Debreceni Egyetem
Informatikai Kar

Multi-modális ember-gép kommunikáció

Témavezető:
Dr. Fazekas Attila
egyetemi docens

Készítette:
Kruppa Ádám András
PTM egyetemi hallgató

Debrecen
2009

Tartalomjegyzék

1. Bevezetés

Az utóbbi években a számítógépek teljesen integrálódtak a mindennapi életünkbe. Naponta új alkalmazás és hardver jelenik meg. A számítógéppel való kommunikáció eszközei azonban továbbra is a billentyűzetre, egérre, fényceruzára, trackball-ra stb. korlátozódnak. Mindezek az interfészek megkívánják azt, hogy a felhasználó valamilyen hardver eszközt tartson a kezében ami kényelmetlen lehet. Ezek az eszközök mára már megszokottak, de használatuk nem felel meg az emberek közötti természetes érintkezés céljaira.

A mindennapi életben az emberek gyakran találkoznak információs rendszerekkel. Használatuktól sokan idegenkednek, mivel meg kell tanulniuk az adott információs rendszer működtetését meghatározó parancsokat. Nagyon jó volna, ha a jövőbeli információs rendszerek, számítógépes alkalmazások vezérlésének módja hasonlítana az emberek közötti kommunikációhoz. Azaz, a legtermészetesebb módon beszéddel, testi gesztusokkal, arckifejezéssel, kézjelekkel tudnánk kapcsolatot teremteni az információs rendszerek eszközeivel és működtetni azokat.

A multi-modális felhasználói interfész egy olyan rendszer, amely képes több kommunikációs csatornán is fogadni a bemeneti adatokat és azokat értelmes módon kombinálni. Gondolhatunk itt a mindenki által jól ismert kő-papír-olló játékra is. Hiszen ebben a játékban a számítógépnek "látnia" kell a játékos által mutatott kézjelet, valamint a gép meg is "hallhatja" a játékos által kimondott szót. A gép a két kommunikációs csatornán kapott információt együttesen feldolgozhatja, majd reagálhat arra. Például az ún. beszélő fej a beszédszintetizátor segítségével kimondhatja a játék során a gép által választott tárgy nevét, majd később a játék eredményének tudatában a beszélő fej kifejezheti tetszését vagy nem tetszését attól függően, hogy ki nyert. Összefoglalóan azt mondhatjuk, hogy a multi-modalitás azt jelenti, hogy a rendszer a felhasználóval több csatornán is kommunikál abból a célból, hogy automatikusan hasznos információt nyerjen ki, illetve továbbítson a felhasználó felé. A multi-modális rendszerek legfontosabb tulajdonsága az, hogy a modalitások megválasztásával olyan ember-gép párbeszéd alakulhat ki, amely nagyon hasonlít az ember-ember kommunikációjához.

A Debreceni Egyetem Komputergrafika és Képfeldolgozás Tanszéknek egy korábbi sikeres projektje volt az emberszerű sakkozó gép létrehozása. Az akkor kialakított komponensek segítségével később egy másik táblát használó játékot, a multi-modális dáma játékot is megvalósították. Felmerült a gondolat, hogy a kő-papír-olló játék is elkészíthető multi-modális módon a rendelkezésre álló eszközökkel, feltéve hogy a számítógép "szeme" a webkamera által továbbított képen sikerül-e a játékos által mutatott kézjelet azonosítani. Az elképzelés az, hogy a játékos a PC vagy laptop előtt ül és a webkamera szemből mutatja a játékost, aki egy adott jelre a kezét a teste elé téve mutatja az általa választott jelet, és egyidejűleg ki is mondja a jel nevét ugyanúgy, mint a valóságos emberek közötti játék során. A gép is választ egy jelet, amelyet "szóban" és képben is közöl a felhasználóval. Ez után kialakul a játék eredménye, amit a gép megállapít és hasonló módon közöl.

Feladatom annak a képfeldolgozási feladatnak a megoldása volt, hogy egy összetett

1. ábra. A vizsgálandó kép egyszerű környezet esetén. A felhasználó a kézjelet az asztal lapján mutatja a webkamera felülről készíti a képet.

háttér előtt mutatott képen azonosítsam a kéz helyét. Majd pedig a mutatott kézjel pontos alakját meghatározzam. A képet, amelyen a kézjel felismerést végre kell hajtanom, a PC-hez csatlakoztatott webkemara által továbbított videófolyamból kellett kinyernem.

A testi gesztusok – speciálisan a kézjelek – használata számos esetben egy természetes eszköz lehet a számítógépekkel folytatott kommunikációban is. Például lehetővé teheti, hogy egy CAD modell forgatását a kéz forgatásával érjük el. További alkalmazás lehet az is, hogy a számítógépes játékokat kézmozdulatokkal irányítsuk. Berendezések irányítását is végezhetjük kézjelekkel. Juan P. Wachs és szerzőtársai cikkükben ismertettek egy olyan eszközt, amely lehetővé teszi, hogy az idegsebészek a műtét alatt a páciensről készült felvételeket kézjelekkel manipulálhassák. A rendszert egy washingtoni kórházban tesztelték. Az érdeklődő olvasó további részleteket találhat [1]-ben.

