Fejmozgás alapú gesztusok felismerése

Bertók Kornél, Fazekas Attila

Debreceni Egyetem
Informatikai Kar
Debreceni Képfeldolgozó Csoport
H-4010 Debrecen, Pf.:12.
bertok.kornel@inf.unideb.hu, attila.fazekas@inf.unideb.hu

Absztrakt. Jelen cikk témája egy fejmozgás alapú gesztusfelismerő rendszer, mely segítségével lehetőségünk nyílik mozdulatsorok valósidejű felismerésére, megértésére, azok rögzítésére és később adatbányászati eszközökkel történő elemzésére. Mindezek mellett a rendszernek részét képezi egy fejmozgás alapú gesztusokat tartalmazó adatbázis is, melyet az egyes gesztusok felismerése során – bizonyos feltételek mentén – online bővítünk, így lehetőségünk nyílik a felismerés futási idejű javítására.
Jelen tanulmány egy módszert definiál a kamera képeken, mint képszekvenciákon megjelenő fejmozgás alapú gesztusok reprezentációjára, tér-, és időbeli behatárolására, gesztus-osztályok kialakítására, továbbá bevezet egy metrikát az osztályokkal történő összehasonlítására. Meg kell jegyeznünk azt is, hogy az elkészült rendszer jól illeszkedhet egy multimodális ember-gép kommunikációt leíró modellbe is, mert használata új fejezetet nyit a metakommunikációhoz tartozó csatornák vizsgálatában.

Kulcsszavak: fejmozgás, gesztus felismerés, gesztus reprezentáció, gesztus adatbázis.

1 Bevezetés

Az ember-számítógép interakció kutatási feladatai közé tartozik, hogy olyan új, esetlegesen alternatív kommunikációs eszközöket és módszereket fejlesszen, amelyek segítik az ember-gép kapcsolatot az ember számára minél természetesebbé, magától értetődővé tenni. A különböző eszközök és programok vezérlésére sokféle megoldás létezik. Csakhogy az eszközök és programok számának növekedésével a különböző vezérlő megoldások száma is növekszik. Tehát mindenképpen szükséges egy természetesebb, eszköz-független módot találni az irányítására. A kommunikáció egyszerűsítésével kapcsolatos ötleteket célszerű a mindennapi életünkben keresni.

A szóbeliség (verbális jelek halmaza) az emberi kommunikáció legtipikusabb módja, jelentős információhordozó. Ugyanakkor gyakran lehet félreértések forrása is, mivel azzal a feltételezéssel élünk, hogy egy-egy szó azonos jelentéssel bír mindenki számára. Pedig azt, hogy egy-egy szónak az adott pillanatban milyen jelentést tulajdonítunk, aktuális szükségleteink is jelentős mértékben befolyásolják. Ezért az egyes kommunikációs szituációkat kontrollálni kell.

A verbális jelek mellett a szóbeli információk kiegészítésére, ellenőrzésére vagy éppen hangsúlyozására a nem szóbeli, úgynevezett non-verbális jelrendszert alkalmazzuk. A non-verbális jelek tipikus megnyilvánulásai a mimika, a tekintet – szemkontaktus – szemmozgás, az úgynevezett vokális jelek, mint hangnem, hanghordozás, hangerő, hangszín; a gesztusok, a testtartás és a távolságtartás-térközszabályozás.

Jelen tanulmány a gesztusok, mint non-verbális jelek felismerésére korlátozódik. Gesztusok alatt értjük a fej-, a kéz-, és a karok mozgását. A fejmozgások gyakoribb jelentései: az igenlés, a tagadás, a helytelenítés, a megszégyenülés, elszomorodás. A kéz-, és karmozgások jelentése: a hívás, elutasítás, tiltakozás, kérés, könyörgés, fenyegetés, köszöntés. A gesztusokat a partner beszédének szabályozására (magyarázás, gyorsítás-lassítás stb.) is használjuk. E mozgásoknak jelentésük van, egy részük tudatos, másik felük öntudatlan.

Jelen cikk témája egy gesztusfelismerő rendszer ismertetése, mely segítségével felismerhetőek és megérthetőek a tudatos fejmozgások. Továbbá egy eszközrendszert biztosít a gesztusok rögzítésére és később adatbányászati eszközökkel történő elemzésére, valamint a már rögzített mozdulatsorok segítségével a felismerés futási idejű javítására.

