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 és megértésére, azok rögzítésére és később adatbányászati eszközökkel történő elemzésére, valamint a már rögzített mozdulatsorok segítségével a felismerés online javítására.
Ugyanakkor az elkészült rendszer illeszkedik egy multimodális ember-gép kommunikációt leíró modellbe is, mert használata új fejezetet nyit a metakommunikációhoz tartozó csatornák vizsgálatában.

Kulcsszavak: fejmozgás, gesztus felismerés, mozgás reprezentáció, dinamikus idővetemítés.

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 lehet 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 a gyakran lehet félreértések forrása, 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, ún. non-verbális jelrendszert alkalmazzuk. A non-verbális jelek tipikus megnyilvánulásai a mimika, a tekintet – szemkontaktus – szemmozgás, az ún. 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 stb. A kéz- és karmozgások jelentése: a hívás, elutasítás, tiltakozás, kérés, könyörgés, fenyegetés, köszöntés stb. 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 lehetőségünk nyílik tudatos fejmozgások felismerésére és megértésére, azok rögzítésére és később adatbányászati eszközökkel történő elemzésére, valamint a már rögzített mozdulatsorok segítségével a felismerés online 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: a modell és minta alapú módszerekre. A modell alapú eljárások csoportjába a különböző rejtett Markov modellek (HMM) és azok variánsai. Marcel et al. [2] egy input-output HMM-et készített EM algoritmus használatával, melyet később a kézfej körvonalából kinyerhető gesztusok felismerésére alkalmazott. A szakirodalomban létezik a tradicionális HMM néhány javítása is, melyek pl. szemantikus hálókkal tökéletesítenek [3], vagy a nem-paraméteres HMM-ek [4], illetve a feltételes valószínűségi mezők (Hidden Conditional Random Field) [5]. Ezen variánsok egyszerre csökkentik a tanítás költségét és az osztályozás pontosságát.

További népszerű modellek a véges állapotú gépek [6], valamint a dinamikus Bayes hálók [7]. Ezen eljárások feltételezik, hogy a fej mozgásának trajektóriája és ez által az artikuláció ismert. Habár ezekkel az eljárásokkal ígéretes eredményekre lehet szert tenni, a robusztusságuk nagyban függ az arc detektálásának és a mozgás követésének sikerességétől. Továbbá használatukat megelőzően sok adatra és számításigényes eljárások alkalmazására van szükség.

Ezzel szemben a minta alapú eljárások alkalmazásával elkerülhető a modell alapú módszerekben rejlő nehézségek nagy része. Mindez az egyes gesztusok – megjelenésére vonatkozóan – invariáns reprezentálásával és azok közvetlen egymáshoz illesztésével érhető el. A meglévő módszerekben leggyakrabban tér-, és időbeli jellemzőket vagy leírókat használnak [8,9,10,11]. Laptov et al. [12] megalkotta a referencia eljárást az irányított gradiensek (HOG) és optikai áramlás hisztogramjai (HOF) – mint leírók – alkalmazásával az érdekes pontok kinyerésére és a gesztusok felismerésére.

Egyes leírók magukban foglalják a mozgás trajektóriáját, tér-, és időbeli gradienseket, és az optikai áramláshoz tartozó globális hisztogramokat [10]. Ezen eljárások legnagyobb hátránya, hogy a futás során közvetlenül illesztik az egyes gesztusokat egy már meglévő adatbázisra, mely rontja az eljárások skálázhatóságát.

2 Fejmozgás reprezentálása

Ebben a fejezetben ismertetjük részletesen az általunk kifejlesztett minta alapú gesztusfelismerő rendszert. Mindemellett megadunk egy hatékony vizuális reprezentációt a mozgást meghatározó jellemzők kinyeréséhez, amely elengedhetetlen a felismerő rendszer nagyméretű gesztus adatbázison történő használatát illetően. Ennek kapcsán bevezettünk egy új és hatékony vizuális reprezentációt a fejmozgásból kinyerhető gesztusok felismerésére vonatkozóan, mely a mozgás menetét ábrázoló képen alapul.

