

Doktori (PhD) értekezés tézisei

**Ember-számítógép interakciók:
az emberi arc modellezése**

Bertók Kornél

Témavezető: Dr. Fazekas Attila



DEBRECENI EGYETEM
Informatikai Tudományok Doktori Iskola

Debrecen, 2018

Tartalomjegyzék

Bevezetés	1
1. Tézis: Az arc lokalizálása	2
1.1. Detektor tanítások eredményei.....	2
1.1.1. Tesztkörnyezet és metrikák.....	2
1.1.2. Tanító és teszt adatbázisok	2
1.1.3. Optimális paraméterek.....	3
1.2. Detektálás és követés eredményei	4
2. Tézis: Arci karakterisztikus pontok	5
2.1. Tanító és teszt adatbázisok.....	5
2.2. Tesztkörnyezet és metrikák	6
2.3. Konklúzió	6
3. Tézis: Arc elhelyezése a térben	8
3.1. Fejtartás reprezentációja és kiszámítása.....	8
3.2. Koordináta-rendszerek	9
3.3. Tesztadatbázisok és tesztkörnyezet.....	9
3.4. Konklúzió	10
4. Tézis: Fejmozgások felismerése	12
4.1. Fejmozgás reprezentációja.....	12
4.2. Felismerés folyamata	13
4.3. Fejmozgás adatbázis	14
4.4. Konklúzió	15
Introduction	18
1. Thesis: Face Localization	19
1.1. Results of Detector Training.....	19
1.1.1. Test Environment and Metrics	19
1.1.2. Training and Test Databases	19
1.1.3. Optimal Parameters	20

1.2.	Results of Detection and Tracking.....	20
2.	Thesis: Facial Feature Points.....	22
2.1.	Training and Test Databases.....	22
2.2.	Test Environment and Metrics.....	23
2.3.	Conclusion.....	23
3.	Thesis: Head Pose Estimation.....	25
3.1.	Head Pose Estimation.....	25
3.2.	Coordinate Systems	26
3.3.	Test Databases and Environment.....	26
3.4.	Conclusion.....	27
4.	Thesis: Head Movement Recognition	29
4.1.	Representation of Head Movements	29
4.2.	Process of Recognition	30
4.3.	Head Movement Database.....	31
4.4.	Conclusion.....	32
	Irodalomjegyzék.....	34
	Publikációk listája	37
	Hitelesített publikációs lista	39

Bevezetés

PhD tanulmányaim során az *ember-számítógép interakció* (*human-computer interaction*, röviden HCI) tudományterületéhez kötődő kutatásokkal foglalkoztam. Kutatásaim az ember és számítógép közötti kapcsolatok vizsgálatára és modellezésére irányultak. Elsődleges célom a kommunikációban szereplő emberi fél tanulmányozása volt, olyan módszerek gyakorlati megvalósítását tűztem ki célul, melyek alapot teremthetnek a számítógéppel való interakciók változatos fajtáinak, és amelyek természetesebbé tehetik a HCI-t az emberi fél számára. Ennek az alapját az emberi fejből és arcból kinyerhető nonverbális jelek és jellegzetességek kinyerését, reprezentálását, valamint felismerését megcélzó új képfeldolgozási módszerek jelentették. Az elsődleges célom tehát az ember-számítógép kommunikáció technológiájának továbbfejlesztése volt. Az eredményeim az alábbi téziseim köré csoportosulnak:

1. Kifejlesztettem egy *arcdetektálással, követéssel és felhasználói azonosítással* foglalkozó alrendszert – a HCI alkalmazásom alapjául.
2. Több eljárás tanulmányozásával és összehasonlító elemzésével meghatároztam egy 2-D pixeltérbeli reprezentációt az *arc karakterisztikus pontjaira* vonatkozóan.
3. Nagy pontossággal meghatároztam az *arc térbeli elhelyezkedését és irányultságát*, a karakterisztikus pontok 2-D pixeltérbeli helyzetének-, és az arc 3-D geometriájának ismeretében.
4. Az elért eredményeimre támaszkodva létrehoztam egy eljárást változatos *fejmozgások tér-, és időbeli reprezentálására, valamint felismerésére*.

1. Tézis: Az arc lokalizálása

PhD tanulmányaim során létrehoztam egy arcdetektálással, követéssel és felhasználói azonosítással foglalkozó alrendszert. A feladat magában foglalta a szükséges tanító-, és teszt adatbázisok összeállítását; az eredmények méréséhez szükséges tesztkörnyezet kialakítását az eredmények összehasonlítására; illetve a képfeldolgozási algoritmusok áttekintését [P1, P2].

1.1. Detektor tanítások eredményei

Az általam használt arcdetektor Viola és Jones munkásságán alapul [1]. Ugyan léteznek szabadon elérhető arcdetektorok, de jónak láttam saját arcdetektorok tanítását, mivel a detektor elkészítése számos paraméter mentén konfigurálható.

1.1.1. Tesztkörnyezet és metrikák

A tesztkörnyezet kialakítása során két szempontot vettem figyelembe: a futásidőt és a pontosságot. A futásidő egy jól mérhető mennyiség (milliszekundumban). A pontosság kapcsán két esetet vizsgáltam: azt amikor a detektor arcot talált, ott, ahol nem volt (*hamis pozitív*), attól az esettől, amikor nem detektált egy létező arcot (*hamis negatív*). Ezekből származtattam a *felidézés*, *precízió* és F_1 mérőszámokat a tanított osztályozóim jóságának számszerűsítésére.

1.1.2. Tanító és teszt adatbázisok

A tanítást megelőző lépésben létrehoztam egy reprezentatív tanító adatbázist. Az adatbázis képeit két diszjunkt halmazba soroltam: pozitív-, és negatív minták halmazába. A pozitív minták kivágott arcokat ábrázoltak, még a negatívok arcot nem tartalmazó képek voltak. A tanításokhoz két pozitív adatbázist hoztam létre a [2, 3, 4] adatbázisokból, melyek mindegyike 6500

db mintát tartalmazott. Az egyik pozitív adatbázisba csak frontális arcokat vettem fel, még a másikba a frontális orientáció mellett $\pm 45^\circ$ -ban a függőleges tengely körül elforgatott arcokat is. A negatív adatbázist tekintve 15000 db mintát gyűjtöttem egy internetes keresőrobot segítségével [5]. Az elkészült osztályozók tesztelésére, az előbb hivatkozott keresőrobotot felhasználva hoztam létre két pozitív-, és egy negatív adatbázist 1000-1000 db kép felhasználásával.

1.1.3. Optimális paraméterek

1. A *jellemzőkinyerő algoritmusok* tekintetében az alábbi következtetésekre jutottam:
 - Ha a végfelhasználásra szánt környezetben nagyrészt csak frontális arcok fordulnak elő, akkor célszerű Haar-féle jellemzőt [6] és frontális arcokat választani a tanításhoz,
 - Ellenkező esetben célravezetőbb az LBP-t [7] választani és az elforgatott arcokkal kiegészített adatbázisra tanítani.
2. A *pozitív tanítóminták felbontásának* hatását három különböző pixel méret mellett teszteltem. A 24×24 -es felbontást választottam, mellyel megfelelő pontosságot értem el úgy, hogy nem növekedett túlságosan a detektálás futási ideje.
3. A *kaszkádk struktúra szintjeit* tekintve arra a következtetésre jutottam, hogy a kaszkád struktúrát célszerű 18-20 boosted osztályozóból összeállítani.
4. A *boosting algoritmus különböző változatainak* arra a következtetésre jutottam, hogy detektor teljesítménye nem igazán függ a boosting algoritmus típusától, de célszerű a gentle AdaBoost [8] használata.

1.2. Detektálás és követés eredményei

Az optimális paraméterek mellett tanított detektorral ugyan nagy pontossággal detektáltam az arcokat, de a futásideje nem tette lehetővé valósídejű HCI alkalmazások létrehozását. A gyakorlatban egy követő eljárással kombináltam a detektálást. Az detektort adott frekvenciával futtattam a bemenő képsorozaton. Két detektálás között pedig egy követő eljárással jósoltam meg az arc pozícióját. A tervezett HCI rendszerem konfigurációja lehetővé tette egy egyszerű, de hatékony követési algoritmus adaptálását. A tesztek megelőzően annotáltam az arc pozícióját egy közel 500 képkockából álló videó folyamon, melyen változatos fejmozgásokat végeztem. Meghatároztam egy hasonlósági mértéket a detektor/követés kimenete és az annotált arcpozíció között. Az alábbi következtetésekre jutottam:

1. Mind a detektor-, mind a követés által meghatározott befoglaló téglalap több, mint 70%-os átfedésben volt az annotált téglalappal.
2. A detektor felidézésének értéke 0.97, ami a gyakorlatban azt jelentette, hogy 13 db képkockán nem talált arcot. A követés esetén a felidézés értéke 1.0, azaz az eljárás végig sikeresen le tudta követni az arcot.
3. A futási időket tekintve a követés egy nagyságrenddel gyorsabb volt, mint a detektálás: 15 ms alatti időket mértem.
4. A követő eljárást felhasználva felhasználói azonosságokat rendeltem a bemenő képen látható arcokhoz. Ezek metaadatok formájában jelentek meg, mint pl. az utolsó detektálás ideje; az aktuális lokációja-, vagy textúrája az arcnak.

