Az emberi fej térbeli helyzetének és a tekintet irányának meghatározása

Bertók Kornél, Fazekas Attila, Sajó Levente

Debreceni Egyetem

Informatikai Kar

Debreceni Képfeldolgozó Csoport

H-4010 Debrecen, Pf.:12.

[bkornel@gmail.com](mailto:bkornel@gmail.com)

[attila.fazekas@inf.unideb.hu](mailto:attila.fazekas@inf.unideb.hu)

[sajolevente@inf.unideb.hu](mailto:sajolevente@inf.unideb.hu)

**Absztrakt.** A cikk témája egy multimodális ember-gép kommunikációt leíró elméleti modell egy moduljának gyakorlati megvalósítása. A modell az eddigi gyakorlatnál emberközelibb módon ragadja meg a számítógép és a felhasználó közötti interakciót. Ezt úgy kívánja elérni, hogy tanulmányozza az ember-ember kommunikáció olyan alapvető jellemzőit, amelyek technológiailag egyrészt relevánsak, másrészt pedig megvalósíthatóak. Az így szerzett elméleti tudásra támaszkodva egy új modellt állít fel az ember-gép interakció tetszőleges és változatos formáira. Ebbe a sorba illeszkedik be az ember – mint kommunikációs fél – fejének térbeli helyzetének, illetve azon belül a tekintetének számítógépes meghatározása.

**Kulcsszavak:** pozíciószámítás, POSIT, szivárványhártya detektálás, tekintet követés.

# Bevezetés

1963-ban a Stanford Kutatóintézetben Douglas Engelbart felvetette egy új beviteli eszköz gondolatát. Mára ez az egérként elhíresült eszköz mindenki számára ismert és magától értetődően használható lett, de bevezetése óta nem igazán történt előrelépés a hétköznapi használatra szánt számítógépes perifériák piacán. Mára a technika fejlődésével megjelent az igény a számítógépek könnyű használhatóságára. Szükségessé vált, hogy az ember ne csak közvetve a kezével tudjon kommunikálni a számítógéppel. Újfajta ember-számítógép interfészek megjelenésére van szükség. Az ember-számítógép interakció kutatási feladatai közé tartozik, hogy olyan új, alternatív kommunikációs (adat ki- és beviteli) eszközöket és módszereket fejlesszen, amelyek segítik az ember-gép kapcsolatot az ember számára minél természetesebbé, magától értetődővé tenni. A mai luxusautókban a fedélzeti számítógépek már megértik a szóbeli utasításokat, sőt akár intelligens kivetítőkre is lehet ujjal rajzolni. Azonban az efféle interfészek nem csak szórakozásra használhatók, megkönnyíthetik a fogyatékkal élők életét is, mint például a szemkövető, vagy jelbeszéd-felismerő rendszerek.

Az előbbiek miatt a mára már kihagyhatatlan perifériának számító kamera is egy fontos beviteli eszközként használható. Képnézegetők, játékok, web-böngészők irányítását lehet megkönnyíteni a kamera képéből kinyerhető mozgásinformáció felhasználásával. Nyugodtan kijelenthetjük, hogy ez a technika – vagyis a felhasználó webkamerával történő megfigyelése – egészen új lehetőségeket teremtett az ember-számítógép kommunikációban.

Ebben a cikkben az ember-számítógép kommunikáció egy újszerű felhasználói eszközével, a felhasználó, vagyis az emberi fél figyelmének számítógépes elemzésével foglalkozom. A rendszer webkamerák által szolgáltatott 800x600 vagy magasabb felbontású videó folyamokon határozza meg a fej térbeli pozíciójához tartozó forgatási mátrixot és eltolási vektort (3. Fejezet), valamint a tekintet esetében a pupilla koordinátáit (4. Fejezet). Egy speciális eljárást dolgoztunk ki arra vonatkozólag, hogy a címben megjelölt feladatokat a lehető legtöbb képkockán, valós időben tudjuk megoldani. Mivel mind a fej térbeli helyzetére, mind a tekintet irányának meghatározására vonatkozó eljárásunk egy térbeli fejmodellen alapszik, így a térbeli alappontokhoz tartozó webkamera képeken megjelenő arci karakterisztikus pontokat nagy biztonsággal és gyorsasággal kell detektálnunk (2. Fejezet).