1.1 Irodalmi áttekintés

A meglévő gesztusfelismerő rendszerekkel kapcsolatban egy áttekintő összefoglalót ismertet a [1] tanulmány. Ebben az alfejezetben csak néhány olyan munkát foglalunk össze, melyeket az előző összefoglalón kívül alaposabban tanulmányoztunk.

A fejmozgás alapú – vagy általánosságban csak a mozgás alapú – gesztusfelismerő eljárások két csoportba oszthatóak: modell alapú és mintaillesztéses módszerekre. A modell alapú eljárások csoportjába tartoznak a különböző rejtett Markov modellek (továbbiakban: HMM) és azok variánsai. Példának okán, Marcel és munkatársai [2] egy input-output HMM-et készítettek az úgynevezett várható értéket maximalizáló algoritmus (EM algoritmus) segítségével. A kapott modellt később arra alkalmazták, hogy a kézfej körvonalából ismerjenek fel gesztusokat. A szakirodalomban megtalálható a hagyományos HMM-ek néhány javított változata is, ezek között akadnak olyanok, melyek működését szemantikus hálók használatával tökéletesítették [3], sok hivatkozást találunk a nem-paraméteres HMM-ek [4] használatára, illetve a feltételes valószínűségi mezők (Hidden Conditional Random Field) [5] alkalmazására is találunk példát. Ezen variánsok bizonyos esetekben egyrészt csökkentik a tanítás költségét, másrészt növelik az osztályozás pontosságát.

Népszerű modell alapú eljárások a véges állapotú gépek [6], valamint a dinamikus Bayes hálók [7] is. Ezek az eljárások feltételezik, hogy a fej mozgásának trajektóriája – vagyis az artikuláció – ismert. Ezekkel az eljárásokkal bíztató eredményeket lehet elérni, de meg kell jegyezni, hogy a robusztusságuk nagyban függ az arc detektálásának és a mozgás követésének sikerétől. Az sem elhanyagolható tény, hogy használatukat megelőzően sok adatra és számításigényes eljárások alkalmazására van szükség.

Ezzel szemben a mintaillesztéses módszerekkel elkerülhető a modell alapú eljárásokban rejlő nehézségek egy része. Ez az egyes gesztusok invariáns

reprezentálásával és azok közvetlen egymáshoz illesztésével érhető el. A meglévő módszerekben leggyakrabban tértől és időtől függő leírókat használnak [8,9,10,11]. A mintaillesztéses módszerek referencia eljárását Laptov és munkatársai [12] alkották meg. Az általuk megalkotott módszer irányított gradiensek (HOG) és az optikai áramlás hisztogramjai (HOF) – mint leírók – alkalmazásán alapul.

Mindezeken kívül egyéb módszerek is léteznek a szakirodalomban, melyek a mozgás pályájának; tér-, és időbeli gradienseknek; illetve az optikai áramláshoz tartozó globális hisztogramok [10] felhasználásán alapulnak. Ezeknek az eljárások az a legnagyobb hátránya, hogy a futás során közvetlenül illesztik az egyes gesztusokat egy már meglévő adatbázisra, mely rontja az eljárások skálázhatóságát.

2 Fejmozgás reprezentálása

A legfontosabb kérdés, amit a munkálatok elkezdése előtt meg kell válaszolni, hogy milyen jellegű mozgásokat szeretnénk felismerni? A felismerni kívánt mozdulatsorozatok várhatóan 3-5 másodperc hosszúak lesznek – elegendő elképzelni egy fejrázás mozdulatsorát. Ugyanakkor az emberek két mozdulatsort, ha nincsenek tudatában annak, hogy hogyan végzik, igen kis valószínűséggel fogják ugyan abban az ütemben elvégezni, tehát a probléma egy nemlineáris illesztést igényel. Lévén itt egy valósidejű felismerést szeretnénk megvalósítani, ezért már egy 10 másodperces intervallumon kívül nincs is értelme mintákat keresni, továbbá a gesztusok viszonylag rövid időtartama miatt már kisszámú adat alapján is jól meg kell tudni különböztetni a gesztusokat.

Ebben a fejezetben részletesen ismertetjük a gesztusfelismerő rendszerünket. Megadunk egy új és hatékony vizuális reprezentációt a fejmozgást leíró jellemzők kinyeréséhez, amely elengedhetetlen a felismerő rendszer nagyméretű gesztus-adatbázison történő használatához. Az általunk bevezetett reprezentáció lényegében a mozgás menetét ábrázoló képsorozat tagjain alapul.