2. Tézis: Arci karakterisztikus pontok

PhD tanulmányaim során, meglévő módszerek tanulmányozása és felhasználása által létrehoztam egy alrendszer az arc karakterisztikus pontjainak reprezentálására. A feladat magában foglalta a szükséges tanító-, és teszt adatbázisok összeállítását; valamint az eredmények méréséhez és összehasonlításához szükséges tesztkörnyezet kialakítását [P1, P2, P6].

Az arc legfontosabb alkotóelemeit, mint például a szemeket, a szájat, az orrot a nevezzük karakterisztikus pontoknak. A karakterisztikus pontok közvetlenül vesznek részt a mimika, az arcmozgás vagy akár az érzelmek kifejezésében. A pontok helyzetéből, mozgásából, és egyéb származtatott mennyiségeiből további következtetéseket tudtam levonni a HCI rendszeremben. A gyakorlatban három különböző statisztikai modell alapú eljárást vizsgáltam meg és választottam ki azt, amelyik a legjobban igazodott az elvárásaimhoz: *active shape models* (ASM) [9], *active appearance models* (AAM) [10], és *constrained local models* (CLM) [11]. A módszerek mindegyike az arc körvonalának geometriai tulajdonságait, illetve a körvonal körüli textúrát modellezi különböző statisztikai eljárások segítségével.

2.1. Tanító és teszt adatbázisok

A tanítás és a tesztelés során az arc detektálás fejezetben felhasznált adatbázisokból válogattam ki képeket [2, 3, 4]. Az alakzatmodellnek a széleskörben elterjedt CMU Multi-PIE adatbázis elrendezését választottam, mely 68 db karakterisztikus pontot tartalmazott [12]. A három módszert ugyanarra a 600 db képre tanítottam. 100 emberről válogattam ki 2 db frontális-, és 2-2 db a fej függőleges tengelye körül $\pm 45^\circ$ -ban elforgatott képet.

A tesztadatbázist hasonló módon állítottam össze, mint a tanítóadatbázist, azonban olyan személyekről szerepeltettem benne képeket, akik a tanítóadatbázisban nem szerepeltek. A három módszert ugyanarra 300 db képre teszteltem. 50 emberről válogattam ki 2 db frontális-, és 2-2 db a fej függőleges tengelye körül $\pm 45^\circ$ -ban elforgatott képet. A kimerítő tesztelés érdekében egy spontán beszédhez tartozó videó felvételre is lefuttattam a három módszert. A felvétel 3,5 perc hosszú volt és 5000 darab képkockából állt. Az annotációs formátuma [13] eltérő volt az enyémhez képest, azonban 4-4 db pont a két szemem, 2-2 db a két szemöldökön, kettő az orron, négy a szájon, illetve három az arc körvonalán megfigyelhető volt az általam használt formátummal.

2.2. Tesztkörnyezet és metrikák

A modellillesztés pontosságát a *root-mean-square* (RMS) hiba kiszámításával mértem – az annotált jellemzőpontok és illesztett modell alakzatpontjai között. Külön mértem az RMS hibát a frontális orientációjú-, és a $\pm 45^\circ$ -ban elforgatott arcokra. Ezzel meg tudtam határozni a modell forgatás invarianciáját. A kiértékelés során aggregáltam az adatbázisokra kapott eredményeket aszerint, hogy az alakzatmodell pontjai mely arcrész modellezésében vesznek részt. Összesítettem a bal szemet, jobb szemet, bal szemöldököt, jobb szemöldököt, orrot és szájat alkotó határpontok RMS hibáit, és az így kapott eredményeket vettem össze egymással.

2.3. Konklúzió

Frontális arcokra tesztelve a modell illesztés pontosságát nem tapasztaltam túl nagy eltérést a három módszer között. Az adatokból az látszott, hogy a szemöldökök hibája közel kétszer

akkora, mint az összes többi arci jellemzőé. Rosszabb eredményekkel szembesültem akkor, amikor $\pm 45^\circ$ -ban elforgatott orientációjú arcokra teszteltem. Az AAM lényegében megbukott a tesztek során: közel 2,5x rosszabb eredményeket produkált, mint frontális esetben. Ezzel szemben, a másik két módszernél csak másfélszeres romlás volt megfigyelhető. Az AAM kapcsán lehetett volna újabb modelltanítással, esetleg másik implementációval kísérletezni, de a másik két implementáció hatékonysága miatt ezeket az ötleteket elvettem. Az esetek közel 80%-ban a CLM jobb eredményt ért el, mint az ASM, de az eltérés nem volt szignifikáns. Így a futásidő lett az a faktor, ami alapján a két módszer közül választottam.

A futásidőt a videófolyamon teszteltem alaposabban. Az ASM illesztés futásidejére fele annyi időt mértem (30 ms), mint a CLM-re (60 ms). A gyakorlati alkalmazásban az illesztést tovább optimalizáltam, úgy, hogy a bemenő képszekvencia $(k + 1)$ -ik elemére az $\bar{s}$ átlagalakzat helyett, k -ik képre illesztett s_k alakzatmodellt feszítettem ki – így az algoritmus gyorsabban konvergált. Az illesztés pontosságát tekintve, az ASM hibája átlagosan 6 pixel volt. A gyakorlatban 2,24 pixellel pontosabb illesztést végzett, mint a CLM a videó felvétel teljes hossza alatt. A különbség nem szignifikáns, de a tesztek során úgy találtam, hogy az ASM illesztése stabilabb, mint a CLM-é. Ezalatt azt értem, hogy az illesztése hibája kevésbé ugrált a bemenő képszekvencia szomszédos képkockái között. Mindent összevetve, úgy találtam, hogy az ASM pontossága és gyorsasága elegendő lesz ahhoz, hogy bonyolultabb HCI alkalmazásokat hozzak létre.

3. Tézis: Arc elhelyezése a térben

PhD tanulmányaim során, az eredményeimre támaszkodva létrehoztam egy rendszert, mely képfeldolgozási algoritmusokat felhasználva határozza meg a HCI rendszeremben az emberi fél fejtartását 2-D képeken. A feladat magában foglalta a fejtartás reprezentációját és kiszámítását; a szükséges teszt adatbázisok összeállítását; valamint az eredmények méréséhez szükséges tesztkörnyezet kialakítását [P1, P2, P4, P6, P7, P8, P10].

3.1. Fejtartás reprezentációja és kiszámítása

Fejtartás alatt a fej térbeli pozíciójának és orientációjának az együttesét értem. A számítógép a fejtartás ismeretében kaphat képet a kommunikációs partnere figyelmének középpontjáról. A fejtartást egy hat szabadsági fokkal (6DoF) rendelkező modell segítségével határoztam meg a 3-D euklideszi térben. A reprezentáció tartalmazta a fej térbeli pozícióját leíró 3-D vektort, mely leírta, hogy hol helyezkedik el a világ koordináta-rendszerben a fej. Továbbá tartalmazta a fej térbeli orientációját is, amely megadta, hogy a világ koordináta-rendszerben milyen irányban áll a fej egy rögzített koordináta-rendszerhez képest.

Az általam kidolgozott módszer 3-D pontok és azok 2-D vetülete alapján határozta meg a fejtartást. A 2-D pontok a kamera koordináta-rendszerben elhelyezkedő képsík felett voltak értelmezve: az előző fejezetben ismertetett alakzatmodell határpontjai voltak. A 3-D pontok egy modell koordináta-rendszerben voltak definiálva és a 2-D pontok által kijelölt alakzathoz hasonló geometriájú 3-D modell kontrollpontjaihoz tartoztak. A 3-D modell pontokat egy sztereó képadatbázist [14] felhasználva állítottam elő emberi arcokról készült mélységtérképek alapján. A 3-D és 2-D pontokra támaszkodva

egy *perspective n-point* (röviden PnP) probléma megoldásaként definiáltam a fejtartást. A PnP probléma megoldását a POSIT [15] eljárás segítségével határoztam meg, mely legalább négy darab, nem egy síkba eső pontpárt felhasználva, skálázott ortografikus vetítéssel közelítette a perspektív vetítést és határozta meg a fej térbeli pozícióját és orientációját.