# Arci jellemzők detektálása és követése

A rendszerünk sikeres működéséhez elengedhetetlen egy nagy megbízhatóságú, robosztus objektumdetektáló rendszer. A robosztusságon azt értjük, hogy a rendszernek érzéketlennek kell lennie a beérkező kép minőségére.

Az IPGD csoportban 2008. első felében az Intel OpenCV [1] szabadon felhasználható képfeldolgozó könyvtár segítségével elkészítettünk egy olyan szoftver eszközt, amely a gyakorlati kísérletek során bebizonyította, hogy megfelel az előbbi bekezdésben megfogalmazott elvárásainknak.

Az alábbiakban ennek a rendszernek az elméleti hátterét tekintjük át a hozzá kapcsolódó algoritmusokkal együtt. Ezzel a komponenssel határozzuk meg a fej térbeli helyzetének és a tekintet irányának megállapításához szükséges arci karakterisztikus pontok koordinátáit a bemeneti képen.

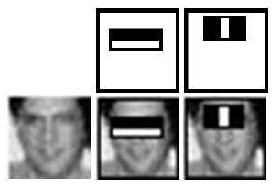
## Viola-Jones detektorok

Paul Viola és Michael J. Jones egy olyan frontális arcdetektor rendszert alakítottak ki [2], mely eléri az előtte publikált legjobb eredmények [3,4,5,6,7] találati és hamis pozitív arányát. Az arcdetektor mintájára szem-, száj-, orr-, szivárványhártya detektorokat készítettünk az OpenCV könyvtár segítségével, melyben Viola és Jones közzétették arcdetektorukat és az implementált tanítási algoritmust.

Arcdetektáló rendszerük tisztán kiemelkedik az eddigi megközelítések közül gyorsaságában. Valós időben 30 kép/másodperc teljesítményt érhetünk el, egy 1.6 GHz-es Intel Pentium M processzorral rendelkező notebookon, 640x480 pixel felbontású kameraképeken. A Viola-Jones detektorok csak szürkeskálás képekből kapott információk alapján képesek magas képfeldolgozási arányt elérni. A detektorok felépítését az alább kifejtett három fő témakör köré csoportosíthatjuk.

### Az integrált képreprezentáció bevezetése.

A rendszer a kép intenzitásértékei helyett képrégiók jellemzőivel dolgozik [8]. Ezek a jellemzők a Haar-féle bázisfüggvényekre emlékeztetnek, bár néhány esetben ezeknél valamivel bonyolultabbak. A jellemzők többféle méret melletti gyors kiszámítása érdekében vezetjük be az integrál képet, amely az eredeti képből képpontonként néhány elemi művelettel – tehát rendkívül gyorsan és hatékonyan – előállítható. Az integrál képreprezentáció ismeretében a jellemzők értéke (lásd 1. ábra) bármely pozícióban vagy skálázás mellett konstans időben meghatározható.



1. ábra: Az AdaBoost által választott első és második jellemző. Értékük úgy számolódik, hogy a fehér téglalapon belüli pixelek összegét kivonjuk a feketén belüliek összegéből.

### AdaBoost algoritmus

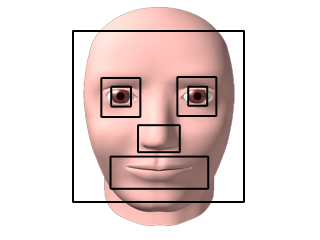
A második fontos tulajdonság egy osztályozó létrehozásának folyamata, a fontos jellemzők egy kis halmazának kiválasztásával, az AdaBoost-on alapuló tanuló algoritmust használva [9]. Egy kép részablakán belül az összes Haar jellemzők száma nagyon nagy, sokkal nagyobb, mint a pixelek száma. Azért, hogy a gyors osztályozást biztosítsuk, a tanuló folyamatnak ki kell zárnia az elérhető jellemzők nagy többségét, és a kritikus jellemzők kis halmazára kell koncentrálnia. Tieu és Viola munkája által motiváltan, a jellemző kiválasztás az AdaBoost egy egyszerű módosítása lett: minden gyenge osztályozót kényszerítenek, hogy annak eredménye csak egy jellemzőtől függjön [10]. Ennek eredményeként a Boost folyamat minden köre, amely új gyenge osztályozót választ, megfelel egy jellemző kiválasztó folyamatnak [11,12,8].