Ezen a képen egy egyszerű FAST sarokdetektorral meghatározzuk azokat a régiókat, melyeken a mozgás a legmeghatározóbb volt. Majd egy adott gesztus sorozat minden szomszédos képkockájára kiszámoljuk az előbb kinyert régiókhoz tartozó optikai áramláshoz tartozó vektorokat és ezek alapján a globális fejmozgáshoz tartozó irányvektorokat. Ennek eredményeként egy-egy darab irányvektort kapunk a gesztus sorozat minden szomszédos képkocka párjára. Végezetül a gesztus sorozathoz tartozó irányvektorok sorozatát dinamikus idővetemítés segítségével egy előre definiált gesztus adatbázis elemeihez hasonlítjuk.

2.1 Mozgás ábrázolása

A mozgás megjelenítésére azt az irodalomban előszeretettel használt módszert alkalmaztuk, mely egy képet hoz létre a mozgás történetére vonatkozóan (Motion History Image - MHI) [13]. Ez egy időalapú sablonozó eljárás, mely nagyon egyszerű, de ugyanakkor robusztus reprezentációt szolgáltat a mozgó objektumokra.

Rengeteg variánsa létezik és a szakirodalomban szinte megszámlálhatatlan tanulmányt találunk a felhasználására [14]. Ebben az alfejezetben csak a módszer lényegét ismertetjük. Bobick és Davis [15] vezette be először azt a reprezentációt a mozgás alapú gesztusok felismerésében mely külön írja le, hogy „hol” és hogy „hogyan” történik a mozgás a képen. Egy úgynevezett mozgási energiát ábrázoló bináris képet (MEI) alkottak meg, mely arra vonatkozóan tartalmaz információt, hogy hol volt mozgás egy kép szekvencián. A MEI lényegében a mozgás alakját és térbeli felosztását írja le.

A metódushoz szükség lesz még egy MHI sablon létrehozására is, amelyben minden egyes pixel a mozgásnak egy sűrűségfüggvényeként értelmezhető az adott helyen. A MEI és MHI sablonok együttesen egy kétkomponensű, az időtől is függő sablonként – vektor értékű képként – értelmezhetőek, amelyben minden egyes pixel értéke a mozgás egy függvénye az adott pixel helyén. Az eljárás az alábbi (1) képlettel számolható:

$$MHI(x,y) = \begin{cases} \delta, ha\ M(x,y) \neq 0, \\ 0, ha\ M(x,y) = 0\ vagy\ M(x,y) < \delta - \tau, \\ MHI(x,y), különben. \end{cases} \quad (1)$$

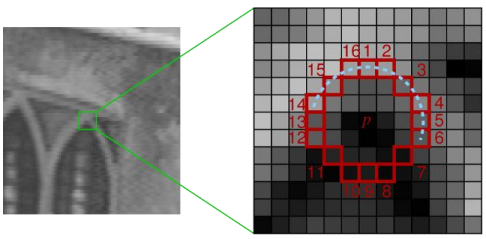
Ahol M függvény egy bináris maszk, mely nem-nulla értékű pixeleket tartalmaz ott ahol mozgás volt a képen, δ a kép szekvencia aktuális időbélyege, τ pedig a maximális értéke a mozgáskövetésnek. Vagyis az MHI-val jelölt kép összes olyan

pixele, ahol mozgás volt δ értéket fog felvenni, még azok a részek ahol nem volt mozgás fokozatosan elhalványulnak és végül törlődnek. Az eljárás grafikus reprezentációját az 1. ábra mutatja. Az MHI eljárást a mozgás szegmentálásában is felhasználtuk. Maga a gesztus képkockák sorozata, vagyis a videó folyam szegmensei alatt fog megjelenni. A szegmensre teljesülnie kell a következő feltételnek: olyan nyitó és záró képkockák tartoznak hozzá, melyeknek a számított MHI képek átlagintenzitás értéke alacsonyabb, mint egy előre definiált küszöbérték. Vagyis olyan képkockák határolják a szegmenset, ahol a mozgás intenzitása alacsony volt.