3.2. Koordináta-rendszerek

Egyrészt szükségem volt a *kamera koordináta-rendszerre*, mely a z tengely pozitív irányába nézett; a kép sorai az x tengellyel párhuzamosan, az oszlopai pedig az y tengellyel párhuzamosan álltak. Másrészt definiáltam egy *3-D modell koordináta-rendszert* is, melyben a 3-D modell pontokat vettem fel. A modell koordináta-rendszer állása megegyezett a kamera koordináta-rendszerével. Ez azért volt szükséges, mert a POSIT a kamerához képest adta meg a modell helyzetét.

3.3. Tesztadatbázisok és tesztkörnyezet

A tesztjeim során kiemelt figyelmet fordítottam a fej x (*pitch*) és y (*yaw*) tengely körüli elforgatásaira, mivel a rendszerem magasabb szintű komponenseiben főként ezeket az értékeket használtam fel. A fejtartás meghatározást két adatbázison teszteltem. A tesztek egy részét a Biwi adatbázison [16] futtattam le, melyet egy Kinect mozgásérzékelő szenzor segítségével rögzítettek a szerzők. Az adatbázis 20 emberről tartalmazott több mint 15000 mérést, ahol a fejtartás a $\pm 75^\circ$ yaw és $\pm 60^\circ$ pitch tartományban mozgott. Az adatbázis annotációját automatikusan végezték el, így nem volt információm az annotációs adatok pontosságáról.

A módszerem pontosságát teszteltem továbbá a GI4E adatbázison [17]. Az adatbázis 10 emberről tartalmazott több mint 36000 mérést, ahol a fejtartás a $\pm 30^\circ$ yaw és $\pm 30^\circ$ pitch tartományban mozgott. Ebben az adatbázisban sokkal több annotációs adatot tettek közzé a szerzők, pl. egy 54 pontból álló alakzatmodell 2-D és 3-D ground truth koordinátáit, valamint a valós fejtartást. Ez utóbbit egy erre a feladatra dedikált szenzor segítségével határozták meg. A szerzők mérése alapján 0.83° roll, 0.86° yaw és 1.05° pitch pontosság feltételezhető a fejtartást tekintve.

3.4. Konklúzió

Mivel a Biwi adatbázis relatív nagy tartományt fedett le a pitch és a yaw tekintetében, így a tesztjeim során fejtartás meghatározás pontosságára fókuszáltam az intervallumok teljes szélességében. Yaw esetén a $\pm 40^\circ$ szögtartomány volt az, amikor stabilan 10° fok alatt volt az orientáció abszolút hibája, a minimumát (5° alatti hibát) pedig a $\pm 12^\circ$ tartományban érte el. A szélső értékekhez közeledve exponenciálisan emelkedett a hiba, amiben komoly szerepet játszhatott az ASM modell-illesztés is. Megjegyzem, hogy a $\pm 45^\circ$ -ban elforgatott arcokra 1,5x nagyobb volt az ASM modellillesztés hibája, mint a tisztán frontális arcokra.

Pitch esetén a $[-32^\circ, +40^\circ]$ szögtartomány volt az, amikor stabilan 10° fok alatt volt az orientáció abszolút hibája, a minimumát pedig a $[-6^\circ, +16^\circ]$ tartományban érte el. A hibafüggvény karakterisztikája nem volt szimmetrikus a 0° -ra, ami azt jelentette, hogy a felfelé néző emberekre nagyobb hibával dolgozott az algoritmus. Mivel a kamera pont szemből nézte a tesztalanyokat, így a jelenség feltárásához további tesztek

elvégzését tartottam szükségesnek. Tesztjeimből kiderült, hogy a vízszintes tengely körüli fejmozgás hibája korrelál a fej függőleges tengelye körüli elforgatásával: a balra-lent és balra-fent irányokban mértem a legnagyobb hibákat. Ez sajnos fakadhatott az adatbázis jellegéből is.

A GI4E adatbázis kisebb tartományt fedett le a pitch és yaw tekintetében, de a segítségével több hasznos tesztet is le tudtam futtatni. Az eredményekből látszott, hogy a pitch itt is pontatlanabb, mint a yaw, de a két függvény karakterisztikája jobban megfelelt egymásnak. A függőleges tengely körüli fejmozgás hibája a $\pm 25^\circ$ tartományban volt 5° alatt, ami sokkal jobb eredmény, mint amit a Biwi esetén mértem. Ezzel szemben az oldalirányú fejmozgás egy szűkebb intervallumban volt 5° alatt, a $\pm 15^\circ$ tartományban – de ez is meglehetősen jó eredménynek számított.

Az adatbázis szerzői a metaadatok között publikálták egy 54 darab pontból álló alakzatmodell valós 2-D és 3-D koordinátáit, mellyel ki tudtam mérni tisztán csak a fejtartás meghatározás hibáját. A ground truth mérésekre számolt abszolút hiba a teljes intervallum hosszában egyenletes, volt: yaw esetén $1,45^\circ$, pitch esetén pedig $1,77^\circ$. Teszteltem továbbá a saját 3-D modellem visszavetítésének hibáját a képsíkra – a szerzők által publikált fejtartás függvényében. A visszavetítés hibája kellően egyenletes volt, 2-6 pixel között mozgott az intervallum teljes hosszában. A módszerem pontossága nem maradt el a szakirodalomban létező megoldásokétól [18], és sikeresen fel tudtam használni a további kutatásaim során.

4. Tézis: Fejmozgások felismerése

PhD tanulmányaim során létrehoztam egy rendszert, mely az arcdetektálás, a karakterisztikus pontok kinyerése és a fejtartás meghatározása során elért eredményeinkre támaszkodva ismert fel komplex fejmozgásokat. A feladat magában foglalta a fejtartás tér-, és időbeli reprezentációját, valamint felismerését; egy fejmozgás adatbázis elkészítését; továbbá az eredmények méréséhez szükséges tesztkörnyezet kialakítását [P1, P2, P5, P9].

A fejmozgások felismerése emberi tevékenységek felismerését jelenti néhány előzetes megfigyelés és mérés alapján. Az angol szakirodalom ezt a folyamatot *human activity recognition*-ként (röviden HAR) hivatkozza. A felismeréshez tanulmányoztam a fejmozgás jellemzőit: egy-egy fejmozgás tipikus hossza 3-5 másodperc, de nem kívántam korlátozni a lehetséges mozgások időbeli hosszát. A felhasználtól nem vártam el azt sem, hogy tudatában legyen annak, hogy egy-egy mozgást pontosan hogyan és mennyi idő alatt kell elvégezni. Ez a gyakorlatban azt jelentette, hogy két fejmozgás sosem volt teljesen megegyező.

4.1. Fejmozgás reprezentációja

A reprezentáció definiálásakor arra törekedtem, hogy változatos és komplex fejmozgásokat tudjak felismerni valós időben. A fejtartásból nyertem ki bizonyos jellemzőket, majd határoztam meg a fejmozgás trajektóriáját. Természetesen a fejmozgásokat időben is be kellett határolni. A fejmozgás kezdetét és végét is a fejtartásból határoztam meg. Az általam javasolt reprezentáció egységbe foglalta a térbeli-, és időbeli dimenzióit a fejmozgásoknak.

A fejmozgásokat, mint idősorokat definiáltam, melyek egy adott attribútum időbeli megfigyeléseiből épültek fel. A térbeli reprezentációt tekintve a fej orientációjának változását vettem figyelembe a bemenő képszekvencia szomszédos elemei között. A hibás fejtartások és a lokális zajok kiszűrésének érdekében egy Kálmán-szűrőt alkalmaztam a fejtartás „nyers” kimenetére. A képszekvencián látható fejmozgáshoz tehát egy $s \in \mathcal{S}$ idősort, mint statisztikai mintát rendeltem, mely az $a_\tau, a_{\tau+1}, \dots, a_{\tau+n}$ megfigyelésekből állt. Az egyes a_i megfigyelések $\mathcal{F}$ -fel jelölt jellemzőtér elemei voltak. Az $m \in \mathcal{M}$ fejmozgásokat az alábbi formában definiáltam:

$$m = (l, a_\tau, a_{\tau+1}, \dots, a_{\tau+n}) = (l, s), \quad (4.1)$$

ahol $l \in L$ az elvégzett tevékenység címkéje (pl. bólintás, vagy fejrázás). Az attribútumok meghatározása során a 3-D fejmodell koordináta-rendszerének z irányú bázisvektorából indultam ki: $\underline{k} = (0,0,1)$. Forgassuk el $\underline{k}$ -t a pillanatnyi fejtartásnak megfelelően és az eredményül kapott vektort jelöljük $\underline{k}_i$ -vel. Az a_i attribútumokat az aktuális-, és a megelőző képkockára számolt irányvektorok különbségeként definiáltam:

$$a_i = \underline{k}_i - \underline{k}_{i-1}. \quad (4.2)$$

Tehát a_i a fej orientációjának a megváltozását jelentette két képkocka között. Az egyes a_i méréseket eltoltam a_τ -val, hogy a mérések függetlenek legyenek a kezdő orientációtól.