A kézjeleket két kategóriába sorolhatjuk, ezek statikusak vagy dinamikusak lehetnek. Statikus kézjel a kéznek egy adott konfigurációja (megformált alakja), amely egyetlen képpel ábrázolható. A dinamikus kézjel az egy mozgó kézjel, amely a képek sorozatával ábrázolható. Esetünkben statikus kézjelről van szó, hiszen feltételezzük, hogy a játékos csak a felszólítás után, rövid ideig mutatja a kézjelet. Ez azt jelenti, hogy nem kell követnünk a kéz mozgását, *csupán* a webkamera által közvetített videófolyamból kell egy adott képet analizálni, a kéz helyzetét megállapítani. A szegmentálás (a kéz helyzetének megállapítása) a legnehezebb folyamat a kézjel felismerés folyamatában, mivel a kéz bonyolult háttér előtt helyezkedhet el. Éppen ezért a kézjel felismerési vizsgálatokban még manapság is sok egyszerűsítést tesznek, például feltételezték, hogy a kézjelet egy fekete háttér előtt mutatják [3, 2]. Feladatom megoldását először én is ilyen egyszerű háttér feltevés mellett kezdtem el.

Vizsgálataimat végül két különböző környezet feltevés mellett végeztem el. Az első esetben a kézjelet a felhasználó egy asztal fölött mutatja és a webkamera felülről készíti el

2. ábra. A vizsgálandó kép komplex, bonyolult környezet esetén. A felhasználó a kézjelet a webkamerával szemben mutatja.

a fényképet. Ekkor a szegmentálás elég egyszerű, ha az asztal lapja egyszínű. A második esetben a webkamera szemből mutatja a felhasználót, ekkor természetesen a szoba háttere bekerül a képbe. Tovább bonyolítja a helyzetet, hogy nemcsak egyetlen bőrszín terület jelenik meg, hanem a felhasználó arca is, valamint a háttérben is elhelyezkedhet bármilyen bőrszínhez hasonló tárgy. Vizsgálataimat az első egyszerűbb esetben sem homogén háttér mellett csináltam. Az asztalon lehetnek különböző tárgyak, például könyvek, ceruzák stb. Egy tipikus helyzetet mutat az 1. ábra. A második esetben amikor a webkamera a felhasználóval szemben helyezkedik el, a környezet sokkal bonyolultabb, összetettebb. A kéz helyének kijelölése ilyenkor nehéz, hiszen nagyobb térrész látszik, ahol elhelyezkedhetnek bőrszínszerű tárgyak, valamint a játékos arca is. A 2. ábra egy tipikus helyzetet mutat arról, amikor a kézjel összetett környezetben szerepel.

A kézjel felismeréssel kapcsolatos megoldások három kategóriába sorolhatók. Az első kategóriába tartoznak azok, amelyek speciális eszközt, például kesztyűt adnak a felhasználóra, és mechanikai-optikai érzékelők érzékelik a kéz mozgását, a kéz helyzetét. A második kategóriába (magas szintű megközelítés) tartoznak azok a módszerek, amelyekben a kéz egy három dimenziós modelljét felhasználva történik meg a képen látható kézjel összevetése a modellel. Így kerülnek meghatározásra azok a paraméterek, amelyből megállapítható a kézjel típusa. Ez a megközelítés nagyon stabil és jó eredményeket ad, de a gyorsaság rovására.

A harmadik megközelítést nevezhetjük alacsony szintű közelítésnek, mivel csupán a pixel intenzitások adataival dolgozik. Ezek a módszerek egyszerűek és gyorsak. Ez utóbbi szempont (gyorsaság) számomra is fontos volt, mivel egy játék számítógépes megvalósításásról van szó a feladatomban, ami egy interaktív alkalmazás, és ekkor a válaszidőnek gyorsnak kell lenni, hiszen a felhasználó nem vehet észre késleltetést az általa mutatott kézjel és a számítógép válasza között.

A kézjel felismerése két lépésben történik. Először ismert kézjeleket mutatunk a gépnek. Ezekre a képekre a későbbiekben tanítóképként fogok hivatkozni. A tanítóképek számát célszerű minél nagyobbnak választani. A képeket a gép bizonyos szempontok szerint elemzi és az eredményekből készít egy adatbázist. A kézjel felismerés második lépésében mutatjuk a gépnek a felismerendő képet. Ezt a képet ugyanolyan módon mint a tanítóképeket a gép elemzi. Az eredményt valamilyen módszerrel összehasonlítjuk az adatbázisban tárolt értékekkel, és ez alapján a gép dönt a kezjel típusárol.

Ezen általános séma első lépéseként választanunk kell olyan mennyiségeket, ami a kézjelet jellemzi. Ha több ilyen mennyiség is van akkor azokat gondolhatjuk egy vektor komponenseinek. Igy minden képhez elkészítünk egy sajátság-vektort. Dolgozatomban két különböző sajátság-vektort használtam. Az első esetben lokális orientációs hisztogrammal reprezentáltam a képet. A második esetben a kézjel momentumai alkották a sajátság-vektort.

A tulajdonképpeni döntési fázisban egy igen egyszerű módszer is jól müködött. A tanítóképhez és a felismerendő képhez tartozó sajátság-vektorok különbségének a hossza jellemzi a két kép hasonlóságát. Ahol a hasonlóság ezen mértéke a legkisebb azzal a kézjellel azonosítjuk a képet.

Dolgozatom hat nagy részből tevődik össze. A *Bevezetés* után a *A kő-papír-olló játék* fejezetben ismertetem a játék szabályait, majd a *Szegmentálás* című részben bemutatom, hogy háttérlevonás és bőrszínfelismerés alkalmazásával sikerül bonyolult hátterű kép esetén is kijelölni egy téglalap alakú tartományt, ahol a kézjel elhelyezkedik. A *Kézjel felismerés* című részben részletesen ismertetem az orientációs hisztogram és a momentumok módszerét, valamint a felismerési algoritmust. A *Kísérleti eredmények* című fejezetben konkrét példák alapján vizsgálom a módszerek jellemzőit, hatékonyságukat. A dolgozatot az *Összefoglalás* zárja.