### Kaszkád struktúra

Az utolsó lépés, hogy az egyre bonyolultabb osztályozókat láncba, egy vízesésszerű sémába rendezzük. Az ötlet azon az észrevételen alapul, hogy gyakran egyszerű azt eldönteni, hogy egy objektum hol fordulhat elő, így a bonyolultabb feldolgozást csak ezeken a „bíztató” területeken kell elvégezni [13,14,7]. Az osztályozók teljesítményének fontos mértéke az úgynevezett „hamis negatív” találati arány, vagyis azon részablakok aránya, amelyeket elutasítunk, annak ellenére, hogy tartalmazzák a keresett objektumot. Ideális esetben a képen ténylegesen szereplő összes célobjektum régióját meg kell találni.

## Detektorok hierarchiája a programon belül

Az arc, orr, száj, balszem, jobbszem és szivárványhártya detektorok csak a kép egyes területein futnak csak le (lásd 2. ábra). A detektorok téglalapok sorozatával térnek vissza, melyek a keresett objektumot tartalmazhatják. Míg az arcdetektor a teljes képkockán lefut, addig a szemdetektor csak az arcdetektor felső felének eredményét dolgozza fel, amennyiben az arcdetektor talált eredményt. Értelemszerűen a balszem detektor csak az arc felső felének bal oldalán fut le, a jobbszem detektor pedig csak a jobb oldalán. Ennek megfelelően az orrdetektor csak az arcdetektor által szolgáltatott ablak közepén fog lefutni, a szájdetektor pedig csak az ablak alsó felében fog keresni. Ha az arcdetektor nem talált eredményt, akkor egyetlen másik detektor sem fog lefutni az aktuális képkockán.



2. ábra: Objektumdetektálás egy 800x600 pixel felbontású képen. Az egyes detektorok értelmezési tartományát befolyásolhatja a többi detektor által visszaadott ablak.

A szemdetektor eredménylistájából már nem a legutolsó és egyben legnagyobb eredményét választom ki a további szivárványhártya kereséshez, hanem azok közül az eredmények közül választom a legnagyobbat, melyekre teljesülnek az alábbi feltételek:

* A szemdetektor által talált téglalap szélessége legyen nagyobb az arcdetektor eredmény szélességének 30%-ánál.
* A szemdetektor által talált téglalap az arcdetektor eredményének felső felében legyen, mivel egy emberi arcon a szemek az arc felső felében helyezkednek el.

Ha van eredménye a szemdetektoroknak, akkor lefuttatom a szivárványhártya detektorokat azon belül. Ha az egyik oldali szemdetektor nem talál eredményt, akkor a képrészletet vertikálisan tükrözzük, és a másik oldali szemdetektort futtatjuk rajta, majd ezek után próbálunk szivárványhártyát detektálni.

A szemkövető program teljes detektorhierarchiáját minden beérkező képen futtatva 25 kép/másodperc teljesítményt érhetünk el, egy 2.8 GHz-es Intel Pentium C2Q processzorral rendelkező PC-n, egy 800x600 pixel felbontású webkamera képen.

A program megfelelő futásához azonban elég, ha csak minden ötödik képen futtatjuk a program teljes detektor hierarchiáját, így látható, hogy a valós idejű teljesítményt gond nélkül elérhetjük akár jóval kisebb teljesítményű számítógépeken is.

## Az arci jellemzők követése

Az előző fejezetben megadtunk egy módszert, mely segítségével magas detektálási arány és alacsony számítási idő mellett, lehetőségünk nyílik az arc és az egyes arci jellemzők detektálására. A fej térbeli helyzetének, illetve a tekintet irányának meghatározásához viszont minden egyes képkockákon tudnunk kell a szemek, a száj, az orr és a szivárványhártyák helyzetét. Könnyen belátható, hogy az előbbi pontok együttes detektálásának a valószínűsége alacsony. Így ha egyszer már detektáltunk egy pontot, akkor annak meg kell jósolnunk a helyzetét azokon a képkockákon, amelyeken a detektálás sikertelen volt.