4.2. Felismerés folyamata

A fejmozgásokhoz tartozó idősorok egymáshoz képest eltérőek lehetnek az idő vagy sebesség függvényében, azaz csak nem-lineárisan illeszthetők egymáshoz az idő dimenzió mentén.

Az illesztést egy dinamikus idővetemítés (DTW) alapú módszerrel valósítottam meg [19], mely két eltérő ütemben kivitelezett idősor lehető legjobb illeszkedését határozza meg az idő tengely transzformálásával. Legyen ez a két idősor:

$$\begin{aligned} X &:= (x_1, x_2, \dots, x_N) \\ Y &:= (y_1, y_2, \dots, y_M) \end{aligned} \quad (4.3)$$

A DTW, az x_i és y_j attribútumokat próbálja megfeleltetni egymásnak. Egy $C \in \mathbb{R}^{N \times M}$ költségmátrixot épít fel egy távolságfüggvény felhasználásával és határozza meg a mátrix minimális költségű útját, mely egyben a két idősor közötti optimális illesztése is.

4.3. Fejmozgás adatbázis

A DTW két idősor optimális illesztését valósítja meg, a gyakorlatban viszont ennél több fejmozgást akartam felismerni, amit egy DTW-n alapuló osztályozó elkészítésével oldottam meg. A HAR rendszer alapos vizsgálatához egy 26 osztályos osztályozót készítettem. Az osztályokat jelöljük $M_i \subset \mathcal{M}$ -vel, ahol $\mathcal{M}$ a fejmozgások halmaza:

$$M = \{M_i \subset \mathcal{M} \}_{i \in [1:26]}. \quad (4.4)$$

Az M_i osztályok az angol ABC fejjel leírt betűinek trajektóriái voltak, így olyan fejmozgásokat tartalmaznak, mint pl. egy p , vagy h betű fejjel történő leírása. Az osztályozó a fejmozgások között fennálló térbeli rendezettség (DTW távolság) és egy fejmozgás adatbázison alapult. A DTW sarkalatos pontja, hogy két fejmozgás között ki kell számítani a $C \in \mathbb{R}^{N \times M}$ költségmátrixot, ami a gyakorlatban meglehetősen számításigényes. A felismerés folyamatát megpróbáltam annyira leegyszerűsíteni,

amennyire az csak lehetséges. Bizonyos elvek mentén lecsökkentettem az M_i osztályok számosságát 1-re, majd egy 1-legközelebbi szomszéd (1-NN) osztályozót hoztam létre.

A meglévő fejmozgásokból választottam ki minden osztályból pontosan egyet: azt, amelyik a legjobban reprezentálta az adott osztály összes elemét. Jelöljük ezt az elemet m_i^r -vel, melynek a meghatározásához egy fejmozgás adatbázist állítottam össze. Az adatbázis összeállítása során a mintákat 10 különböző embertől gyűjtöttem össze, mindenki 20 darab fejmozgást végzett osztályonként, így 200 db fejmozgást kaptam minden egyes betűre. Osztályonként 100 darab fejmozgást használtam fel arra, hogy meghatározzam az m_i^r fejmozgást. A maradék 100-100 db fejmozgást kizárólag tesztelési célokra használtam fel. Az m_i^r elemtől azt vártam, hogy a legjobban szeparálja a hozzá tartozó M_i osztályt a többi osztálytól. Az m_i^r elem meghatározása során kiszámoltam a saját M_i osztályán belül az összes lehetséges elempár DTW távolságát, majd azt az elemet választottam m_i^r -nek, amelyik minimalizálta az osztályon belüli távolságot.

4.4. Konklúzió

A tesztelés első lépésében megvizsgáltam az egyes m_i^r elemek átlagos távolságát (μ_i) és korrigált tapasztalati szórását (σ_i) a többi elemhez képest az M_i osztályokon belül. Ebből azonban csak az osztályok térbeli kiterjedésére vonatkozóan kaptam visszacsatolást, a felismerés pontosságára nem. Azt tapasztaltam, hogy az e , vagy v betűk esetén a klaszterközéppontok nagyon közel helyezkednek el a többi ponthoz, viszont például a t , vagy q betűk esetén sokkal nagyobb DTW távolságokat kaptam a

többi elemhez képest – ezeknek az osztályoknak az elemei közel háromszor akkora sugarú körben helyezkedtek el, mint az előző esetben. További vizsgálataim során azt tapasztaltam, hogy a μ_i és σ_i értékek az M_i osztályon belüli idősorok átlagos hosszával korrelálnak. A magas μ_i és σ_i értékek a DTW jellegéből fakadtak. Azaz nagyobb μ_i és σ_i értékek várhatók olyan osztályok esetén, melyek hosszabb idősorokat tartalmaznak, de korántsem biztos, hogy a felismerés pontossága rosszabb lesz ezen osztályok esetén.

További vizsgálataim az az első-, és másodfajú hiba mérésére vonatkoztak a felidézés és precizitás mérőszámok meghatározása által. A 26 darab osztályra számolt precizitás átlagos értéke 0,81 és az átlagos felidézés pedig 0,83 volt, amelyek garanciát jelentettek mind a hamis pozitívok, mind a hamis negatívok alacsony darabszámára a felismerés során. Látható volt azonban, hogy az a , k , l , q , t és y osztályok esetén relatív sok hamis pozitív találatra kell számítani, ezeknél az osztályoknál a precizitás értéke 0,56. Ugyanakkor sok hamis negatív találatra kell számítani az a , d , e és h osztályok esetén, ezeknél a felidézés értéke 0,63. Meghatároztam továbbá az F_1 metrikát is az osztályokra. Az F_1 mérőszám alapján a 10 legkisebb hibával felismerhető fejmozgásra az r , u , j , i , w , f , v , s , x és z osztályok esetén számíthattunk. Külön érdekesség, hogy a z osztály esetén mind a 100 db fejmozgás helyesen lett felismerve. Ellenben a 10 legnagyobb hibával felismerhető fejmozgásra az a , q , l , t , d , h , n , k , m és y osztályok esetén számíthattunk. A legnagyobb hibával felismerhető fejmozgásoknál a probléma okát a más osztályok mozgásformáival való hasonlóságokban kell keresni. Pl. a betűt

hasonlóan írjuk, mint d -t, de ugyanígy hasonlóság mutatkozik pl. az $m - n$, vagy $l - e$ betűk esetén. Megvizsgáltam továbbá, hogy van-e korreláció a fejmozgások átlagos hossza és az F_1 pontszámuk között. Kijelenthető, hogy nem függ össze a két mérés: rövidebb minták felismerése is történhet magas pontossággal (pl. i , v és z), illetve hosszabbaké is alacsonyabb pontossággal (pl. q , h és m) és vice versa. Mindent egybevetve azt tapasztaltam, hogy a rendszer pontossága az osztályok felére 80%-on felüli (mind a precizitást, mind a felidézést tekintve), ami mindenféleképpen jó eredménynek számít.

Introduction

During my PhD studies, I elaborated on some aspects of *human-computer interactions* (hereinafter HCI). I focused on examining and modeling human-computer relationships. My primary goal was to study the human party that is being involved in the communication. I aimed to develop such methods that could provide a basis for the multiple types of interactions in order to make HCI more natural to the human party. The focal point of my research was the development of new image processing methods based on the information provided by the human head and face. It covered the extraction, representation and recognition of nonverbal signals and signs. I focused on the improvements of the HCI's technology; my results were grouped around the following theses:

1. I developed a subsystem as the basis of my HCI system for *face detection, face tracking and user identification*.
2. I determined a *2-D spatial representation of facial feature points* by studying and analyzing of several approaches.
3. I determined the *position and orientation of the human head* based on the 2-D location facial feature points on the image plane and the corresponding 3-D geometry of the head.
4. Based on my achievements, I developed an approach for the spatio-temporal *representation and recognition of varying head movements*.