2. A kő-papír-olló játék

A kő-papír-olló játékot két ember játszhatja a kezével. Ez a játék a világ minden részén ismert és közkedvelt. Japánban jankennek nevezik, a többi országban ugyanilyen néven ismerik. A játék menete a következő. A játékosok háromig számolnak és minden számolásnál felemelik ökölbe szorított kezüket. A harmadik számolás után a játékosok kezükkel mutatják a három jel egyikét és megmutatják az ellenfelüknek. A követ zárt ökölle, a papírt nyitott tenyérrel és az ollót kinyújtott, szétnyitott mutató és középső ujjal mutatják.

Az a cél, hogy olyat mutassunk, ami legyőzi az ellenfél által mutatott jelet. A döntés szabályai a következők. A kő kicsorbítja az ollót, ekkor a kő győz. Az olló elvágja a papírt és ezért az olló győz. A papír becsomagolja a követ és így a papír győz. Ha mindketten ugyanazt mutatják, a játék döntetlen és újat játszanak.

A játékelmélet segítségével bizonyítható, hogy az egyetlen ésszerű játékmód, ha minden lépésben véletlenszerűen mutatjuk a három jel valamelyikét. Ezt a stratégiát játszva

3. ábra. Bőrszín felismerés alkalmazása az 1. ábrán látható képre az YC_rC_b színtéren, a (3.8) képlet alapján.

elérhetjük, hogy ellenfelünk lépéseitől függetlenül, hosszú ideig játszva, a lehető legkevesebbszer veszítsünk. Ha ellenfelünk is így játszik, akkor minden lépésben ugyanakkora az esélye, hogy veszítsünk, nyerjünk, vagy döntetlen legyen a játék. Sok játék menet után mindkét játékos nagyjából ugyanannyiszor fog nyerni, mint veszteni.

3. Szegmentálás

A látás alapú kézjel felismerés két fő lépésből áll. Az első lépés a szegmentálás, vagyis kijelöljük a képen azt a területet, ahol a kéz elhelyezkedik. A második lépés a tulajdonképpeni kézjel felismerés, amikor megállapítjuk, hogy milyen kézjelet mutatott a felhasználó. A jellemzőkön alapuló kéz- és arcdetektálási módszerek közül a bőrszín, mint felismerési kulcs, nagyon népszerűvé vált. A színt használó módszerek gyors végrehajtást tesznek lehetővé és nagyon stabilak az arc és kéz geometriai változásaival szemben. Mivel a bőrszín felismerés különböző színtereken történhet, most röviden ismertetem a leggyakrabban használt színtereket és azok tulajdonságait.

3.1. Színterek

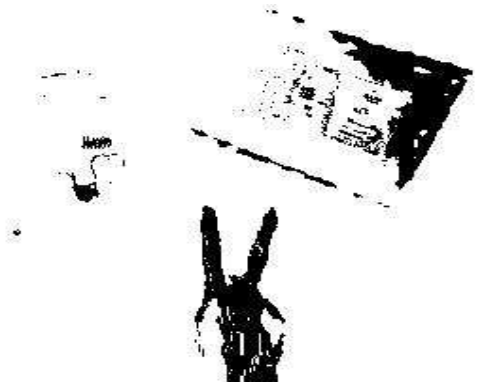
Amit fényként vagy különböző színként érzékelünk, az tulajdonképpen elektromágneses sugárzás egy szűk frekvencia tartományban. Mivel a fény elektromágneses hullám, ezért a különböző színeket a fény frekvenciájával jellemezhetjük. Egy fényforrás, mint például a Nap, vagy egy villanyégő, valamilyen frekvencia tartományban sugároz. Amikor ez a fehér fény ráesik egy tárgyra, annak egy része elnyelődik, másik része visszaverődik. A visszavert fény frekvenciáinak kombinációja határozza meg milyen színűnek látjuk a tárgyat. Ha

az alacsony frekvenciák dominálnak, akkor a tárgyat vöröses színűnek látjuk. Ebben az esetben azt mondhatjuk, hogy az érzékelt fénynek van egy domináns frekvenciája, ezt a színárnyalatot gyakran hue-nak is szokás nevezni. A domináns frekvencia mellett más tulajdonságok is szükségesek, hogy leírjuk a fény jellemzőit. A szemünk még két másik tulajdonságot is megkülönböztet. Ezek egyike a fényerősség, ami a szemünkben elnyelt fény energiájával, intenzitásával kapcsolatos. A harmadik fontos jellemzője a fénynek a színtelítettség, más néven a szaturáció. A domináns frekvenciát és a szaturációt együttesen kromatikusságnak szokás nevezni.

Az RGB színtér a monitorok fejlesztése során alakult ki, amikor kényelmes volt a színt három különböző színű sugárnyaláb (piros, zöld, kék) kombinációjaként előállítani. Az RGB színtér a leggyakrabban használt additív színtér digitális adatok processzálására és tárolására. Azonban az egyes színcsatornák között nagy korreláció van. A kromatikus és luminancia értékek keverednek. Ezért az RGB színteret nem nagyon alkalmazzák bőrszín analízisre és színen alapuló felismerési algoritmusokban. Ennek ellenére néhány munkában ezt is felhasználták [4].

A normalizált RGB reprezentáció a következő formulákkal adható meg

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}. \tag{3.1}$$



4. ábra. Bőrszín felismerés alkalmazása az 1. ábrán látható képre a TSL színtéren, a (3.12) képlet alapján.

Mivel a normalizált komponensek összegére teljesül az, hogy $r + g + b = 1$, ezért a harmadik komponens nem hordoz semmi jelentős információt és b-t elhagyhatjuk, ezáltal csökken a komponens csatornák száma. A normalizált komponenseket (r-t és g-t) gyakran nevezik tiszta színeknek.