Tehát vizsgálnunk kell a vizuálisan reprezentált tárgyak elmozdulását. Amit általában képpontokból eredő vektorokkal írunk le. Ebben a fejezetben a mozgást és azon belül a mozgás irányát próbáljuk meghatározni. Az alábbiakban látni fogjuk, hogyan lehet a feladatot hatékonyan és gyorsan megoldani. A mozgásérzékelés megvalósítására több módszer is létezik, például a fázis korreláció [15], mely azon alapul, hogy az eredeti képen az eltolás a frekvenciatérben fáziseltolódásként jelenik meg. Egy másik algoritmus az illesztő algoritmus [16], amely egy szabályosan kiválasztott képrészletekre alkalmazott egyszerű mintaillesztő algoritmus. Azonban mi a Lucas-Kanade módszernek egy módosított változatát választottuk, a Kanade-Lucas-Tomasi (továbbiakban csak KLT) követő algoritmust. A Lucas-Kanade módszer viszonylag régóta ismert [17]. Azon alapszik, hogy érdemes csak néhány előre meghatározott jellemző pontoknak kiszámítani a mozgását [18], így megvalósítása egyrészt gyengébb hardveren is lehetséges, másrészt a mi modellalapú rendszerünkbe is jól illeszkedik.

### KLT algoritmus

Az algoritmus leginkább abban különbözik a Lucas-Kanade módszertől, hogy egy új szimmetrikus számítási módszert vezettek be a mozgás számítására az egymást követő képekre [19]. Ennek köszönhetően az algoritmus használhatóvá vált valós időben is. A KLT algoritmus a Viola-Jones detektorokhoz hasonlóan csak szürkeárnyalatos képekre alkalmazható.

Eredeti célunk az volt, hogy a mozgásokat minden egyes képkockán észleljük. Ehhez meg kell találnunk az egyes tárgyak elmozdulását az előző képkockához képest. A megoldáshoz a mintaillesztés algoritmusát alkalmazzuk. Ehhez minden egyes képkockán ki kell választanunk egy négyzet alakú területet, majd a rákövetkező képkockán meg kell találnunk azt a területet, amely a leginkább hasonlít az eredetihez. Adott tehát az egyik képen egy minta, amit meg kell találni a rákövetkező képkockán. Amennyiben elég sűrűn mintavételezünk, akkor feltételezhetjük, hogy:

* Az adott minta a következő képkockán nem mozdul el jelentősen, tehát a pozíciója közel marad az eredeti pozícióhoz.
* A minta csak kicsit változik meg a két kép között, azaz nem távolodik–közeledik a kamerához, továbbá az elfordulás és torzulás hatása is elhanyagolható.

Van tehát egy mintánk, és a rákövetkező képen keressük azt a pozíciót, ahol a hasonló méretű és alakú minta a leginkább hasonlít az eredeti mintához. A mintákat több módon is össze lehet hasonlítani. Az egyik lehetséges módszer, amit mi is használtunk a minták új pozíciójának kiszámítására az, hogy a képből és annak gradienséből következtetünk a minta optimális helyére. Ebben az esetben tulajdonképpen egy Taylor-soros közelítés alapján számítjuk ki a minta elmozdulását [17].

A KLT algoritmus tehát úgy számolja ki az optimális helyet, hogy a megfelelő eltolás vektort alkalmazva az eredeti mintát odébb teszi (vagyis iterálja az eljárást), és arról a pozícióról próbálkozik újra, egészen addig, ameddig az elmozdulás kisebb lesz egy előre rögzített nagyon kicsi határértéknél.

A gradiens alapú összehasonlítás előnye, hogy viszonylag gyors, hátránya, hogy tévedhet, mivel nincsen garancia arra, hogy az elsőfokú Taylor-soros egyszerűsítés során nem veszik el olyan információ, amely félreviszi az eredményt. A gradiens alapú módszernek további előnye, hogy a perspektíva miatti deformációkat is tudja kezelni a számítási idő növekedése nélkül.