1. ábra. A bal oldali ábrán egy mozdulatsorhoz tartozó MEI sablon látható, a jobb oldalin pedig a hozzá tartozó MHI.

2.2 A mozgást meghatározó régiók

Következő lépésben megkeressük azokat a régiókat az MHI-n, melyen meghatározóak a fejmozgásban. Erre a FAST algoritmust [16] használtuk, mely egy egyszerű sarokdetektor – jellemző pontok kinyerésére. Hatékonysága az alacsony számításigényében rejlik, mely által valós idejű feldolgozásra alkalmas. Veszi a kép minden 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 egy szűkebb környezetben, ezért szokás egy metrikát alkalmazni a sarkok erősségének mérésére. Általában egy kétmenetes algoritmusként implementálják, mely rendkívüli gyors számítást tesz lehetővé.



2. ábra. A FAST detektor által vizsgált tartomány egy potenciális sarokpont esetén. A FAST detektor is egy vizsgált pont körüli kör mentén – például egy 3 sugarú, 16 kerületű – vizsgálódik, ha ebből valahány – például 9 – eltér a pixelnél legalább egy küszöbbel magasabb értékkel, akkor az adott középpont egy jellemző pont.

2.3 Optikai áramlás

A FAST algoritmus jellemzőpontok egy halmazával fog visszatérni a MHI-n. A következő lépésben ezen jellemzőpontokra számítjuk ki az optikai áramláshoz tartozó vektorokat az aktuális képkockára és arra, amelynek az időbélyege megegyezik az MHI időbélyegével, vagyis az aktuális képkocka megelőzőjével.

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. Az optikai folyamon tehát azt értjük, ahogy a képintenzitások mozgása megjelenik egymás utáni képeken. Különböző típusú képbemenetekhez az egyes optikai folyam algoritmusok más-más eredményt adhatnak, ezért célszerű a bemeneti adatok milyenségének figyelembevételével választani a lehetséges algoritmusok közül, hogy a kapott vektormező minél jobban közelítse a képeken látható objektumok valós fizikai mozgását. Inputként a videó egymáshoz közeli képkockáit szokás megadni.

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:

$$-\frac{\partial \mathrm{I}}{\partial \mathrm{t}} = \frac{\partial \mathrm{I}}{\partial \mathrm{x}} \frac{\mathrm{dx}}{\mathrm{dt}} + \frac{\partial \mathrm{I}}{\partial \mathrm{y}} \frac{\mathrm{dy}}{\mathrm{dt}}. \tag{2}$$

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 teljesség igénye nélkül az alábbi technikák a legelterjedtebbek.

A differenciális módszerek

Régebbi technológiák, de megbízhatóak. Az újabbaknak nem sikerült jelentős minőségjavulást hozniuk. Ilyen pl. a Horn–Schunck féle módszer [17], mely ugyan az objektumok határát nem kezeli és egy úgynevezett egyenletességi korlátozással egészíti ki az optikai folyam egyenletet. Felteszi, hogy a szomszédos pontok közel azonos sebességgel mozognak, így nincs hirtelen ugrás bennük. Ezek alapján olyan vektorokat keresünk, amelyek egymáshoz képest a legkevesebbet változnak (és persze kielégítik az optikai folyamkorlátozást).

Másik népszerű módszer a Lucas–Kanade módszer [18], mely ellentétben az előző eljárással egy pont sebességére úgy tekint, hogy az csak a pont helyi környezetétől függ (lokális technika). Másképpen, a kiindulási feltétel az, hogy a kép kisméretű szegmenseiben a sebesség állandó értékű, így egyetlen sebességvektorhoz több egyenlet is keletkezhet. Az optikai folyam korlátozás kiegészítése a konstans lokális sebesség módszerével azt a feltételt jelenti, hogy a vizsgált pixel és a

szomszédságában lévő képpontok sebessége ugyanaz. A szakirodalomban gyakran használják ennek a módszernek és ennek piramisos változatát is.