Mivel ez a transzformáció nagyon egyszerű, ezért szívesen alkalmazzák különböző vizsgálatokra.

A HSI színtér közelebb áll az emberi érzékeléshez. A hue adja meg a domináns színt, a szaturáció méri a szín telítettségét, az intenzitás pedig a szín luminanciájával kapcsolatos. Az RGB színtérből a HSI színtérbe való transzformáció a következő képletekkel adható meg:

$$H = \begin{cases} \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-B)(G-B)}} \text{ , ha } B > G, \\ 2 - \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-G)(G-B)}} \text{ , egyébként.} \end{cases} \tag{3.2}$$

$$S = 1 - 3\frac{\min(R,G,B)}{R+G+B}, \tag{3.3}$$

$$I = \frac{1}{3}(R+G+B). \tag{3.4}$$

5. ábra. A bőrszín felismerés alkalmazása a 2. ábrán látható képre, YC_rC_b színtéren a (3.8). képlet alapján.

Az YC_rC_b színteret gyakran használják képtömörítéseknél. A színt itt is három szám jellemzi. A luma (Y), az RGB komponensek súlyozott átlaga. A további két számot, C_r-t és C_b-t pedig úgy kapjuk, hogy a vörös és kék komponensekből kivonjuk a lumát. Képletekben ez a következőket jelenti:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B, \\ C_r &= R - Y, \\ C_b &= B - Y. \end{aligned} \tag{3.5}$$

Ez a transzformáció is nagyon egyszerű, és az az előnye, hogy explicit módon szeparálódnak a luminancia és a kromatikus komponensek. Ezért ezt a színteret igen gyakran használják bőrszín felismerő modellekben.

A TSL színtér a normalizált RGB komponensek transzformációjával adható meg. A pontos formula a következő:

$$\begin{aligned} T &= \arctan(\frac{r'}{g'})/2\pi + 1/4 \quad g' > 0, \\ S &= \sqrt{\frac{9}{5}(r'^2 + g'^2)}, \\ L &= 0.299R + 0.587G + 0.114B, \end{aligned} \tag{3.6}$$

ahol $r' = r - \frac{1}{3}$ és $g' = g - \frac{1}{3}$.

3.2. Bőrszín felismerés

A szegmentálási folyamat első lépése a bőrszín felismerése. A szín, mint egy kulcsjellemző azért fontos, mert a kísérletek azt bizonyítják, hogy az emberi bőrnek jellegzetes színe van, amely könnyen felismerhető. Ezért az az ötlet, hogy alkalmazzuk a bőrszínt kézjel felismerésre nagyon természetes. A bőrszín detektálás végső célja az, hogy megadjon egy olyan döntési szabályt, amely különbséget tud tenni bőrszínű és nem bőrszínű pixelek között.

Az egyik lehetséges megoldás a következő. Egy választott színtérben kijelölünk egy tartományt, és azokat a pixeleket, amelyek ide esnek bőrszínűnek tekintjük. A paramétereket nem alkalmazó bőrmodellek meghatározzák a bőrszín eloszlását egy tanulási folyamat során. Ismert bőrszíneket mutató képeket elemeznek és elkészítik a bőrszín eloszlását meghatározó függvényt.

Amikor egy rendszert készítünk a bőrszín felismerésre, három fő problémával kell szembenéznünk. Az első kérdés az, hogy milyen színteret válasszunk. A második kérdés amire válaszolnunk kell, hogy a bőrszín eloszlást pontosan hogy modellezzük. Harmadszor pedig az, hogy a színszegmentálás eredményét hogyan továbbítsuk a kézjel felismeréshez.

A bőrszín felismerési algoritmusok két különböző kategóriába sorolhatók. Az első a pixel alapú bőrdetektálás, ekkor minden pixelről eldöntjük, hogy az bőrszínű-e vagy sem, függetlenül a környezetétől. Ezzel szemben a terület alapú módszerek a bőrpixelek térbeli kiterjedését is figyelik. Dolgozatomban a pixel alapú megközelítést választottam.

A bőrszín felismerő módszerek közül mi azt választjuk, amikor explicit módon kijelölünk egy területet a színtérben. Egy nyilvánvaló előnye ennek a módszernek az, hogy a bőrszín felismerési szabály egyszerű, és nagyon gyors elemzést tesz lehetővé. A különböző színterekben különböző tartományokat jelölhetünk ki bőrszín területnek.

Az RGB színtérben a következő formulával lehet kijelölni bőrszín területet [4]:

$$\begin{aligned} &R > 95 \text{ és } G > 40 \text{ és } B > 20 \text{ és} \\ &\max(R,G,B) - \min(R,G,B) > 15 \text{ és} \\ &|R - G| > 15 \text{ és } R > G \text{ és } R > B. \end{aligned} \tag{3.7}$$

Az YC_bC_r színtérben egyszerűen egy téglalap ez a terület [5]:

$$133 < C_r < 173 \text{ és } 77 < C_b < 127. \tag{3.8}$$

Egy másik vizsgálat szerint [7] ebben a színtérben öt egyenes által határolt tartomány jelöli ki azt a területet, ahová a bőrszín esik

$$\begin{aligned} C_r &< 1.5862C_b + 20, \\ C_r &> 0.34648C_b + 76.2069, \\ C_r &> -4.5652C_b + 234.5652, \\ C_r &< -1.15C_b + 301.75, \\ C_r &< 2.2857C_b + 432.85. \end{aligned} \tag{3.9}$$