# Fej térbeli helyzetének meghatározása

Az ember-gép kommunikációban az emberi fej térbeli helyzetének ismerete fontos képesség a számítógép számára, mivel a gép ezáltal képet kaphat a kommunikációs partnere figyelmének középpontjáról, kommunikációban való részvételének hajlandóságáról, illetve annak egyéb belső állapotáról, viselkedésmódjáról. A fej térbeli helyzetének meghatározására az alább ismertetett POSIT eljárást használtuk fel. A eljáráshoz szükségünk van egyrészt a fej térbeli modelljét alkotó alappontokra (lásd 1. táblázat), a hozzájuk tartozó képsíkbeli – azaz webkamera képbeli – pontok koordinátáira, valamint a kamera néhány paraméterére.

|  |  |  |  |
| --- | --- | --- | --- |
| Koordináta | X | Y | Z |
| Orr | 0 | 0 | 0 |
| Bal szem | -15 | 15 | -10 |
| Jobb szem | 15 | 15 | -10 |
| Száj | 0 | -15 | -10 |

1. táblázat: A programban használt fejmodell koordinátái.

## Modell alapú pozíciódetektálás

Rögzített számú jellemző ponttal megadott tárgyak térbeli helyzetének vagy irányának ismerete fontos lehet egyes kalibrációs, térképészeti, vagy tárgykövető alkalmazás esetében. A pozíciódetektálás során a háromdimenziós euklideszi tér azon pontjait szeretnénk megadni, melyek csupán a kétdimenziós képsíkon ismertek számunkra. A problémát az alábbi módon formalizálhatjuk:

Legyen adva , (1)

a vizsgált objektumnak egy tetszőleges pontja a saját koordinátarendszerében.

Legyen továbbá , (2)

melyet úgy kaptunk meg, hogy -t vetítettük a képsíkra. Természetesen a mélységinformáció – a koordináta – a vetítés során eltűnik, így a normalizálás eredményeként kapott pontot célszerű átírni az alábbi formába:

. (3)

Értelemszerűen a térbeli és a képsíkbeli pontok függnek egymástól, tehát ha a modell egy pontja elmozdul, akkor ez változást fog előidézni a vetítésben is. A térbeli és a vetített pontok között az alábbi módon adhatjuk meg a kapcsolatot:

, (4)

ahol a forgatási-, és pedig az eltolási vektor. Tehát egy olyan eljárást szeretnénk definiálni, mely képes kiszámítani bizonyos alappontokkal megadott, webkamera képeken elhelyezkedő síkbeli objektumok térbeli helyzetét, vagyis az és a vektort.

Több eljárás is létezik erre vonatkozólag, mi a lentebb ismertetett POSIT eljárást használtuk fel. A továbbiakban feltételezzük, hogy ismerjük a vizsgált térbeli objektumnak a geometriáját, amit véges számú térbeli alapponttal adunk meg. Továbbá feltesszük azt is, hogy a webkamera képeken ismerjük négy vagy több olyan pontnak a koordinátáját, melyek a térbeli objektumon nem esnek egy síkba. Fontos azt is leszögezni, hogy az eljárás iteratív módon számolja ki térbeli pozíciót és csak a lineáris algebra eszközeiből építkezik. A POSIT működéséhez – nem úgy, mint a Newton-Raphson és az arra építkező eljárások esetén [20,21,22] – nem szükséges a kezdőpozíció rögzítése és a mátrixok invertálása az iteráció törzsében. A számításokhoz egy webkamera áll a rendelkezésünkre, tehát nem használhatjuk ki a sztereó látás előnyeit sem a mélységérzékelés során.

### Vetítési modell

A fej térbeli helyzetének kétdimenziós pontok alapján történő meghatározásához szükség van egy vetítési modellre. A gyakorlatban általában a perspektivikus, vagy gyenge perspektivikus vetítési modelleket szokták alkalmazni. Mint minden vetítés az előbb felsorolt kettő is egy dimenziócsökkentő művelet, tehát olyan transzformáció, amely n-dimenziós objektumokat kisebb dimenziós terekbe visz át. A vetítés eredménye a vetület, ami egy térbeli síkon, a vetítési síkon képződik. A tárgy-, és képpontokon átmenő egyenest vetítősugárnak nevezzük. Az egyes tárgypontok képe pedig a vetítősugár döféspontja a vetítési síkkal.