A sorozat minden tagján egy egyszerű FAST (Features from Accelerated Segment Test) detektorral meghatározzuk azokat a régiókat, melyek a legmeghatározóbbak a mozgás szempontjából. Következő lépésben a képsorozat minden szomszédos tagjára kiszámoljuk azokat az optikai áramlás vektorokat, melyek a FAST által meghatározott régiókhoz tartoznak. A régiók mozgása alapján meghatározzuk a fejmozgás globális mozgásvektorát, így lényegében egy irányvektort kapunk a gesztus sorozat minden szomszédos tagjára. Következő lépésben szegmentáljuk a képsorozatot, vagyis kijelöljük azokat a tagokat, melyek a gesztust határolják. Végezetül a gesztushoz tartozó irányvektorok sorozatát a dinamikus idővetemítés segítségével egy előre definiált gesztus-adatbázis elemeihez hasonlítjuk.

2.1 Mozgás ábrázolása

A mozgás megjelenítésére az úgynevezett MHI (Motion History Image) reprezentációt választottuk, mely egy képszekvencia mozgó objektumainak változásait írja le [13]. Az MHI reprezentáció egy időalapú sablonozó eljárás eredménye, mely ugyan egyszerű, de robusztus a mozgó objektumokra nézve. A

metódus leírásához legyen adva egy $\mathrm{I}(\mathrm{x}, \mathrm{y}, \mathrm{t})$ képszekvencia és egy $\mathrm{D}(\mathrm{x}, \mathrm{y}, \mathrm{t})$ bináris maszk, mely a képszekvencia azon régióit jelöli ki, ahol mozgás történt a t-edik időpillanatban. A MHI reprezentáció egy az időtől is függő sablonként értelmezhető, ahol minden egyes pixel értéke a mozgás egy függvényeként értelmezhető. Az eljárás az alábbi képlettel számolható:

$$\mathrm{MHI}_{\tau}(\mathrm{x}, \mathrm{y}, \mathrm{t}) = \begin{cases} \tau, \text{ ha } \mathrm{D}(\mathrm{x}, \mathrm{y}, \mathrm{t}) = 1, \\ \mathrm{Max}(0, \mathrm{MHI}_{\tau}(\mathrm{x}, \mathrm{y}, \mathrm{t} - 1) - 1), \text{ különben,} \end{cases} \tag{1}$$

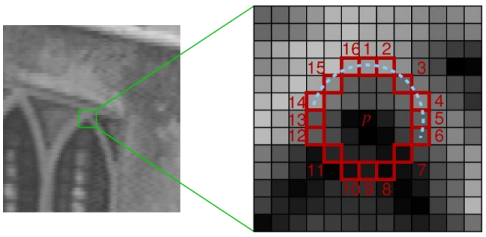
ahol τ a képszekvencia aktuális időbélyege, mely egyfajta korlátként működik, ugyanis általa nem jelennek meg az MHI reprezentációban azok a mozgások, melyek τ-nál régebben történtek. Vagyis az MHI-val jelölt kép összes olyan pixele, ahol mozgás volt τ értéket fog felvenni, még azok a részek ahol nem volt mozgás, fokozatosan elhalványulnak, majd törlődnek. Az eljárás grafikus reprezentációját az 1. ábra mutatja. Az MHI eljárást a mozgás szegmentálásában is felhasználtuk.

A gesztus a képszekvencia részsorozataként tekinthető, vagy máshogy fogalmazva a képszekvencia szegmensei alatt jelenik meg. A szegmensre teljesülnie kell a következő feltételnek: olyan nyitó és záró tagok határolják, hogy a hozzájuk számított MHI képek átlagintenzitás értéke alacsonyabb, mint egy előre definiált küszöbérték. Vagyis olyan képkockák határolják a szegmenst, ahol a mozgás intenzitása alacsony volt.

1. ábra. A bal oldali ábrán egy mozdulatsorhoz tartozó D(x, y, t) maszk látható, a jobb oldalin pedig a MHI reprezentáció.