1. Thesis: Face Localization

During my PhD studies, I developed a subsystem as the basis of my HCI application for face detection, tracking and user identification. The task included the preparation of the necessary training and test databases; the development of a test environment for measuring and comparing results; as well as an overview of image processing algorithms [P1, P2].

1.1. Results of Detector Training

The face detector was based on the work of Viola and Jones [1]. While there are freely available face detectors, but I considered it useful to train my own ones because the detectors can be configured along several parameters.

1.1.1. Test Environment and Metrics

I considered two aspects in the establishment of the test environment: the runtime and accuracy. The runtime is an easily measurable amount (in milliseconds). Furthermore, I considered the following cases in case of accuracy: on the one hand, the case when the detector found a face where it was not any (*false positive*); and on the other hand, the case when it did not detect an existing face (*false negative*). From these, I derived the following measures in order to quantify the efficiency of my detectors: *precision*, *recall* and F_1 score.

1.1.2. Training and Test Databases

I created a representative training database from several images. The images were divided into two disjoint sets: positive and negative set of samples. The positive samples were cropped faces, and the negatives were non-face images. I created two positive databases for the trainings from the databases [2, 3, 4]

containing 6,500 samples each of them. One of the positive databases contained only frontal faces, and the other one contained faces that were rotated in $\pm 45^\circ$ around the vertical axis. I also collected 15,000 samples for the negative database using a web crawler [5]. Using the previously mentioned crawler, I created two positive-, and a negative test database with 1000-1000 images.

1.1.3. Optimal Parameters

1. Regarding *feature extraction methods* I came to the following conclusions:
 - If mostly frontal faces can be occurred in the end-use environment, then it is recommended to choose Haar features [6] and frontal faces for the training,
 - Otherwise, it is more useful to train with local binary patterns [7] on the database with rotated faces.
2. Regarding the *resolution of positive training samples*, I chose the 24×24 size. It made possible to get high accuracy with relatively low runtime of the detection.
3. Concerning the *stages of cascade hierarchy*, I found that it should be constructed from 18-20 boosted classifiers.
4. Regarding the *versions of the boosting algorithm*, I found that the performance of a detector does not really depend on the boosting algorithm, but it might be recommended to use Gentle AdaBoost [8].

1.2. Results of Detection and Tracking

I was able to detect faces beside high accuracy with the detector trained along the optimal parameters. However, its runtime did not really allow me to develop real-time HCI applications. I combined the detection with tracking in practice. I run the face

detector with a given frequency over the element of an image sequence, and I tried to predict the face position between two detections with a tracking procedure. The configuration of my HCI system made it possible to adapt a simple but efficient template-based tracking algorithm. I annotated the face on a video stream with almost 500 frames where I performed various head movements. I determined the degree of similarity between the output of the detector/tracking and the annotated face position and came to the following conclusion:

1. Both the detector and tracking algorithm were more than 70% overlapped with the annotated rectangle.
2. The detector's recall was 0.97, which means in practice that my face was not detected only on 13 frames overall. In case of tracking, the recall was 1.0 that is the procedure was able to follow my face completely.
3. Regarding the runtimes, the tracking was faster with a magnitude than the detection: I measured less than 15 ms runtime for it.
4. Using the face tracking method, I assigned user identities for the faces on the input image. These were essentially a set of metadata, e.g. the last time of detection, the current location or texture of the face.

2. Thesis: Facial Feature Points

During my PhD studies, I developed a subsystem by studying and using existing methods for the representation of facial feature points. The task included the preparation of the necessary training and test databases; and the development of a test environment for measuring and comparing results [P1, P2, P6].

Facial feature points – such as certain points of the eyes, mouth or nose – are the most important components of the face. These points are directly involved in the expression of mimics, head movements or even emotions. I was able to get several consequences from the knowledge of location, movement and other derived quantities of these feature points. I analyzed three different approaches in practice and selected that one that fitted the best to my expectations: *active shape models* (ASM) [9], *active appearance models* (AAM) [10] and *constrained local models* (CLM) [11]. Each of these methods examines the geometric properties of the face contour and the texture around the contour using various statistical methods.

2.1. Training and Test Databases

I selected several images from the [2, 3, 4] databases used in the chapter face detection for training and testing the different approaches. For the configuration of the shape model, I chose the format of CMU Multi PIE database, which contained 68 feature points [12]. I trained the three methods on the same 600 pictures from 100 people. I selected two frontals-, and 2-2 rotated ones in $\pm 45^\circ$ around the vertical axis of the head.

I composed the test database in a similar way to the training database, with the exception that I included images only from

people who did not appear in the training database. I tested the three methods on the same 300 pictures from 50 people. I selected 50 frontals-, and 2-2 rotated ones in $\pm 45^\circ$ around the vertical axis of the head. For testing purposes, I also applied the three different approaches on a video stream that belonged to a spontaneous speech. The stream was 3.5 minutes long and consisted of 5,000 frames. Its annotation format was different as my one [13], however, four points on the two eyes, 2-2 on the two eyebrows, two on the nose, four on the mouth, and three on the face contour were consistent with the format I used.

2.2. Test Environment and Metrics

During testing the three approaches, I analyzed the accuracy of model fitting by calculating the *root-mean-square* (RMS) error – between the annotated feature points and the points of the fitted shape model. I separately measured the RMS error for the frontal and rotated faces. I was able to determine the rotation invariance of the methods with this test case. During the evaluation, I aggregated the results for the feature points according to the facial parts, such that for the left eye, right eye, left eyebrows, right eyebrows, nose, mouth, and I performed the comparisons based on these groups.

2.3. Conclusion

I did not experience significant differences between the three methods during testing them for frontal faces. It appeared clearly from the results that the error of eyebrows was almost twice as for the other feature points. However, I faced with worse results when I tested the approaches for the rotated faces. The AAM was failed during the tests: it produced nearly 2.5x worse results than in the frontal case. By contrast, the other two

methods only had 1.5x degradation. I could perform more experiment with AAM e.g. with more parameters or try another implementation, but due to the effectiveness of the other two implementations, I dismissed these ideas. Nearly 80% of test cases, the CLM achieved better results than ASM, but the difference was not significant. Thus, the runtime became one of the most significant factor as the basis of my choice.

I thoroughly tested the runtime on a video stream. The runtime of ASM fitting (30 ms) was half as the CLM (60 ms). I further optimized the runtime of fitting in practice with the following approach: instead of fitting the $\bar{s}$ average shape on the $(k + 1)$ element of the incoming image sequence, I decided to warp the shape from the k th iteration. In this way, the fitting converged faster. As for the accuracy of the fitting, the ASM error was almost always lower – about 6 pixels – than the CLM. The ASM was more accurate with 2.24 pixels than the CLM over the full length of the video stream. The difference was not significant, but during the tests, I found that ASM fitting was more stable than CLM. By this, I mean that the error function of fitting was smoother between the consecutive frames of the input image sequence. As a summary, I found that the accuracy and runtime of ASM could be appropriate to create more complex HCI applications in the future.

3. Thesis: Head Pose Estimation

During my PhD studies, I created a system based on my previous results that used image processing algorithms to determine the head pose on 2-D images. The task included the representation and estimation of the head pose; composition of the necessary test databases; and the development of a test environment for measuring results [P1, P2, P4, P6, P7, P8, P10].

3.1. Head Pose Estimation

Head pose is the combination of the spatial position and orientation of the head. The computer can have a feedback about the focus of human's attention in the knowledge of head pose. I determined the head pose by defining a six degrees of freedom (6DoF) model in the 3-D Euclidean space. The representation consisted of a 3-D vector that described the spatial position of the head, which means where the head in the world coordinate system was located. Furthermore, it contained the orientation of the head, which means how the head was aligned in the world relative to a fixed coordinate system.

The method – I developed – determined the head pose based on 3-D points and their 2-D projections. The 2-D points were defined on the image plane of the camera coordinate system: these were the boundary points of the shape model – described in the previous chapter. The 3-D points were defined in a model coordinate system and belonged to a 3-D model that had a similar geometry as the 2-D shape model had. I defined the 3-D model points by the use of a stereo image database that contained depth maps of human faces [14]. Based on the 3-D and 2-D points, I defined head pose as a solution of a *perspective n-Point problem* (PnP problem). The solution of the PnP problem

was determined by POSIT [15]. POSIT approximated the perspective projection by scaled orthographic projection based on at least four non-coplanar points on the 3-D space and determined the spatial position and orientation of the corresponding model.