Amikor perspektív vetítésről beszélünk, akkor a vetítősugarak mindegyike áthalad egy vetítési középponton, a centrumponton. A perspektív vetítés az objektumok realisztikus ábrázolását teszi lehetővé, ilyenkor a távolabbi objektumok kisebbnek tűnnek, a vetítési síkkal nem párhuzamos egyenesek pedig egy pont felé tartanak. Perspektív vetítés esetén a látótér tulajdonképpen egy végtelen piramisként képzelhető el, amelynek csúcsa a nézőpont, vagy szem. A perspektivikus vetítés modellje egy képsíkból, valamint a vetítés középpontjából – az pontból – áll (lásd ). A vetítés középpontja és a képsík közötti távolság , amit gyújtótávolságnak is szokás hívni. Az -ból kiinduló és a képsíkot metsző egyenes az optikai tengely.

Untitled-3.tif

3. ábra: A perspektivikus vetítés térbeli modellje.

A perspektív vetítés legnagyobb problémája, hogy nem-lineáris transzformáció, így valós idejű használata nem kifejezetten javallott. Célszerű tehát közelíteni egy lineáris transzformációval, például a merőleges vetítéssel, ami a perspektíva határértékének is tekinthető az miatt. Az approximációra a gyenge perspektivikus vetítés modelljét használjuk fel, amely tulajdonképpen egy merőleges vetítésből és egy skálázásból áll. Továbbá csak akkor hajtható végre, ha fennáll az alábbi két előfeltétel:

* A vetíteni kívánt tárgy közel van az optikai tengelyhez.
* A tárgy mérete vagy kiterjedése kisebb, mint az átlagos távolsága a kamerától.

A gyenge perspektivikus vetítést értelmezhetjük úgy is, hogy a transzformáció során minden pontot ugyanazzal a faktorral skálázunk.

### POS algoritmus

A pozíció kiszámítására vonatkozó eljárásunk két metódus kombinációjaként jön létre. Az első a POS (Pose from Orthography and Scaling) mely az előbb ismertetett gyenge perspektivikus vetítésen alapul.

Untitled-1.tif

4. ábra: A kameramodell. a fejmodell egy pontja, a neki megfelelő perspektivikusan vetített pont a G képsíkon. pedig a gyenge perspektivikus vetítése -nek.

Konkretizálva a problémát, a 4. ábrán egy klasszikus kameramodell látható, melyen a vetítés középpontja , a képsíkot pedig -vel jelöltük. A képsík két tengelyét -et és -t tekinthetjük a kameraszenzornak, pedig az optikai tengelynek feleltethető meg. Továbbá azt is tudjuk, hogy a képsík távolságra van -tól. A tárgy-koordinátarendszer középpontja , a tengelyei pedig rendre , és . Mivel ismerjük a fej térbeli geometriáját, így az összes pont tárgy-koordinátarendszerbeli koordinátája ismert. A problémát az jelenti, hogy a fej térbeli pozíciója egyelőre még nem ismert, így az pontok kamera-koordinátarendszerbeli koordinátái sem ismertek számunkra. Jelöljük ezeket a koordinátákat (,,)-vel. E koordinátarendszerbeli koordináták kiszámításában lesz segítségünkre a POSIT algoritmus. Az ábrán csak az és pontokat szerepeltettük.

A forgatási mátrix kiszámításához a képsíkhoz tartozó és egységvektorokat kell felírni a tárgy-koordinátarendszerben. Az eltolási vektort pedig a tárgy-koordinátarendszerben felírt és vektorok keresztszorzataként kaphatjuk meg [23]. Tehát a POS algoritmus egy skálázott merőleges vetítéssel közelíti a gyenge perspektivikus vetítést. Fogalmazhatunk úgy is, hogy egy lineáris egyenletrendszer megoldásával számítja ki a forgatási mátrixot és az eltolási vektort.

### POSIT algoritmus

A POS algoritmus során tulajdonképpen az képsíkbeli pontokból kaptunk egy közelítést a tárgy pozíciójára vonatkozóan. Jelöljük a közelítő fejpozícióhoz tartozó térbeli pontokat -gal. Toljuk el ezeket az pontokat abba a pozícióba, amit egy helyes számítás eredményeként kaptunk volna. Majd a pontokhoz tartozó képsíkbeli pontokra futtassuk le újból a POS eljárást [23]. A POSIT eljárást négy-öt iteráción át futtatva megkapjuk a helyes pozíciót. Az algoritmus eredménye az 5. ábrán látható.