A normalizált RGB térben pedig két parabola közötti terület adja meg a bőrszín tartományt. A következő formulákkal jelölhető ki a bőrszín terület:

$$g < -1.3767r^2 + 1.0743r + 0.1452 \quad \text{és} \quad g > -0.776r^2 + 1.0743r + 0.1452. \tag{3.10}$$

A HSI színtérben egyszerűen két szakasz jelöli ki a bőrszín tartományt [7]:

$$0 < H < 25 \quad \text{vagy} \quad 230 < H < 360. \tag{3.11}$$

A kísérleti vizsgálatok szerint a TSL színtérben a bőrszín tartományt a következő egyenlőtlenségek jelölik ki

$$0,4 < T < 0.6 \quad \text{és} \quad 0.038 < S < 0,25 \quad \text{és } L \geq 80. \tag{3.12}$$

A bőrszín felismerést a fejezet elején ismertetett módszerek mindegyikével megvizsgáltam. Az eredmények közül néhányat bemutatok. Az YC_bC_r színteret alkalmazva a 3. ábrát kaptam. A TSL színteret alkalmazva az eredményt a 4. ábra mutatja. Megfigyelésem az volt, hogy a legjobb bőrszín felismerést az YC_rC_b színtér adta, ahol is a bőrszín tartományt a (3.8) képlet definiálja. Terrillion és munkatársai [8], kilenc különböző színtérben vizsgálták a bőrmodellezést és azt találták, hogy a TSL tér messze a legjobb az összes színtér között. Más vizsgálatok szerint viszont az YC_rC_b a legmegfelelőbb [5]. Úgy találtam, hogy az irodalomban nincs egységes álláspont arról, hogy melyik színtér a legmegfelelőbb a bőrszín felismerésre. Valószínűleg ez nagyban függ a konkrét alkalmazástól.

A 1. ábrán látható kép egy szobában készült, a megvilágítás természetes szórt fény volt. Ismeretes hogy a bőrszín tartomány függ a megvilágítástól, és különböző fényforrások mellett a bőrszín tartomány változik [9]. Valószínű, hogy természetes nappali külső felvételnél mindegyik színtér nagyon jó eredményt ad. A további vizsgálatokban az YC_rC_b színteret használtam, és a bőrszín tartományt a (3.8) egyenlőtlenség jelöli ki.

6. ábra. A háttérlevonás alkalmazása az 1. ábrán látható képre, a (3.23) módszerrel.

7. ábra. A háttérlevonás alkalmazása a 2. ábrán mutatott képre, a (3.23) módszerrel.

8. ábra. A háttérlevonás alkalmazása a 2. ábrán mutatott képre, a (3.15-3.17) képletekkel adott módszerrel.

3.3. Háttérlevonás

Láthatjuk azt, hogy a bőrszín felismerésen alapuló szegmentálással még nem kapjuk meg pontosan a kéz helyzetét. Különösen igaz ez akkor, ha a kamera szemből mutatja a felhasználót, és olyan szerencsétlen körülmény áll fenn, hogy nagyon sok bőrszínhez hasonló tárgy veszi őt körül. A 2. ábra a webkamera által készített képet mutatja a felhasználóról az 5. ábrán pedig a bőrszín felismerésen alapuló szegmentálást láthatjuk. Ebben az esetben nem jutottunk előre, nem tudjuk egyértelműen kijelölni a kéz helyzetét. A most felmerült problémán segíthet a háttérlevonás alkalmazása.

9. ábra. A háttérlevonás után a bőrszín felismerés alkalmazása a 6. ábrán mutatott képre.

A háttérlevonás fontos lépése nagyon sok számítógépes vizuális alkalmazásnak. Gyakran használják videókamerát alkalmazó problémáknál. Így például közlekedés ellenőrzés,

biztonsági kamerák, kézjel felismerés. Az általános cél a következő. Adott egy rögzített kamera által készített képsorozat, és szeretnénk kiválasztani ezekből az általunk szükségesnek tartott tárgyakat (előteret). Feltételezzük, hogy ismerjük a statikus háttérképet és az aktuális képet. Ekkor egy természetes gondolat, hogy képezzük a háttér és az aktuális kép különbségét és választunk valamilyen küszöbértéket. Ha az eltérés nagyobb mint a küszöbérték, akkor megkapjuk az előteret. Az első kérdés ami felmerül az, hogy hogyan határozzuk meg a jelenet statikus hátterét. A háttérkép ugyanis nem rögzített, megváltozhatnak a fényviszonyok, a kamera remeghet, stb.

A két legalapvetőbb módszer igen egyszerűen megfogalmazható. A frame különbség módszere azt jelenti, hogy a háttérkép pontosan az aktuális képet megelőző kép volt. Természetesen itt is alkalmazzuk a küszöbölési eljárást. Ennek a módszernek egy változata, hogy a háttér az aktuális képet megelőző n darab kép átlagképe.

Bonyolultabb háttér modellek esetén, az aktuális képnél korábban készült képek alapján meghatározzuk az intenzitás (vagy színcsatornák) átlagértékét és szórását. Minden egyes pixel színcsatornáinak eloszlását Gauss eloszlással modellezzük. Természetesen a háttér időben változik, ezért az eloszlás paramétereit frissíteni kell. A paraméterek ezen karbantartására jól kidolgozott módszerek vannak.

A kő-papír-olló számítógépes játék nem ilyen jellegű háttér levonást igényel. Ha a számítógép webkamerája a felhasználóval szemben van, akkor természetesen van egy nagyon stabil háttér, például a szoba falai, bútorai. De a kép nagy részét betöltő emberi fej és test sokat mozog. Éppen ezért a háttér modell paramétereinek a megváltoztatása, karbantartása nem lehetséges. További indok erre még, hogy a játék gyors ritmusú, éppen ezért egy speciális háttérlevonási módszert alkalmazok. Közvetlenül az aktuális kézjel bemutatása előtt készített öt darab felvétel alapján készítem el a háttérmodellt. Az öt felvételt és a háttér modell kialakítását, minden egyes kézjel bemutatása előtt megismétlem.