3.2. Coordinate Systems

On the one hand, I needed a *camera coordinate system* that looked towards the positive direction of the z axis; the image rows were parallel to the x axis and the columns to the y axis. I also defined a 3-D *model coordinate system* in which I expressed the 3-D model points. The model coordinate system had the same handedness as the camera coordinate system. This was recommended because the POSIT gave the model pose relative to the camera.

3.3. Test Databases and Environment

During my tests, I paid special attention to the rotation of the head around the x (*pitch*) and y (*yaw*) axis, as I used mostly these values in the higher levels components of my system. I tested my head pose estimation approach on two databases. Some of the tests were run on the Biwi database [16], which was recorded by a Kinect motion sensor. The database contained more than 15,000 measurements from 20 people, where the yaw covered the $\pm 75^\circ$ range and so did the pitch the $\pm 60^\circ$ range. The database was annotated automatically so there was no information about the accuracy of the annotation data.

I also tested the accuracy of my method on the GI4E database [17]. The database contained more than 36,000 measurements from 10 people, with the range $\pm 30^\circ$ yaw and

$\pm 30^\circ$ pitch. In this database, the authors provided much more metadata, e.g. the ground truth coordinates of the 2-D and 3-D shape models with 54 points and the real head pose. The latter was determined by a dedicated sensor. Based on the authors' measurements 0.83° roll, 0.86° yaw and 1.05° pitch accuracy could be considered for the annotation data.

3.4. Conclusion

The Biwi database covered a relative large range for pitch and yaw. I was mostly interested in the accuracy of the head pose estimation along the whole width of the intervals with the use of Biwi. For yaw, the angle range of $\pm 40^\circ$ was that where the absolute error of the orientation was below 10° , and the error function reached its minimum (below 5°) in the $\pm 12^\circ$ range. Approaching the extreme values, the error increased exponentially. Here ASM fitting could also play a major role. Remember that the ASM fitting had 1.5 times greater error for the faces that were rotated in $\pm 45^\circ$ than for the pure frontal faces.

For pitch, the angle range of $[-32^\circ, +40^\circ]$ was that where the absolute error of the orientation was below 10° and the error function reached its minimum (below 5°) in the range of $[-6^\circ, +16^\circ]$. The error function was not symmetrical to 0° , which meant that the algorithm worked with higher error for people facing upwards. As the camera was looking straight away towards the test subjects, I needed to perform further tests to reveal this fact. My tests showed that the error of the head movement around the horizontal axis correlated with the rotation around the vertical axis of the head: I measured the

highest errors in the bottom-left and the top-left directions. This could unfortunately arise from the nature of the database.

The GI4E database covered a small range of pitch and yaw, but I was able to perform a lot useful tests by the use of it. The results showed that the pitch was also more inaccurate than yaw, but the characteristics of the two functions were fit better to each other. For yaw, the angle range of $\pm 25^\circ$ was that where the absolute error of the orientation reached its minimum (below 5°). It was much better than what I measured for Biwi. For pitch, the angle range of $\pm 15^\circ$ was that where the absolute error of the orientation reached its minimum (below 5°).

The authors of the database published some valuable metadata e.g. the real 2-D and 3-D coordinates of a 54-point shape model. I used them to verify the error of my head pose estimation procedure. The absolute error – determined for the ground truth measurements – was uniform over the intervals. I measured 1.45° for yaw and 1.77° for pitch. I also tested the error of reprojection my own 3-D model onto the image plane – in function of the head pose published by the authors. The reprojection error was also uniform over the whole interval. I measured 2-6 pixels as reprojection errors. The accuracy of my method was as reliable as the solutions in the literature [18], and enabled me to use it successfully in my further researches.

4. Thesis: Head Movement Recognition

During my PhD studies, I created a system that was able to detect complex head movements. The system was based on my results achieved during face detection, facial feature points extraction and head pose estimation. The task involved the spatio-temporal representation and recognition of head movements; the composition of a head movement database; furthermore the development of a test environment for measuring and comparing the results [P1, P2, P5, P9].

Recognition of head movements implied the recognition of human activities (HAR) based on several previous observations and measurements. I also studied the features of head movements: the typical length of a movement was three to five seconds, but I did not want to define any limitations for them. I did not even expect the user to be aware of exactly how-, and how much time a movement should be performed. This means that two head movements were never be the same.

4.1. Representation of Head Movements

My primary goal was to recognize complex head movements in real-time. I considered certain features of the head pose for the representation and determined the trajectory of head movements from them. I also localized the head movements in time. I calculated the start-, and end points of the movements from the head pose as well. The representation proposed by myself included the spatial and temporal dimensions of head movements.

I defined the head movements as time series, which were constructed from several temporal observations of a particular attribute. Regarding the spatial representation, I considered the

change of head orientation over the consecutive frames of an input image sequence. I also applied a Kalman filter to the „raw” output of the head poses in order to filter out the wrong poses and local noise. I assigned an $s \in \mathcal{S}$ time series as a statistical sample to each head movements shown in the image sequence, which was consisted of some $a_\tau, a_{\tau+1}, \dots, a_{\tau+n}$ observations. The a_i observations were the elements of the $\mathcal{F}$ feature space. The $m \in \mathcal{M}$ head movements were defined as follows:

$$m = (l, a_\tau, a_{\tau+1}, \dots, a_{\tau+n}) = (l, s), \quad (4.1)$$

where $l \in L$ was the label of the activity performed (e.g. nodding or shaking). During the measurement of attributes, I took the basis vector of the 3-D head model coordinate system pointing to the z direction: $\underline{k} = (0,0,1)$. I rotated $\underline{k}$ according to the current head pose and denoted the resulted vector by $\underline{k}_i$. Then the a_i attributes were defined as the difference of direction vectors calculated for the current and the previous frames:

$$a_i = \underline{k}_i - \underline{k}_{i-1}. \quad (4.2)$$

Thus a_i expressed the change of head orientation between two consecutive frames. I also shifted the a_i measurements with a_τ , so the measurements became independent from the starting orientation of the head.

4.2. Process of Recognition

The time series of head movements may differ in time or velocity. Therefore, they can only be aligned non-linearly regarding the time dimension. I developed the alignment on a dynamic time warping (DTW) method [19]. DTW determines

the optimal alignment of two time series by transforming the time axis. Let the two time series be:

$$\begin{aligned} X &= (x_1, x_2, \dots, x_N) \\ Y &= (y_1, y_2, \dots, y_M) \end{aligned} \quad (4.3)$$

DTW tries to fit the x_i and y_j attributes to each other. A $C \in \mathbb{R}^{N \times M}$ cost matrix is constructed with the use of a distance function and then it determines the path that has the minimal cost in the matrix, which is the optimal fit between the two time series.

4.3. Head Movement Database

DTW determines the optimal alignment of two time series, but I wanted to detect more than two kind of head movements. I accomplished it by the definition of a DTW-based classifier. I created a 26-class classifier for the intensive analysis of my HAR system. Let us denote the classes with $M_i \subset \mathcal{M}$, where $\mathcal{M}$ is a set of head movements:

$$M = \{M_i \subset \mathcal{M} \}_{i \in [1:26]}. \quad (4.4)$$

The M_i classes were the trajectories of head-drawn letters of the English alphabet, so each M_i class contained head movements like, a head-drawn p or h letter. The classifier was based on the spatial-ordered nature of head movements (regarding the DTW distance) and a head movement database. The decisive point of DTW is to calculate the $C \in \mathbb{R}^{N \times M}$ cost matrix between two head movements, which is computationally expensive in practice. I attempted to simplify the process of recognition as much as possible. Along with certain principles, I

reduced the cardinality of each M_i class to one, and then I created a 1-nearest neighbor (1-NN) classifier.

I determined one head movement per classes: the one that represents all the elements of its class the best; mark this element with m_i^r . Therefore, I determined 26 pieces of m_i^r head movements. During the composition of the database, I collected 20 head movements per letter from 10 people, thus so I got 200 head movements for each letter. I used 100 head movements per class to determine the m_i^r head movement. The remaining 100-100 head movements were used for testing purposes only. I expected from m_i^r that it distinguishes the best the M_i from the other. Thus, I calculated the DTW distances between all possible element pairs within the M_i class and I selected m_i^r to that element which minimized the distance within its class

4.4. Conclusion

In the first step of testing, I examined the average distance (μ_i) and corrected sample standard deviation (σ_i) of each m_i^r element within the M_i classes. However, I got only a feedback for the spatial extent of elements within the classes, and not for the accuracy of recognition. I found that in the case of e and v the m_i^r elements were very close to the other points for example. Furthermore, I got bigger DTW distances for the m_i^r elements in case of t or q letters – the elements of these classes were located in a circle that had nearly three times bigger radius. In my further examinations, I found that the values μ_i and σ_i correlated with the average length of the time series in the M_i class. The high μ_i and σ_i values derived from the nature of DTW. That is, larger μ_i and σ_i values were expected for classes

that contain longer time series, but it was not even certain that the accuracy of the detection was worse for these classes.

