its Applications in Information Retrieval (2002)
Technical report, 2002. (In Hungarian)
- [Den06] Zhigang Deng, Pei-Ying Chiang, Pamela Fox, Ulrich Neumann:
Animating Blendshape Faces by Cross-Mapping Motion Capture Data (2006)
In Proc. of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (2006), Pages: 43 - 48
- [Gle02] Michael Gleicher, Nicola Ferrier:
Evaluating Video-Based Motion Capture (2002)
Proceedings of Computer Animation 2002 CA, Publisher: IEEE Comput. Soc, Pages: 75-80

- [Gue98] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, Fredrick Pighin:
Making Faces (1998)
Proceedings of the 25th annual conference on Computer graphics and interactive techniques
<http://research.microsoft.com/apps/pubs/default.aspx?id=102141> (Utoljára látogatva: 2010. Nov. 8.)
- [Gue05] Gutemberg Guerra-Filho:
Optical Motion Capture: Theory and Implementation (2005)
Journal of Theoretical and Applied Informatics (RITA), Brazilian Computing Society, 12(2), 61-89.
<http://ranger.uta.edu/~guerra/publications.html> (Utoljára látogatva: 2010. Nov. 7.)
<http://ranger.uta.edu/~guerra/OpticalMotionCapture.html> (Utoljára látogatva: 2010. Nov. 7.)
- [Haj05] Hajder Levente:
Mozgásérzékelés Webkamerával (2005)
Elektrotechnika, 98. évf. 10. sz. / 2005, 273-275. o.
- [Har04] Richard Hartley, Andrew Zisserman:
Multiple View Geometry in Computer Vision (2004)
Cambridge University Press, Second Edition, March 2004, ISBN: 0521540518
- [Jon10] A. Jones, G. Fyffe, X. Yu, Ma A., J. Busch, M. Bolas, P. Debevec:
Head-Mounted Photometric Stereo for Facial Performance Capture (2010)
SIGGRAPH 2010 Talk | SIGGRAPH 2010 Emerging Technologies, July 2010
<http://gl.ict.usc.edu/Research/HeadCam/> (Utoljára látogatva: 2010. Dec. 16.)
- [Lee97] Seungyong Lee, George Wolberg, Sung Yong Shin:
Scattered Data Interpolation with Multilevel B-Splines (1997)
IEEE Transactions on Visualization and Computer Graphics, 1997, Vol. 3, Pages: 228-244
- [OCV] OpenCV
<http://opencv.willowgarage.com/wiki/>
- [USB] USB Device Class Definition for Video Devices, Revision 1.1, June 2005
http://www.usb.org/developers/devclass_docs/USB_Video_Class_1_1.zip
- [Wei09] Thibaut Weise, Hao Li, Luc Van Gool, Mark Paul:
Face/Off: Live Facial Puppetry (2009)
Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)
http://people.agg.ethz.ch/~hli/Hao_Li/Hao_Li_-_publications_Face_Off_Live_Facial_Puppetry.html (Utoljára látogatva: 2010. Nov. 8.)
- [Woo07] Alexander Woodward, Patrice Delmas, Georgy Gimel'farb, Jorge Marquez:
Low Cost Virtual Face Performance Capture Using Stereo Web Cameras (2007)
D. Mery and L. Rueda (Eds.): PSIVT 2007, LNCS 4872, pp. 763-776, 2007.
- [Zha99] Zhengyou Zhang :
Flexible Camera Calibration By Viewing a Plane From Unknown Orientations (1999)
ICCV 1999, Pages: 666-673