2.2 A mozgást meghatározó régiók

Következő lépésben megkeressük azokat a régiókat az MHI reprezentáción, melyek meghatározzák a fejmozgást. Erre a FAST (Features from Accelerated Segment Test) algoritmust [14] használtuk. Ez lényegében egy egyszerű sarokdetektor, melynek hatékonysága az alacsony számításigényében rejlik. Működése során veszi a kép egyes pixelét, melyeknek egy adott sugarú környezetében vizsgálja a többi pixel értékét (lásd 2. ábra). Ha a környezetben szerepelő intenzitás értékek jelentősen nagyobbak, vagy kisebbek, mint a középpont, akkor azt sarokként osztályozza. Általában sarkok egy halmazát találja meg, ezért szokás mérni valahogy a sarkok erősségét is.



2. ábra. A FAST detektor által vizsgált tartomány egy potenciális sarokpont esetén. A detektor egy vizsgált pont körüli kör mentén vizsgálódik. Ha ebből valahány eltér a pixelnél legalább egy küszöbbel magasabb értékkel, akkor az adott középpont egy jellemző pont. Ebben az esetben a kör sugara 3 és 9 darab pixel tér el a küszöbtől.

2.3 Optikai áramlás

A következő lépésben a FAST által visszaadott jellemzőpontokra (lásd 3. ábra) számítjuk ki az optikai áramláshoz tartozó vektorokat az aktuális és a megelőző képkockára. Az optikai áramlás (Optical Flow) meghatározása lényegében nem más, mint több képen azonos képrészletek megfeleltetése. Az eredmény egy vektormező, amely az elmozdulásokat, vagyis a sebességvektorokat tartalmazza. Optikai folyamon tehát azt értjük, ahogy a képintenzitások mozgása megjelenik egymás utáni képeken.

3. ábra. A baloldali ábrán az MHI képről kinyert FAST jellemzőpontok láthatók. A jobboldalin ábrán, ugyanezen pontok az arc régióra szűkítve – szegmensek képpárjain ezekre számítjuk az optikai áramláshoz tartozó vektorokat.

Az optikai folyam algoritmusok az összetartozó képpontok megtalálásához feltételezik, hogy ezek intenzitása közel megegyezik. Szinte az összes módszer alapját ez a feltételezés adja, amit optikai folyam korlátozásként ismerünk. Jelölje $I(x, y, t)$ egy adott t pillanatban a képintenzitást, amely egy időben változó képsorozatból származik. A továbbiakhoz a következő feltételezéssel élünk: a mozgó vagy álló objektumok pontjainak intenzitása (lényegében) nem változik az idő múlásával. Legyen néhány objektum a képen, ami dt idő alatt (a gyakorlatban egymás utáni képvétel alatt) elmozdul (dx, dy) távolságra. Az $I(x, y, t)$ intenzitásértékek Taylor-sorba fejtésével és az előbb feltételezett állítások felhasználásával kapjuk:

$$-\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt}. \qquad (2)$$

Ezt a kifejezést rendszerint az optikai folyam feltételi egyenletének (vagy csak optikai folyamkorlátozásnak) nevezik, ahol $u = dx/dt$ és $v = dy/dt$ az optikai folyammező x és y koordináta irányú összetevői. Az egyenlet két ismeretlent (u, v) tartalmaz. A megoldásra, a Lucas–Kanade módszert [15] választottuk, mely úgy tekint egy pont sebességére, hogy az csak a pont helyi környezetétől függ. A szakirodalomban gyakran használják ennek a módszernek a piramisos változatát is.

2.4 Fejmozgás iránya

A gesztusok felismeréséhez le kell tudni írni azokat az optikai folyam vektorok függvényében. Ismeretes, hogy annyi optikai folyam vektor fog keletkezni a képszekvencia két szomszédos tagja között, amennyi sarokponttal a FAST eljárás tért vissza. A gesztusok könnyebb definiálása és felismerése érdekében definiálunk egy átlagvektort, az N darab optikai folyam vektor számtani közepeként. Majd ennek meghatározzuk az y tengely pozitív oldalával bezárt szögét az alábbi egyenlet segítségével:

$$\alpha = \mathrm{atan2}\left(x_{\text{kezdő}} - x_{\text{vég}}, y_{\text{vég}} - y_{\text{kezdő}}\right), \tag{3}$$

ahol $\mathrm{atan2}(x, y)$ függvény az x és y által meghatározott érték arkusz tangensét adja vissza radiánban. Ez hasonló az $y(x)$ arkusz tangenséhez, attól eltekintve, hogy a paraméterek előjele meghatározza, hogy az eredmény melyik környegyedbe esik. Az egységesség miatt gondoskodunk arról, hogy a koordinátarendszerünk jobbsodrású legyen (lásd (4)-es egyenlet).

$$\alpha = (2\pi - \alpha) \bmod 2\pi \tag{4}$$

4. ábra. Adott szegmens egy képpárja közötti fejmozdulás értéke szögben.