My further tests were related to analysis of precision and recall. The average precision for the 26 classes was 0.81 and the average recall was 0.83, which was a guarantee for the low number of false positives and false negatives during recognition. However, I experienced more false positives in case of *a, k, l, q, t* and *y* classes, the precision for these classes was 0.56. Furthermore, more false negatives were appeared for the *a, d, e* and *h* classes, here the average recall was 0.63. I also determined the F_1 score for the classes. The following head movements had the lowest error based on F_1 score: *r, u, j, i, w, f, v, s, x* and *z*. I have to highlight that all of the 100 head movements were correctly recognized in case of *z*. Furthermore, the head movements with the lowest accuracy could be expected for the *a, q, l, t, d, h, n, k, m* and *y* classes. The cause of problem here was related to the similarities of certain movements to the other classes. For example, an *a* letter was drawn similarly as *d* and the same for the letters *m – n* or *l – e*. I also examined the possible correlation between the average length of head movements and their F_1 score. It can be stated that the measurements were not correlated: shorter samples could be detected with high accuracy (e.g. *i, v* or *z*), or longer samples with low accuracy (e.g. *q, h* and *m*) and vice versa. Summarizing everything, I found that the accuracy of my system was over 80% for all of the classes (both in terms of precision and recall), which could be considered as a good result.

Irodalomjegyzék

- [1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511-518, 2001.
- [2] L. Hunyadi, K. Bertók, I. Szekrényes, Á. Abuczki, G. Nagy, N. Nagy, P. Németi and A. Bódog, "The outlines of a theory and technology of human-computer interaction as represented in the model of the HuComTech project," *2nd International Conference on Cognitive Infocommunications*, pp. 1-5, 2011.
- [3] A. S. Georghiades, P. N. Belhumeur and D. J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643-660.
- [4] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 94-101, 2010.
- [5] Neowise Software, "NeoDownloader, Download and view pictures & videos from websites," 22 January 2014. [Online]. Available: <http://www.neodownloader.com/>. [Accessed 2015].
- [6] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," *International Conference on Image Processing*, pp. 900-903, 2002.
- [7] S. Liao, X. Zhu, Z. Lei, L. Zhang and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," *Proceedings of the 2007 International Conference on Advances in Biometrics*, pp. 828-837, 2007.

- [8] A. Torralba, K. P. Murphy and W. T. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 762-769, 2004.
- [9] T. F. Cootes, C. J. Taylor, D. H. Cooper and J. Graham, "Active shape models - Their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38-59, 1995.
- [10] T. F. Cootes, G. J. Edwards and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681-685, 2001.
- [11] D. Cristinacce and T. Cootes, "Automatic Feature Localisation with Constrained Local Models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054-3067, 2008.
- [12] R. Gross, I. Matthews, J. F. Cohn, T. Kanade and S. Baker, "Multi-PIE," *Proceedings of the Eighth IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.
- [13] K. Messer, J. Matas, J. Kittler, J. Luetttin and G. Maitre, "XM2VTSDB: The Extended M2VTS Database," *2nd International Conference on Audio and Video-based Biometric Person Authentication*, pp. 72-77, 1999.
- [14] S. Gupta, M. K. Markey and A. C. Bovik, "Anthropometric 3D Face Recognition," *International Journal of Computer Vision*, vol. 90, no. 3, pp. 331-349, 2010.
- [15] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision - Special issue: image understanding research at the University of Maryland*, vol. 15, no. 1-2, pp. 123-141, 1995.
- [16] G. Fanelli, M. Dantone, J. Gall, A. Fossati and L. Gool, "Random Forests for Real Time 3D Face Analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 437-458, 2013.

- [17] Grupoinvest GI4E, "GI4E Head Pose Database," Public University of Navarra, Campus Arrosadía, 31006, Pamplona Navarre (Spain), 2015.
- [18] E. Murphy-Chutorian and M. M. Trivedi, "Head Pose Estimation in Computer Vision: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607-626, 2009.
- [19] D. J. Bemdt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," AAAI-94 Workshop on Knowledge Discovery in Databases, 1994.

Publikációk listája

Könyvfejezet (külföldi)

- [P1] K. Bertók, A. Fazekas, "Gesture Recognition: Performance, Applications and Features," *Hauppauge, NY: Nova Science Publishers*, pp. 1-30, 2018, (megjelenés alatt).

Referált folyóiratcikk (külföldi)

- [P2] K. Bertók, A. Fazekas, "Recognizing Complex Head Movements," *Australian Journal of Intelligent Information Processing Systems*, vol. 14, no. 4, pp. 3-17, 2016.
- [P3] L. Sajó, K. Bertók, A. Fazekas, "Perceptual Color Representation of the Face: Extracting the Color of Skin, Hair and Eyes," *JPRR* vol. 6, no. 2, pp 166-174, 2011.

Konferenciakötetben megjelent cikk (külföldi)

- [P4] K. Bertók, A. Fazekas, "Face Recognition on Mobile Platforms," *7th IEEE International Conference on Cognitive Infocommunications*, pp. 37-43, 2016.
- [P5] K. Bertók, A. Fazekas, "Recognizing Human Activities Based on Head Movement Trajectories," *5th IEEE Conference on Cognitive Infocommunications*, pp. 273-278, 2014.
- [P6] K. Bertók, A. Fazekas, "Gesture Recognition - Control of a Computer with Natural Head Movements," *International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications*, pp. 527-530, 2012.
- [P7] L. Hunyadi, K. Bertók, T.E. Németh, I. Szekrényes, Á. Abuczki, G. Nagy, N. Nagy, P. Némethi, A. Bódog, "The outlines of a theory and technology of human-computer

interaction as represented in the model of the HuComTech project," *2nd Cognitive Infocommunications International Conference*, pp. 249-255, 2011.

Folyóiratcikk (hazai)

- [P8] K. Bertók, L. Sajó, A. Fazekas, "A robust head pose estimation method based on POSIT algorithm," *Argumentum* vol. 7, pp. 348-356, 2011.

Konferenciakötetben megjelent cikk (hazai)

- [P9] K. Bertók, A. Fazekas, "Fejmozgás alapú gesztusok felismerése," *Képfeldolgozók és Alakfelismerők országos 9. konferenciája*, pp. 736-746, 2013.
- [P10] K. Bertók, A. Fazekas, L. Sajó, "Az emberi fej térbeli helyzetének és a tekintet irányának meghatározása," *Képfeldolgozók és Alakfelismerők 8. konferenciája*, pp. 393-406, 2011.
- [P11] Fazekas, S. Szeghalmy, K. Bertók, L. Sajó, "Multi-modális ember-gép kapcsolatok," *Képfeldolgozók és Alakfelismerők 8. konferenciája*, pp. 387-392, 2011.

Egyéb

- [P12] T. Balogh, K. Bertók, "Közgazdász vagy informatikus? - A gazdaságinformatikus képzés hallgatói megítélése a Debreceni Egyetemen," *Informatika a felsőoktatásban 2011*, pp. 110-119, 2011.