10. ábra. A háttérlevonás után a bőrszín felismerés alkalmazása a 7. ábrán mutatott képre.

11. ábra. Az utófeldolgozás első lépésével kapott képek. A kiinduló kép a 3. ábrán látható. Az eredeti pixelrácsot 8x8-as, 4x4-es és 2x2-es egységekre bontottam. Az ezeknek megfelelő eredmények az ábra bal oldalán, középen és a jobb oldalon láthatók.

Két fajta háttér levonási módszert alkalmaztam. Az egyik esetben, minden egyes képpont esetén meghatározom az összes színcsatorna átlagértékét (RGB színtér)

$$\bar{R}(i,j) = \frac{1}{N_h}\sum_{k=1}^{N_h} R^{(k)}(i,j), \tag{3.13}$$

ahol N_h a háttérképek száma és $R^{(k)}(i,j)$ a k. háttérkép piros komponensének értéke az (i,j) koordinátájú pixel helyen. A szórást a következő képlettel számoljuk:

$$\sigma_R(i,j) = \sqrt{\frac{1}{N_h}\sum_{k=1}^{N_h}(R(i,j) - \bar{R}(i,j))^2}. \tag{3.14}$$

Természetesen a többi színcsatornára hasonlóan értelmezzük az átlagértéket($\bar{G}(i,j)$, $\bar{B}(i,j)$) és szórást ($\sigma_G(i,j)$, $\sigma_B(i,j)$). Az aktuális kép adott (i,j) pixelénél legyenek az RGB komponensek $R(i,j)$, $G(i,j)$ és $B(i,j)$. Egy (i,j) képpontot akkor tekintünk háttér pixelnek, ha teljesülnek a következő egyenlőtlenségek:

$$|R(i,j) - \bar{R}(i,j)| < 2\sigma_R(i,j), \tag{3.15}$$

$$|G(i,j) - \bar{G}(i,j)| < 2\sigma_G(i,j), \tag{3.16}$$

$$|B(i,j) - \bar{B}(i,j)| < 2\sigma_B(i,j). \tag{3.17}$$

A második módszer [10] esetén szintén a korábban elkészített háttérképek alapján dolgozunk. Minden egyes pixel esetén áttérünk szürkeskálás képre a következő transzformációval:

$$I(i,j) = 0,3R(i,j) + 0,6G(i,j) + 0,1B(i,j). \tag{3.18}$$

Minden képpont esetén, a háttérképek alapján meghatározom az intenzitás minimális és maximális értékét

$$I_{min}(i,j) = \min\{I^{(1)}(i,j), I^{(2)}(i,j), \dots I^{(N_h)}(i,j)\}, \tag{3.19}$$

$$I_{max}(i,j) = \max\{I^{(1)}(i,j), I^{(2)}(i,j), \dots I^{(N_h)}(i,j)\}, \tag{3.20}$$

ahol $I^{(k)}(i,j)$ a k. háttérkép intenzitás értéke az (i,j) pixel helyen. A háttér levonás második módszere esetén még egy újabb mennyiség is szükséges a háttérmodell megadásához. Egy adott pixel helyén meghatározzuk az egymást követő képkockák (frame-k) közötti intenzitás különbségek maximumát:

$$\begin{aligned} d(i,j) &= \max\{|I^{(2)}(i,j) - I^{(1)}(i,j)|,\ |I^{(3)}(i,j) - I^{(2)}(i,j)|, \\ &\dots\ ,\ |I^{(N_h)}(i,j) - I^{(N_h-1)}(i,j)|\}. \end{aligned} \tag{3.21}$$

Jelöljük d-vel az egész képre vonatkozó maximális keretek közötti intenzitás különbségek átlagát

$$d = \frac{1}{N_h} \sum_i \sum_j d(i,j), \tag{3.22}$$

ahol az i-re és j-re való összegzés kiterjed a kép egész területére. Az aktuális kép (i,j) képpontja akkor tekinthető háttérpixelnek, ha teljesíti a következő egyenlőtlenséget:

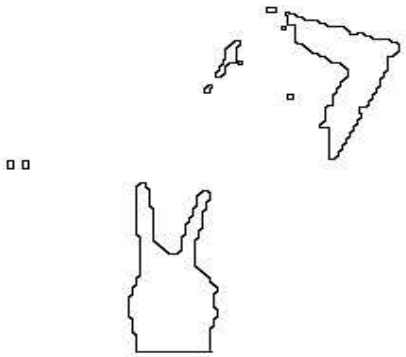
$$I_{min}(i,j) - kd < I(i,j) < I_{max}(i,j) + kd. \tag{3.23}$$

A fenti formula tartalmazza a k paramétert. Ezt kísérletileg kell meghatározni. A [10] cikkben azt javasolták, hogy legyen $k = 2$. Ezt az értéket én is megfelelőnek találtam.