Így a vizsgált szegmens minden egyes képpárjára kapunk egy $\alpha \in [0, 2\pi]$ szöget, amelyet szögosztályok valamelyikébe soroljuk be az alábbi formulában definiált $f(\alpha)$ függvény segítségével:

$$f(\alpha) = (k+1) \cdot \frac{2\pi}{16} \;\Big|\; \alpha \in \left[k \cdot \frac{2\pi}{16}, (k+1) \cdot \frac{2\pi}{16}\right], \text{ahol } k = 0, \ldots, 15. \tag{5}$$

Az (5)-ös formula segítségével tulajdonképpen annyit csinálunk, hogy minden egyes α szöget a teljes szög azon tizenhatodába soroljuk be, amelyikbe esik. Így a képszekvencia k. szegmense alatt megjelenő gesztus leírható az $\{f^k(\alpha_0), f^k(\alpha_1), \ldots, f^k(\alpha_n)\}$ sorozattal, ahol α_i az i. képpárhoz számított átlagvektornak az y tengely pozitív oldalával bezárt szöge. A modul kimenetét a 4. ábra mutatja.

3 Gesztusfelismerés

Ebben a fejezetben egy eljárást ismertetünk az előző fejezetben definiált $\{f(\alpha_i)\}_{i=0}^{n}$ szögsorozatok egymáshoz történő illesztésére. Az illesztéshez szükség lesz egy előre definiált gesztus-adatbázisra. Az adatbázisban elempárokat tárolunk, melyekben benne van a gesztus neve és ahhoz egy előzetesen meghatározott szögsorozat. Erre látható egy példa a (6)-os formulában, melyben idézőjelek között szerepel az adott gesztus neve és mellette szögletes zárójelek között soroljuk fel a gesztus egyik szögsorozatát (az egyszerűség kedvéért fokokban):

$$\{\text{"fejrázás"}; [90°, 90°, 90°, 90°, 270°, 270°, 270°, 225°, 135°]\}. \quad (6)$$

A gesztusfelismerés során a képszekvencia minden egyes szegmensére meghatározzuk a hozzá tartozó szögsorozatot. Ezt követően a kapott szögsorozatot az úgynevezett dinamikus idővetemítés (DTW) [16] segítségével illesztjük az előre definiált gesztus-adatbázis elemeihez. Az adatbázisban egy gesztushoz több szögsorozat is létezhet, így az adatbázison belül a gesztusok csoportokat alkotnak. Az aktuális szögsorozatnak képezzük a gesztus-adatbázis összes elemével a DTW távolságát, majd ahhoz a gesztus-csoporthoz soroljuk be, melyhez a DTW átlagosan a legkisebb távolságot adta.

3.1 Dinamikus idővetemítés

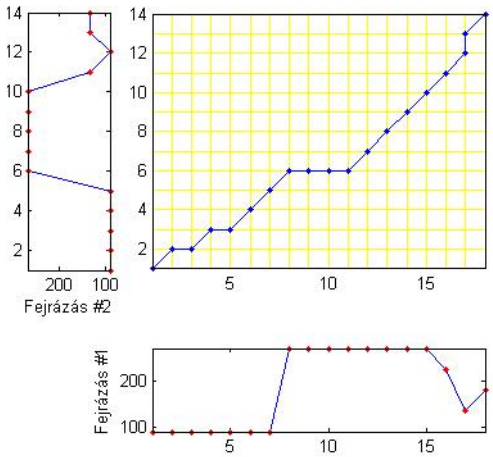
Az eljárás során tehát a felismerendő szögsorozatot az összes gesztus-csoport összes referenciamintájával össze kell hasonlítani, és az átlagosan legkisebb távolságú csoport neve lesz a felismerés eredménye. A DTW eljárás ismertetéséhez két szögsorozatnak, mint vektornak az egymáshoz való illesztésére redukáljuk a feladatot.