Untitled-4.tif

5. ábra: A fejpozíció, illetve a szivárványhártya detektálás kimenete. Az ábra két szélén láthatók a fej térbeli helyzetét szemléltető síkok. Ezeknek a kombinációja látható a középső ábrán, ahol a piros körök a szivárványhártya detektorok eredményeit jelentik.

# Tekintetkövetés

Az ember-gép kommunikációban a figyelem középpontjának vizsgálata során fontosabb az emberi fél nézési irányának meghatározása, mint a fej térbeli helyzetének ismerete. Ennek az a magyarázata, hogy a fejünk és a szemünk egymástól függetlenül mozoghat, így a fej helyzetéből csak akkor következtethetünk a figyelem középpontjára, ha a felhasználó szeme nem látható a webkamera képeken. Tehát a figyelem középpontjának meghatározásához szükségünk van egy algoritmusra, mellyel számítógépesen követni tudjuk az emberi tekintetet.

Az elmúlt évek alatt folyamatosan fejlődött a szem és tekintet követésének technológiája. Először csupán arra használták, hogy megtudhassuk, hogy milyen gyorsan és hová néz az ember a képernyőn. A legújabb fejlesztések és törekvések –ahogy a mi fejlesztéseink is – inkább a szemmel történő irányításra koncentrálnak.

Mindazonáltal megjegyezzük, hogy a tekintetfigyelés technológiája nem új, hiszen évek óta foglalkoznak ezzel a fejlesztők. Kezdetben hatalmas, kamerákkal felszerelt sisakokat kellett hordaniuk a vállalkozó szellemű tesztelőknek, azonban a technológia az idők folyamán letisztult. A 90-es évektől jelentek meg azok az eszközök, melyek teljesen lebénult, csupán a szemüket mozgatni képes embereknek segítettek a külvilággal kommunikálni, használva a szemmozgást figyelő szenzorokat és a képernyőn megjelenített virtuális billentyűzetet. A legújabb törekvések azonban ezen is túlmutatnak, a hétköznapi használatba szeretnék bevinni a technológiát, magyarán a szemkövetést a lehető legegyszerűbb és a legszélesebb körben elérhető eszközökkel szeretnék megvalósítani (pl. személyi számítógép és webkamera).

A szemkövetés megvalósításához, első lépésben a szivárványhártya középpontját próbáljuk megtalálni. Erre vonatkozólag több eljárást is kipróbáltunk. Próbálkoztunk pl. Hough transzformáció segítségével köröket, valamint körszeleteket detektálni a fejpozíció kiszámításánál használt szemdetektorok eredményein, de csak akkor jártunk sikerrel, amikor a webkamera előtt ülő felhasználó meredten nyitott szemekkel bámult a kamerába. Ezután próbálkoztunk a pupillának, mint relatíve sötét foltnak és a szemfehérjének, mint relatíve világos foltnak a detektálásával, de ezzel a módszerrel sem jártunk sikerrel, mivel a szemöldökök, a szempillák, az erős smink és a világos bőr bezavart a számításokba. Ezt követően megpróbáltunk egy infra kamerával és infra LED-eknek egy csoportjával vörös szem effektust létrehozni a kameraképeken, de egy speciális szinkronizációs hardver hiányában ezt a módszert sem tudtuk a gyakorlatban megfelelő hatékonysággal alkalmazni.

Vizsgálataink során rájöttünk, hogy az előbbi módszerek főként a szivárványhártya alacsony felbontásának köszönhetően vallottak kudarcot a gyakorlatban. Így egy olyan eljárást kifejlesztésére fektettük a hangsúlyt, mely képes lehet alacsony felbontású webkamera képeken is szivárványhártyát detektálni. A detektálás során a pupilla, szivárványhártya, szemfehérje és szemhéj részek geometriáját és mintázatát szerettük volna felhasználni, így készítettünk egy Viola-Jones detektort az előbb felsorolt objektumokra vonatkozólag. Az így kapott detektorok a bal-, és jobbszem detektorok eredményeire futtatva, már kellően nagy pontossággal (lásd 5. ábra) tudjuk lokalizálni a szivárványhártyát, valamint annak középpontját, a pupillát.