Esetünkben a webkamera elég közelről mutatja a felhasználót, a kézjelek bemutatása gyorsan történik, a felhasználó gyorsan mozgathatja a fejét vagy a testét, így nincs lehetőség arra, hogy a háttérmodell paramétereit időben változtassuk. Éppen ezért az előzőleg ismertetett, egyszerű háttér levonási technikát alkalmaztam. Az aktuális kézjel bemutatása előtt készítünk öt felvételt, és ezen képek alapján határozzuk meg a háttérmodellt. Miután a háttérmodellünk megvan, sor kerülhet az aktuális kézjel bemutatására. Azaz a webkamera által közvetített videó folyamból kiválasztunk egy képet, és ezen végrehajtjuk a háttérlevonást. Az egyszerű és bonyolult környezet esetén az eredeti képek az 1. és a 2. ábrákon láthatóak. A háttér levonás utáni színes képeket a 6. és a 7. ábrák mutatják. Itt a háttér levonásra a (3.23) képlettel adott eljárást használtam. Az összetett környezet esetén a háttérlevonást a (3.15-3.17) képletekkel definiált módon is végrehajtottam, az eredményt a 8. ábra mutatja. Ha összehasonlítjuk a 7. és 8. ábrákon látható képeket, akkor láthatjuk, hogy a háttér levonás sokkal jobban sikerült azzal a módszerrel, amelynek az eredménye a 7. ábrán látható. Ezért a háttér levonást a (3.23) képlettel definiált módon fogom alkalmazni a későbbiekben.

12. ábra. Az utófeldolgozás második lépésével kapott kép. A kiinduló kép a 11. ábra jobb oldalán látható.

13. ábra. Az utófeldolgozás utolsó lépésével kapott kép. A kiinduló kép a 12. ábrán látható.



14. ábra. A bináris éldetektálás alkalmazása a 13. ábrán mutatott képre.

Ha a háttér levonás utáni képekre alkalmazom az általam használt bőrszín felismerési módszert, akkor az így keletkezett képeket a 9. és 10. ábrákon mutatom be. Láthatjuk, hogy az eredeti képen még meglévő hatalmas bőrszínhez hasonló falfelületek már eltűntek a képről, és a felhasználó arcából is csak egy-két kisebb felület látható.

Sajnos a háttér levonásnak nem kívánatos eredménye is van. A kézjelet mutató kéz felületéről is eltűntet néhány, a felismeréshez szükséges területet. Ez világosan látszik, ha összehasonlítjuk a 3. és 9. ábra képeit.

Előfordulhat az is, hogy a háttérképek és az aktuális kép elkészítése közötti időben valamilyen drasztikus változás történik, például a felhasználó túlságosan sokat mozog, megváltoznak a fényviszonyok. Ekkor a háttér levonási módszer nem működik. Mivel ezt ki akarjuk küszöbölni, ezért követjük a [10] cikkben ajánlottakat. Ez a következőt jelenti, ha a háttér levonás után az előtér a kép több mint 25 %-át kitölti, akkor megkérjük a felhasználót, hogy mutassa újból a kézjelet. Ekkor természetesen újból készítünk öt új háttérképet.

A háttérlevonás és bőrszín felismerésen alapuló szegmentálás után nyert bináris képet tovább kell elemezni, mivel a képen több mint egy alakzat lehet. Nekünk pedig pontosan ki kell jelölni azt a területet, ahol a kéz elhelyezkedik. Ehhez további bináris képfeldolgozási eljárásokat kell végrehajtani, olyanokat amelyek előnyösek a további feldolgozás számára. Mielőtt ezeket ismertetném, áttekintem a bináris digitális képeken végezhető műveletek alapjait.

3.4. Műveletek bináris képeken

A digitális kép, ha formálisak akarunk lenni, akkor egy kétváltozós függvénnyel adható meg. Olyan kétváltozós függvénnyel, amelynek értelmezési tartománya és értékkészlete

15. ábra. Az utófeldolgozás első lépésével kapott kép összetett környezet esetén. A kiinduló kép a 10. ábrán látható.

nagyon speciális. Az értelmezési tartomány a következő halmaz:

$$\{(i,j)|\ 1 \leq i \leq m,\ 1 \leq j \leq n\}, \tag{3.24}$$

ahol m és n természetes számok és $m, n > 0$. Az f függvény értékkészlete pedig az alábbi halmaz:

$$\{0, 1, 2, \ldots, W\}, \tag{3.25}$$

ahol W a fehér szín értéke. W értékét az határozza meg, hogy az intenzitás értékeket hány biten ábrázoljuk. A legtöbb esetben $W = 255$. Az $f(i,j)$ értékeket mátrix formába is elrendezhetjük:

$$F = \begin{pmatrix} f(1,1) & f(1,2), & \ldots & f(1,n) \\ f(2,1) & f(2,2), & \ldots & f(2,n) \\ \ldots & \ldots, & \ldots & \ldots \\ f(m,1) & f(m,2), & \ldots & f(m,n) \end{pmatrix}. \tag{3.26}$$

Az F digitális képen az (i,j) egész számpár jellemzi a pozíciót, amelyet képpontnak vagy pixelnek nevezünk. Az mn értéket a kép térbeli felbontásának nevezzük. Szürkeskálás képen f az intenzitást jelenti. Színes digitális kép esetén három, az előzőhöz hasonló típusú függvénnyel adhatjuk meg a képet. Például az RGB színtér esetén $R(i,j)$, $G(i,j)$ és $B(i,j)$ jelöli az (i,j) képponthoz tartozó piros, zöld és kék komponensek értékét.

Ha adott egy f digitális képünk, akkor azon különböző műveleteket értelmezhetünk. A lokális operátorok a következő képlettel adhatók meg:

$$\begin{aligned} c(i,j) = g(&f(i-1,j-1), f(i-1,j), f(i-1,j+1), f(i,j-1), \\ &f(i,j), f(i,j+1), f(i+1,j-1), f(i+1,j), f(i+1,j+1)), \end{aligned} \tag{3.27}$$

ahol g egy kilencváltozós függvény. Ez a művelet az f képet a c képbe viszi át. Azt mondjuk, hogy ennek a műveletnek egy 3x3-as maszkja (sablonja) van, azaz egy pixel értékét saját maga és az őt körülvevő nyolc másik pixel értéke határozza meg.