Korrelációs technikák
A módszer elve [19], hogy a videó szomszédos képkockáin olyan blokkokat keres, melyek pixelei nagyjából megegyeznek. A blokkok egymással fedésben lehetnek, és ha két egymást követő képkockán az algoritmus megtalálta ugyanazt a részt, akkor az ezek közti elmozdulás adja az optikai folyam vektorát. Az algoritmus feltételezi, hogy az adott blokkon belül az összes pixel ugyanazzal a vektorral mozdult el. Mivel a módszer az elmozdulásokat csak a blokkok közt keresi, azért így határozza meg minden egyes pixelre az optikai folyam vektort. Minél nagyobb a blokkméret, annál kevesebb optikai folyam vektort kapunk az adott képre. Az algoritmus nagyobb blokkméret esetén kevésbé érzékeny a zajra, és pontosabb, mivel egy vektor előállításához több pixelt vizsgál.

2.4 Fejmozgás iránya

Mint ahogy az előző alfejezet elején már említettük a bejövő videó folyamon a FAST sarokdetektor segítségével és a MHI képhez meghatározunk egy maszkot, melynek a fehér színű képpontjai felett kiszámoljuk az optikai áramláshoz tartozó vektorokat minden egyes képpárra, lásd 3. ábra. Az optikai áramláshoz tartozó vektorokat a Lukas–Kanade eljárás GPU-ra optimalizált változatával határozzuk meg.

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

A gesztus felismeréséhez szükség van egy reprezentációra mellyel egy adott szegmenshez tartozó fejmozgás leírható az optikai folyamhoz tartozó vektorok – melyet a szegmens minden képpárjához kiszámolunk – függvényében. Ha a FAST eljárás N darab jellemzőpontot detektált, akkor ugyanennyi optikai folyam vektor fog keletkezni egy képpár között. A gesztusok könnyebb definiálása és felismerése érdekében meghatározzuk egy átlagvektort, melyet az N darab optikai folyam vektor számtani közepeként kapunk meg. Meghatározzuk az átlagvektornak az Y tengely pozitív oldalával bezárt szögét a (3)-as egyenlet segítségével.

$$\text{radián} = \text{atan2}\left(x_{\text{kezdő}} - x_{\text{vég}}, y_{\text{vég}} - y_{\text{kezdő}}\right). \quad (3)$$

Ahol $\mathrm{atan2}(x, y)$ függvény 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 körnegyedbe esik. A (4)-es egyenletek segítségével az eredményt átváltjuk szögbe és gondoskodunk arról, hogy a koordinátarendszerünk jobbsodrású legyen.

$$\begin{aligned} \alpha &= (radián \times 180°)/\pi \\ \alpha &= (360° - \alpha)\ mod\ 360° \end{aligned} \tag{4}$$

Így a vizsgált szegmens minden egyes képpárjára kapunk egy $\alpha \in \{0°, 360°\}$ szöget, melyekhez értékétől függően egy címkét rendelünk az (5)-ös formulában definiált $f(\alpha)$ függvény segítségével.

$$f(\alpha) = k + 1 * \frac{360°}{16} | \alpha \in \left\{ k * \frac{360°}{16}; k + 1 * \frac{360°}{16} \right\}, \text{ahol } k = 0, \dots, 15. \tag{5}$$

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

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

3 Gesztusfelismerés

Ebben a fejezetben egy eljárást ismertetünk, az előző fejezetben definiált $\{f^k(\alpha_0), f^k(\alpha_1), \dots, f^k(\alpha_n)\}$ címkesorozatok egymáshoz történő illesztésére vonatkozóan. Az illesztéshez szükség lesz egy előre definiált gesztus adatbázisra. Ebben az adatbázisban olyan elempárokat tárolunk, melyekben benne van a gesztus neve és az ahhoz egy előzetesen meghatározott címkesorozat, pl. a (6)-os formulában definiált rekord, melyben szögletes zárójelek között soroljuk fel a *fejrázás* gesztus egyik címkesorozatát.