A DTW feladata, hogy azonos időtengelyre vetítsen egy aktuálisan detektált-, és egy tárolt fejmozgást, azaz hogy egy szögsorozatot összevethessünk a tárolt referenciákkal (a gesztus-csoportok elemeivel). Az összehasonlításhoz definiálni kell egy távolságot. A DTW algoritmusa lényegében egy N dimenziós vektort illeszt egy M dimenziós felismerendő vektorhoz. Az illesztés során a $(0, 0)$ kezdőpontból a (N, M) végpontba kell eljutni. Közben az útvonalkereső algoritmus lépésenként haladva a mintákat egymással összehasonlítja, és a távolság minimalizálására törekszik.

A két vektor távolságát többféleképpen számíthatjuk ki, tapasztalataink azonban azt mutatták, hogy a leggyakrabban használt módszerek közül az euklideszi távolság biztosítja a leghatékonyabb összehasonlítást, ezért a programunk is ezzel a távolsággal dolgozik. Az 5. ábra $(0, 0)$ pontjából induló, $(18, 14)$ pontjában végződő szakasza – vagyis a téglalap átlója – jelenti azt az utat, amely mentén haladva egyenletesen nyújtjuk, illetve zsugorítjuk a bemenő vektort az összehasonlításhoz. Ez a lineáris idővetemítés.

A vetemítő-görbe (az ábrán kék törött vonallal jelezve) tulajdonságai közé tartozik, hogy mindig monoton növekvő, lokális korlátok jellemzik és hogy lokális optimumokon keresztül elért teljes optimum. A vetemítés útvonala tehát nem lehet tetszés szerinti. Nem haladhat visszafelé. Ezen kívül az előre haladást is

sokféleképpen korlátozhatjuk, attól függően, hogy mekkora ingadozást engedünk meg az illesztés vonalán.



5. ábra. Egy futás alatti fejrázás gesztus illesztése az adatbázis egy fejrázás csoportjába tartozó elmére. A lineáris illesztést a koordinátarendszer $(0, 0)$ pontjából induló és $(18, 14)$ pontjában végződő szakasz jelentené. Az optimális nem lineáris illesztést a kék törött vonal jelzi. A DTW által adott távolság a két sorozatra: 5,6.

3.2 Gesztus adatbázis bővítése

Az előző fejezetben már tárgyaltuk, hogy a képszekvencia egyes szegmenseihez számított szögsorozatokat egy előre definiált adatbázis elemeihez illesztjük. Az illesztés jósága függ az adatbázisban tárolt rekordok darabszámától, illetve azok eloszlásától. kézenfekvőnek tűnt a rendszer működését úgy beállítani, hogy ha futás során a k. szegmenst *fejrázásnak* osztályoztuk, akkor új elemként felvesszük az adatbázis *fejrázás* osztályába a k. szegmenshez tartozó szögsorozatot. Ezzel a módszerrel a következő szegmens osztályozása során már egy bővebb információ halmaz áll a rendelkezésünkre az adott gesztus megjelenési formájáról.

4 Kísérletek és eredmények

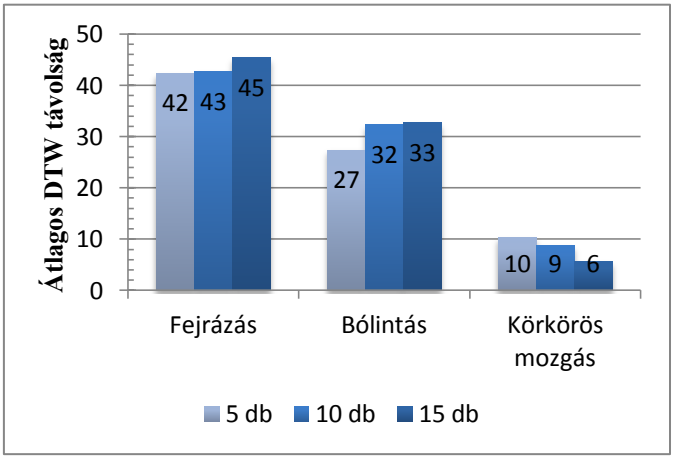
A tesztelés célja lényegében az volt, hogy megtaláljuk azokat a paramétereket, melyekkel a DTW algoritmus, a gesztus-adatbázis alapján kellő pontossággal képes diszkriminálni a különböző gesztusokat. Tehát tesztelés mellett lehetőségünk nyílik az optimális rendszer paraméterek beállítására is. Az alábbiakban felsoroljuk a fontosabb rendszerparamétereket:

1. Minimális és maximális gesztus hossz,
2. Maximális hiba,
3. Gesztus-adatbázis felépítése.

A minimális gesztushossz meghatározása azért fontos, mert a DTW könnyen ráillesztené a túl rövid gesztusokat a már egy kicsit is hasonló adatsorokra. Ez persze a maximális meredekség kritériumát állítva korlátozható. Ezért a minimális gesztushossz kezdetben nagyobb, mint egy másodperc. A DTW algoritmus futási ideje nagyban függ a gesztusok számától és azok hosszától. Célszerű tehát egy maximális gesztushosszt is meghatározni. Ha egy-egy gesztus maximális hossza 3-4 másodperc, és a rendszer 30FPS-sel halad végig a képszekvencián, akkor szükség van egy olyan pufferre, melyben közel 150 darab szöget el lehet menteni. Azonban a mintavételezés sebességét lecsökkentjük a harmadára, akkor a főbb mozdulat-komponensek, de – az illesztést gyorsítandó – a gesztusok maximális hossza lecsökken 50-re.

A maximális DTW hibát (ami valójában a gesztusok távolsága az aktuális szögsorozattól) legegyszerűbben empirikus módon lehet meghatározni. Mivel a DTW algoritmussal átlaghibát számoltatunk, ezért ennek maximális értéke könnyen becsülhető pár megfigyelés alapján is. Elvégeztük a rendszer betanítását három gesztusra, majd teljesen véletlenszerű mozgás közben figyeltük az átlaghiba változását. A mérések azt mutatták, hogy 30-60 között változott a hiba. Amikor valamelyik gesztushoz hasonló mozdulatsort végeztünk, akkor 15 alá csökkent. Ezek alapján a maximális távolságot (hibát) 15-re állítottuk.

Másik alapvető fontosságú jellemzője a rendszernek az előre definiált gesztus-adatbázis. Egyelőre nem minden részletében tisztázott az, hogy a gesztus csoportok számosságát hogyan célszerű megválasztani. Azonban az alábbi grafikonból (lásd 6. ábra) jól látszik, hogy bizonyos határok között érdemes az adatbázist online bővíteni a felismerés során. Erre vonatkozóan elvégeztünk egy tesztet. Az egyszerűség kedvéért három különböző gesztus osztály számosságát vizsgáltuk: a fejrázását, a bólintását és a körkörös fejmozgását. A futás során körkörös fejmozgáshoz hasonló gesztusokat illesztettünk az előbbi három osztály elemeihez.



6. ábra. Az osztályonkénti átlagos DTW távolság 20 darab körkörös fejmozgásra. Látható, hogy javul a DTW szeparáló teljesítménye, ha bizonyos határok között növeljük az egyes osztályok számosságát.

A 6. ábra, 20 darab körkörös fejmozgásnak az előbbi három darab gesztus osztálytól vett átlagos DTW távolságát szemlélteti különböző méretű – 5, 10 és 15 gesztus osztályonként – gesztus-adatbázisok esetén. Az ábrán jól látszik, hogy minél nagyobb az adatbázis mérete, annál kisebb az illesztett gesztusnak az átlagos távolsága a saját osztályától. A másik két osztálynál is megfigyelhető egy ellentétes irányú tendencia, habár ez nem minden esetben szembetűnő.

Általánosságban elmondható, hogy egy bővebb gesztus-adatbázis jó hatással lehet az osztályozás pontosságának növelésére és, hogy a gesztus-adatbázis online bővítésével is jobban illeszkedhet a gesztus-adatbázis az aktuális felhasználó gesztikulálásához. Azonban a gesztus-adatbázis mérete valós idejű feldolgozás követelménye miatt nem nőhet tetszőlegesen nagyra. Még abban az esetben sem, ha a DTW által illesztett szögsorozatok hossza nem lehet nagyobb 50-nél.

A jelenlegi rendszerben nyolc különböző gesztust szerepeltetünk, a négy alapirányt (fel, le, jobbra, balra), a körkörös-, és cikkcakk alakú fejmozgást, valamint a fejrázást és a bólintást. Ezekhez az osztályokhoz előzetesen harminc darab gesztust definiáltunk – osztályonként – a gesztus-adatbázisban. A futási idejű felismerés, azaz az osztályok szeparálása közel 100%-os, az egyes gesztusok és gesztusosztályok között hasonló DTW távolságok adódnak, mint amit a 6. ábra is mutat.