# Összefoglalás

A cikkben röviden összefoglaltuk egy, az ember-gép kommunikációban az emberi fél figyelmét elemző rendszernek egy lehetséges megvalósítását. A rendszer ugyan már működőképes, de a részletes tesztelési szakasz még nem érkezett el, így a komolyabb összehasonlítások csak később fognak elérkezni. Azonban előjáróban néhány szót elárulhatunk a várható teljesítményről. A fej térbeli helyzetét kiszámoló modul teljesítménye csak az egyes arci jellemzőkre betanított Viola-Jones detektoroktól függ, melyek hatásfokát már leírtuk a 2.2-es fejezetben. Az arci jellemzők helyes pozíciójának ismeretében a POSIT eljárás minden esetben jó eredményt fog szolgáltatni, ha a felhasználó 20-100cm-re ül a webkamera előtt, ennél távolabb azonban már a detektorok sem adnak vissza pozitív találatot. A szivárványhártya detektorokra vonatkozólag még nem tudunk tesztadatokat szolgáltatni, de már most tisztán látszik, hogy nagyságrendekkel megbízhatóbb a működése, mint az általunk kipróbált összes eddigi eljárásnak.

# Köszönetnyilvánítás

Irodalom

|  |  |
| --- | --- |
| [1] | Intel® IPP. (2010) Open Source Computer Vision Library. [Online]. <http://sourceforge.net/projects/opencvlibrary/> |
| [2] | R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," *IEEE ICIP*, pp. 900-903, 2002. |
| [3] | K.-K. Sung and T. Poggio, "Example-based learning for view-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 39-51, 1998. |
| [4] | H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 22-38, 1998. |
| [5] | T. Kanade and H. Schneiderman, "A statistical method for 3d object detection applied to faces and cars," International Conference on Computer Vision, 2000. |
| [6] | D. Roth, M. Yang, and N. Ahuja, "A snowbased face detector," *Neural Information Processing*, vol. 12, 2000. |
| [7] | Y. Amit, D. Geman, and K. Wilder, "Joint induction of shape features and tree classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, pp. 1300-1305, 1997. |
| [8] | C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," *Sixth International Conference on Computer Vision (ICCV'98)*, p. 555, 1998. |
| [9] | Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *Computational Learning Theory: Eurocolt '95*, vol. 37, pp. 23-37, 1995. |
| [10] | K. Tieu and P. Viola, "Boosting Image Retrieval," *International Journal of Computer Vision*, vol. 56, pp. 17-36, 2004. |
| [11] | R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *The Annals of Statistics*, vol. 26, no. 5, pp. 1651-1686, 1998. |
| [12] | E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997. |
| [13] | J. K. Tsotsos, et al., "Modeling visual attention via selective tuning," vol. 78, no. 1-2, pp. 507-545, 1995. |
| [14] | L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-based Visual Attention for Rapid Scene Analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254-1259, 1998. |
| [15] | D. C. E and M. C, "Registration of translated and rotated images using finite Fourier transforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, 1987. |
| [16] | G. Aglika, K. Chandrika, and S. C. Cheung, "BlockMatching for Object Tracking," Lawrence Livermore Nation Laboratory, Technical Report, 2003. |
| [17] | L. D. Bruce and K. Takeo, "An iterative image registration technique with an application to stereo vision," *International Joint Conference on Artificial Intelligence*, p. 674–679, 1981. |
| [18] | T. Carlo and K. Takeo, "Detection and tracking of point features," Carnegie Mellon University, Pittsburgh, Technical Report, 1991. |
| [19] | J. Shi and T. Carlo, "Good Features to Track," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994. |
| [20] | L. David, "Perceptual Organization and Visual Recognition," *Kluwer Academic Publishers*, 1985. |
| [21] | Y. John, "A General Photogrammetric Method for Determining Object Position and Orientation," *IEEE Trans. on Robotics and Automation*, vol. 5, pp. 129-142, 1989. |
| [22] | T. Roger, "A Versatile Camera Calibration Technique for High Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE J. Robotics and Automation*, vol. 3, pp. 323-344, 1987. |
| [23] | D. F. DeMenthon and L. S. Davis, "Model-based object pose in 25 lines of code," *Lecture Notes in Computer Science*, vol. 588, no. 335-343, 1992. |
| [24] | J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, pp. 81-106, 1986. |
| [25] | Y. Judy and P. C. Parag, "Human-computer interaction issues for mobile computing in a variable work context," *HCI Issues in Mobile Computing*, vol. 60, pp. 771-797, 2004. |