Hitelesített publikációs lista



**DEBRECENI
EGYETEM**

**DEBRECENI EGYETEM
EGYETEMI ÉS NEMZETI KÖNYVTÁR**
H-4002 Debrecen, Egyetem tér 1, Pf.: 400
Tel.: 52/410-443, e-mail: publikaciok@lib.unideb.hu

Nyilvántartási szám: DEENK/58/2018.PL
Tárgy: PhD Publikációs Lista

Jelölt: Bertók Kornél
Neptun kód: G8PK4D
Doktori Iskola: Informatikai Tudományok Doktori Iskola
MTMT azonosító: 10047414

A PhD értekezés alapjául szolgáló közlemények

Idegen nyelvű, külföldi könyvrészletek (5)

1. **Bertók, K.**, Fazekas, A.: Facial gesture recognition and its applications.
In: Gesture recognition : performance, applications and features. Eds.: Gilberto Coleman, Roosevelt Ward, Nova Science Publishers, New York, 1-30, 2018. ISBN: 9781536134919
2. **Bertók, K.**, Fazekas, A.: Face recognition on mobile platforms.
In: 7th IEEE International Conference on Cognitive Infocommunications: CogInfoCom 2016 : Proceedings. Szerk.: Péter Baranyi, IEEE Computer Society, Washington, 37-43, 2016.
3. **Bertók, K.**, Fazekas, A.: Recognizing human activities based on head movement trajectories.
In: 5th IEEE International Conference on Cognitive Infocommunications. Ed.: Péter Baranyi, IEEE, [Danvers], 273-278, 2014. ISBN: 9781479972791
4. **Bertók, K.**, Fazekas, A.: Gesture recognition - control of a computer with natural head movements.
In: GRAPP & IVAPP 2012: Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications. Eds.: Paul Richard, Martin Kraus, Robert S. Laramée, José Braz, SciTePress, Rome, 527-530, 2012. ISBN: 9789898565020
5. Hunyadi, L., **Bertók, K.**, Németh, T. E., Szekrényes, I., Abuczki, Á., Nagy, G., Nagy, N., Németh, P., Bódog, A.: The outlines of a theory and technology of human-computer interaction as represented in the model of the HuComTech project.
In: Proceedings of 2nd Cognitive Infocommunications International Conference. Ed.: by Baranyi Péter, IEEE Computer Society, New York, 249-255, 2011. ISBN: 9781457718069

Idegen nyelvű tudományos közlemények hazai folyóiratban (1)

6. **Bertók, K.**, Sajó, L., Fazekas, A.: A robust head pose estimation method based on POSIT algorithm.
Argumentum. 7, 348-356, 2011. ISSN: 1787-3606.





Idégen nyelvű tudományos közlemények külföldi folyóiratban (1)

7. **Bertók, K.**, Fazekas, A.: Recognizing complex head movements.
Aust. J. Intell. Inform. Process. Syst. 14 (4), 3-17, 2016. ISSN: 1321-2133.

Magyar nyelvű konferencia közlemények (2)

8. **Bertók, K.**, Fazekas, A.: Fejmozgás alapú gesztusok felismerése.
In: Képfeldolgozók és Alakfelismerők országos 9. konferenciája : KÉPAF Konferencia
kiadvány. Szerk.: Czúni László, NJSZT-KÉPAF, Bakonybél, 736-746, 2013.
9. **Bertók, K.**, Fazekas, A., Sajó, L.: Az emberi fej térbeli helyzetének és a tekintet irányának meghatározása.
In: Képfeldolgozók és Alakfelismerők VIII. konferenciája. Szerk.: Kató Zoltán, Palágyi Kálmán, NJSZT-KÉPAF, Szeged, 393-406, 2011.

További közlemények

Idégen nyelvű tudományos közlemények külföldi folyóiratban (1)

10. Sajó, L., **Bertók, K.**, Fazekas, A.: Perceptual color representation of the face: Extracting the color of skin, hair and eyes.
J. Pattern Recognit. Res. 6 (2), 166-174, 2011. ISSN: 1558-884X.
DOI: <http://dx.doi.org/10.13176/11.259>

Magyar nyelvű konferencia közlemények (2)

11. Balogh, T. L., **Bertók, K.**: Közgazdász vagy informatikus?: a gazdaságinformatikus képzés hallgatói megítélése a Debreceni Egyetemen.
In: Informatika a felsőoktatásban 2011 konferencia [elektronikus dokumentum] : konferencia kiadvány. Szerk.: Cser László, Herdon Miklós, Debreceni Egyetem Informatikai Kar, Debrecen, 110-118, 2011.
12. Fazekas, A., Szeghalmi, S., **Bertók, K.**, Sajó, L.: Multi-modális ember-gép kapcsolatok.
In: Képfeldolgozók és Alakfelismerők VIII. konferenciája. Szerk.: Kató Zoltán, Palágyi Kálmán, NJSZT-KÉPAF, Szeged, 387-392, 2011.

A DEENK a Jelölt által az IDEa Tudóstérbe feltöltött adatok bibliográfiai és tudománymetriai ellenőrzését a tudományos adatbázisok és a Journal Citation Reports Impact Factor lista alapján elvégezte.

Debrecen, 2018.03.07.





Registry number:

DEENK/58/2018.PL

Subject:

PhD Publikációs Lista

Candidate: Kornél Bertók

Neptun ID: G8PK4D

Doctoral School: Doctoral School of Informatics

MTMT ID: 10047414

List of publications related to the dissertation

Foreign language international book chapters (5)

1. **Bertók, K.**, Fazekas, A.: Facial gesture recognition and its applications.
In: Gesture recognition : performance, applications and features. Eds.: Gilberto Coleman, Roosevelt Ward, Nova Science Publishers, New York, 1-30, 2018. ISBN: 9781536134919
2. **Bertók, K.**, Fazekas, A.: Face recognition on mobile platforms.
In: 7th IEEE International Conference on Cognitive Infocommunications: CogInfoCom 2016 : Proceedings. Szerk.: Péter Baranyi, IEEE Computer Society, Washington, 37-43, 2016.
3. **Bertók, K.**, Fazekas, A.: Recognizing human activities based on head movement trajectories.
In: 5th IEEE International Conference on Cognitive Infocommunications. Ed.: Péter Baranyi, IEEE, [Danvers], 273-278, 2014. ISBN: 9781479972791
4. **Bertók, K.**, Fazekas, A.: Gesture recognition - control of a computer with natural head movements.
In: GRAPP & IVAPP 2012: Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications. Eds.: Paul Richard, Martin Kraus, Robert S. Laramée, José Braz, SciTePress, Rome, 527-530, 2012. ISBN: 9789898565020
5. Hunyadi, L., **Bertók, K.**, Németh, T. E., Szekrényes, I., Abuczki, Á., Nagy, G., Nagy, N., Németi, P., Bódog, A.: The outlines of a theory and technology of human-computer interaction as represented in the model of the HuComTech project.
In: Proceedings of 2nd Cognitive Infocommunications International Conference. Ed.: by Baranyi Péter, IEEE Computer Society, New York, 249-255, 2011. ISBN: 9781457718069

Foreign language scientific articles in Hungarian journals (1)

6. **Bertók, K.**, Sajó, L., Fazekas, A.: A robust head pose estimation method based on POSIT algorithm.
Argumentum. 7, 348-356, 2011. ISSN: 1787-3606.

Foreign language scientific articles in international journals (1)





7. **Bertók, K.**, Fazekas, A.: Recognizing complex head movements.
Aust. J. Intell. Inform. Process. Syst. 14 (4), 3-17, 2016. ISSN: 1321-2133.

Hungarian conference proceedings (2)

8. **Bertók, K.**, Fazekas, A.: Fejmozgás alapú gesztusok felismerése.
In: Képfeldolgozók és Alakfelismerők országos 9. konferenciája : KÉPAF Konferencia
kiadvány. Szerk.: Czúni László, NJSZT-KÉPAF, Bakonybél, 736-746, 2013.
9. **Bertók, K.**, Fazekas, A., Sajó, L.: Az emberi fej térbeli helyzetének és a tekintet irányának
meghatározása.
In: Képfeldolgozók és Alakfelismerők VIII. konferenciája. Szerk.: Kató Zoltán, Palágyi
Kálmán, NJSZT-KÉPAF, Szeged, 393-406, 2011.

List of other publications

Foreign language scientific articles in international journals (1)

10. Sajó, L., **Bertók, K.**, Fazekas, A.: Perceptual color representation of the face: Extracting the color
of skin, hair and eyes.
J. Pattern Recognit. Res. 6 (2), 166-174, 2011. ISSN: 1558-884X.
DOI: <http://dx.doi.org/10.13176/11.259>

Hungarian conference proceedings (2)

11. Balogh, T. L., **Bertók, K.**: Közgazdász vagy informatikus?: a gazdaságinformatikus képzés
hallgatói megítélése a Debreceni Egyetemen.
In: Informatika a felsőoktatásban 2011 konferencia [elektronikus dokumentum] : konferencia
kiadvány. Szerk.: Cser László, Herdon Miklós, Debreceni Egyetem Informatikai Kar,
Debrecen, 110-118, 2011.
12. Fazekas, A., Szeghalmy, S., **Bertók, K.**, Sajó, L.: Multi-modális ember-gép kapcsolatok.
In: Képfeldolgozók és Alakfelismerők VIII. konferenciája. Szerk.: Kató Zoltán, Palágyi
Kálmán, NJSZT-KÉPAF, Szeged, 387-392, 2011.



The Candidate's publication data submitted to the iDEa Tudóstér have been validated by DEENK on
the basis of Web of Science, Scopus and Journal Citation Report (Impact Factor) databases.

07 March, 2018