$$\{\text{fejrázás}; [90,90,90,90,270,270,270,225,135]\}. \tag{6}$$

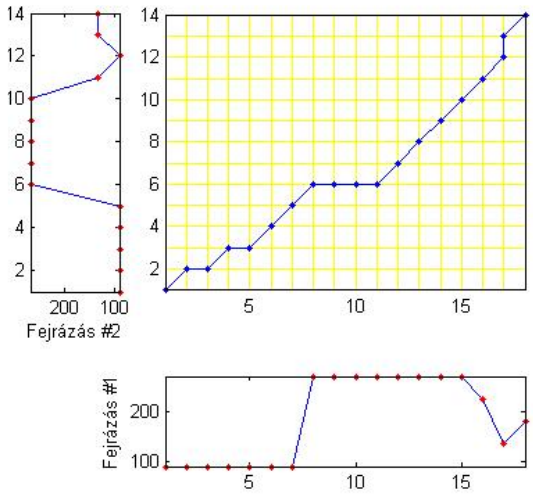
A gesztusfelismerés során folyamatosan szegmentáljuk a videó folyamot és számítjuk a szegmensekhez tartozó címkesorozatokat. Az egyes címkesorozatokat a dinamikus idővetemítés (Dynamic Time Warping - DTW) [20] eljárás segítségével illesztjük az előre definiált gesztus adatbázis elemeihez. Az adatbázisban egy gesztushoz több sorozat is létezhet, így az adatbázison belül a gesztusok csoportokat alkotnak.

Végezetül a szegmenshez tartozó címkesorozatot ahhoz a gesztus-csoporthoz fogjuk besorolni, melyhez a DTW átlagosan a legkisebb távolságot adta.

3.1 Dinamikus idővetemítés

A DTW feladata hogy, azonos időtengelyre vetítse az aktuálisan detektált és a tárolt fejmozgást, hogy címkesorozatot összevethessük a tárolt referenciákkal (gesztus-csoportokkal). 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 (vektorokat) egymással összehasonlítja, és a távolság minimalizálására törekszik. Az eljárás során a felismerendő címkesorozatot az összes gesztus-csoport összes referenciamintájával össze kell hasonlítani, és a legkisebb távolságú elem lesz a felismerés eredménye.

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



5. ábra. Egy futás alatti fejrázás gesztus illesztése az adatbázis egy fejrázás csoportjába tartozó elmére. A lineáris illesztést a koordinátarendszer $(\mathbf{0}, \mathbf{0})$ pontjából induló és $(\mathbf{18}, \mathbf{14})$ pontjában végződő szakasz jelentené. Az optimális nem lineáris illesztést a kék törött vonal jelzi. A DTW által adott távolság a két sorozatra: 5,60081.

A vetemítőgörbe (az ábrán a kék törött vonal) 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.

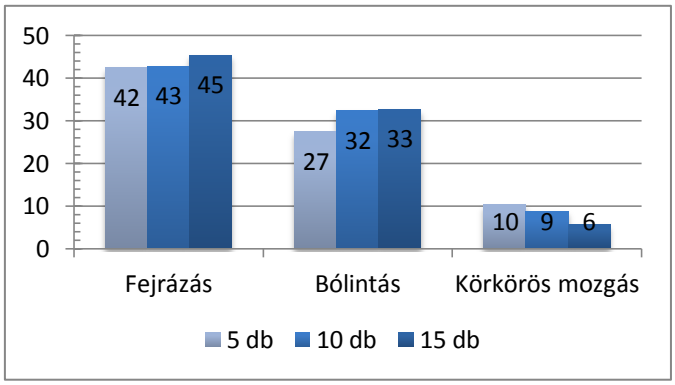
3.2 Mozgás adatbázis

Az előző fejezetben már tárgyaltuk, hogy a videó folyam szegmenseihez számított címkesorozatokat egy előre definiált adatbázis elemeihez illesztjük. Ebben az adatbázisban olyan elempárokat tárolunk, melyekben benne van a gesztus neve és az ahhoz egy előzetesen meghatározott címkesorozat, pl. a (6)-os formulában definiált rekord, melyben szögletes zárójelek között soroljuk fel a *fejrázás* gesztus egyik címkesorozatát.