4.1 Összefoglalás

A jelenlegi rendszer kétségkívül bebizonyította, hogy alkalmas valós idejű, fejmozgás alapú gesztusok felismerésére. Mindemellett pár nyitott kérdést is megfogalmazott. Ilyen például a gesztus-adatbázisbeli osztályok méretének ésszerű meghatározása. További érdekes feladat lehet egy olyan összetettebb metrika készítése, mely a futás során a mozgás szegmensekből kinyert gesztusokat nem csak azoknak a gesztus-osztályoktól vett távolsága alapján ismeri fel, hanem egyéb az adatbázisból kinyert információt is alkalmaz. Használható javítás lehet az is, ha valahogy csökkentjük a gesztus-adatbázisbeli elemekkel való illesztések számát, vagy ha ajánlást adunk arra vonatkozóan, hogy mely elemekhez célszerű illeszteni a vizsgált sorozatot.

Véleményünk szerint további érdekes törvényszerűségeket lehet felfedezni az emberi gesztikulációra vonatkozóan, ha adatbányászati eszközökkel alaposabban megvizsgáljuk a gesztus-adatbázist. Mindenféleképpen szükséges az adatbázisba új gesztus-osztályokat felvenni, ezáltal némi ráfordítással elkészíthető egy olyan rendszer, mellyel fejmozgás segítségével lehet például szavakat táplálni a számítógépbe. Végül, de nem utolsó sorban az elért eredményeket meg lehet próbálni felhasználni a szándékos és a nem-szándékos fejmozgások vizsgálatára is.

Köszönetnyilvánítás

A publikáció elkészítését a TÁMOP-4.2.2.C-11/1/KONV-2012-0001 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Irodalomjegyzék

[1] T. Acharya S. Mitra, "Gesture recognition: a survey," *IEEE Trans. on Systems, Man and Cybernetics*, pp. 311–324, 2007.

[2] O. Bernier, D. Collobert S. Marcel, "Hand gesture recognition using input–output hidden Markov models," *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 456-461, 2000.

[3] G. Qian, T. Ingalls, J. James S. Rajko, "Real-time gesture recognition with minimal training requirements and on-line learning," *CVPR '07. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[4] V. Shet, Y. Yacoob, L.S. Davis A. Elgammal, "Learning dynamics for exemplar-based gesture recognition," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, pp. 16-22, 2003.

[5] A. Quattoni, L.P. Morency, D. Demirdjian, T. Darrell S. Wang, "Hidden conditional random fields for gesture recognition," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1521-1527 2006.

[6] M. Turk, T.S. Huang P. Hong, "Gesture modeling and recognition using finite state machines," *FG*, pp. 410–415, 2000.

[7] B. Sin, S. Lee H. Suk, "Recognizing hand gestures using dynamic bayesian network," *8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1-6, 2008.

[8] V. Rabaud, G. Cottrell, S. Belongie P. Dollár, "Behavior recognition via sparse spatio-temporal features," *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65-72, 2005.

[9] T. Tuytelaars, L.J.V. Gool G. Willems, "An efficient dense and scale-invariant spatio-temporal interest point detector," *ECCV '08 Proceedings of the 10th European Conference on Computer Vision*, pp. 650-663, 2008.

[10] A. Ravichandran, G. Hager, R. Vidal R. Chaudhry, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1932–1939, 2009.

[11] J. Ahmed, M. Shah M.D. Rodriguez, "Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[12] T. Lindeberg I. Laptev, "Space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107-123, 2003.

[13] G. Bradski J. Davis, "Motion segmentation and pose recognition with motion history gradients," *IEEE Workshop on Applications of Computer Vision*, pp. 238-244, 2000.

[14] T. Drummond E. Rosten, "Machine learning for high-speed corner detection," *Proceedings of the 9th European conference on Computer Vision*, pp. 430-443, 2006.

[15] T. Kanade B. Lucas, "An Iterative Image Registration Technique with an Application to Stereo Vision," *7th International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

[16] M. Müller, *Information Retrieval for Music and Motion*, 1st ed.: Springer, 2007.