Az illesztés jósága nyilvánvalóan függ az adatbázisban tárolt rekordok darabszámától, illetve azok eloszlásától. A rendszer jelenleg úgy működik, hogy ha futás során a k. szegmenshez tartozó címkesorozatot pl. *fejrázásnak* osztályozta, akkor új elemként felveszi az adatbázis *fejrázás* osztályába ezt a címkesorozatot, így a következő szegmens osztályozása során már figyelembe veszi ezt az új információt is.

4 Kísérletek és eredmények

A rendszer jelenleg még nincs kész teljesen. Egyelőre nyitott kérdés, hogy az előre rögzített gesztus csoportok számosságát, hogyan lenne célszerű megválasztani, azonban az alábbi grafikonból (6. ábra) jól látszik, hogy bizonyos határok között érdemes az adatbázist online bővíteni a felismerés során. Erre vonatkozóan egy tesztet végeztünk el, az egyszerűség kedvéért három különböző gesztus csoport számosságát vizsgáltuk: a fejrázását, a bólintását és a körkörös fejmozgását. Futási időben a körkörös fejmozgáshoz tartozó gesztusokat hasonlítottunk az előbbi három osztály elemeihez.



6. ábra. DTW átlagos eredménye 20 darab futási időben végzett gesztusra.

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

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

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 a csoportokhoz előzetesen harminc darab gesztust definiáltunk kategóriánként az adatbázisban. A futási idejű felismerés, azaz a csoportok szeparálása közel 100%-os, az egyes gesztusok és adatbázisbeli gesztuscsoportokra átlagosan olyan DTW távolságok adódnak, mint amit a 6. ábra is mutat. Egy gesztus felismerése 18-20 ms-nyi időt vesz igénybe.

4.1 Összefoglalás

A jelenlegi rendszer kétségkívül bebizonyította, hogy alkalmas valós idejű fejmozgás alapú gesztusok felismerésére, azonban pár nyitott kérdést is megfogalmazott. Ilyen pl. az gesztus adatbázisbeli csoportok méretének ésszerű meghatározása. További érdekes feladat lehet egy olyan összetettebb metrika készítése, mely a futás során a mozgás szegmensekből kinyert gesztusokat nem csak azoknak az adatbázis gesztus csoportjaitól vett átlagos távolsága alapján osztályozza, hanem egyéb az adatbázisból kinyert információt is alkalmaz a folyamat során. Esetlegesen csökkenti az adatbázisbeli elemekkel való illesztések számát, meghatározza, hogy melyekhez célszerű illeszteni a vizsgált sorozatot.

Véleményem szerint további érdekes törvényszerűségeket – akár egyénre, akár az összes emberre – vonatkozó lehet felfedezni az emberi gesztikulációra vonatkozóan, ha adatbányászati eszközökkel alaposabban megvizsgáljuk az online módon bővített adatbázist. Továbbá mindenféleképpen szükséges az adatbázisba új gesztuscsoportokat felvenni, némi ráfordítással elkészíthető egy olyan rendszer, mellyel fejmozgás segítségével lehet 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.

Referencia

[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] J. K. Tan, H. Kim, S. Ishikawa A. R. Ahad, "Motion history image: its variants and applications," *Machine Vision and Applications*, vol. 23, no. 2, pp. 255-281, 2012.

[15] J. Davis A. Bobick, "An appearance-based representation of action," *Proceedings of the 13th International Conference on Pattern Recognition*, pp. 307–312, 1996.

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

[17] B. Schunck B. Horn, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185-204, 1981.

[18] 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.

[19] T. D. Tran, R. Etienne-Cummings Y. M. Chi, "Optical Flow Approximation of Sub-Pixel Accurate Block Matching for Video Coding," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 1017–1020, 2007.

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