Bináris képek olyan digitális képek, ahol az értékkészlet két elemű halmaz $\{0, 1\}$. Azt a konvenciót fogom követni, hogy az 1-es érték a feketét a 0 pedig a fehér színt jelöli. Konkrét programozási környezetben egy bináris kép legkönnyebben egy logikai értékeket tartalmazó tömbbel adható meg. A 0 és 1 értékeket logikai változóknak is gondolhatjuk. Ezért a bináris képen végezhető műveletek egy részét megadhatjuk logikai műveletekkel is.

16. ábra. Az utófeldolgozás második lépésével kapott kép összetett környezet esetén. A kiinduló kép a 15. ábrán látható.

A fekete területeket kiterjeszthetjük (expand black areas) a

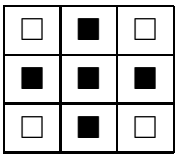
$$c(i,j) = f(i-1,j-1) \oplus f(i-1,j) \oplus f(i-1,j+1) \oplus f(i,j-1) \oplus \\ f(i,j) \oplus f(i,j+1) \oplus f(i+1,j-1) \oplus f(i+1,j) \oplus f(i+1,j+1) \qquad (3.28)$$

módon definiált művelettel. Itt a $\oplus$ jel a logikai- vagy műveletet jelenti. A (3.28) képlet szemléletes jelentése a következő: az (i, j) pixel fekete lesz, ha 3x3-as környezetében van fekete pixel. Ezt a műveletet EXB-vel fogom jelölni. A fekete pixelek számát a következő művelettel csökkenthetjük (shrink black areas):

$$c(i,j) = f(i-1,j-1) \otimes f(i-1,j) \otimes f(i-1,j+1) \otimes f(i,j-1) \\ \otimes f(i,j) \otimes f(i,j+1) \otimes f(i+1,j-1) \otimes f(i+1,j) \otimes f(i+1,j+1), \qquad (3.29)$$

ahol $\otimes$ a logikai- és műveletet jelöli. A (3.29) képlet szemléletes jelentése a következő: az (i, j) pixel csak akkor marad fekete, ha a 3x3-as környezetében minden pixel értéke 1 volt, azaz fekete volt. Ezt a műveletet a továbbiakban SKB-val fogom jelölni.

17. ábra. A B struktúra elem.



Bináris képeken lévő alakzatok felismerését nagyban segíti, ha az alakzatok nem tartalmaznak kontúrra utaló zajt (haj). A haj eltávolítására szolgáló művelet az előbbi két művelet segítségével adható meg. Először végre kell hajtanunk az SKB-t, majd az eredményül kapott képen végrehajtani az EXB műveletet. Az alakzatok tartalmazhatnak repedéseket. Ezen repedéseket szintén eltüntethetjük, ha először az EXB műveletet hajtjuk végre, majd az eredményül kapott képen az SKB-t.

Bináris képeken az éldetektálás igen egyszerű, a következő művelet végrehajtását jelenti.

$$c(i,j) = f(i,j) \otimes \neg(f(i-1,j-1) \otimes f(i-1,j) \otimes f(i-1,j+1) \\ \otimes f(i,j-1) \otimes f(i,j+1) \otimes f(i+1,j-1) \otimes f(i+1,j) \otimes f(i+1,j+1)), \quad (3.30)$$

ahol a $\neg$ jel a logikai negálás. Ez a formula szemléletesen azt jelenti, hogy egy fekete pixel élpixel, ha a 3x3-as környezetében van fehér pixel.

A dilatáció az A bináris kép fekete területeinek "megnövelése", egy adott B struktúra elemmel. Az eredményül kapott új képet $A \oplus B$-vel jelöljük. A pontos definíció a következő:

$$A \oplus B = \cup_{a \in A} B_a, \quad (3.31)$$

ahol B_z a B halmaz eltoltja a z vektorral, $B_z = \{b + z \mid b \in B\}, \forall z \in Z^2$ (Z az egész számok halmaza).

Maszkok segítségével ez a művelet a következő módon adható meg. A B struktúra elemet a 17 ábra mutatja. A kiinduló A bináris képet pedig a 18 ábra tartalmazza. A dilatáció eredményét a 19. ábrán láthatjuk.

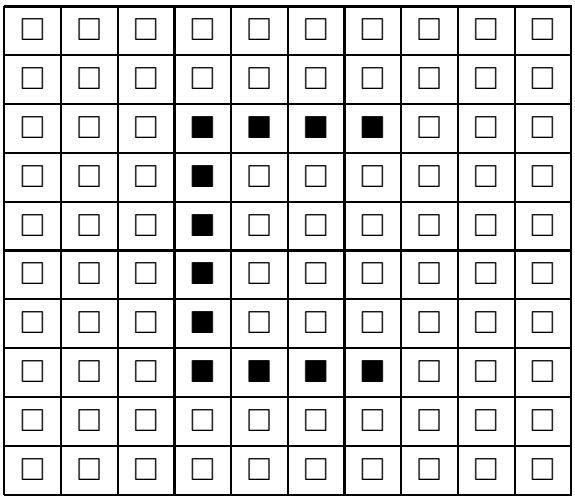
Egy másik művelet az erózió. Itt is szükségünk van egy B struktúra elemre, ha a kiinduló kép A, akkor a művelet eredményét a következő módon jelöljük: $A \ominus B$ és így adható meg:

$$A \ominus B = \{z \in Z^2 \mid B_z \subseteq A\} \quad (3.32)$$

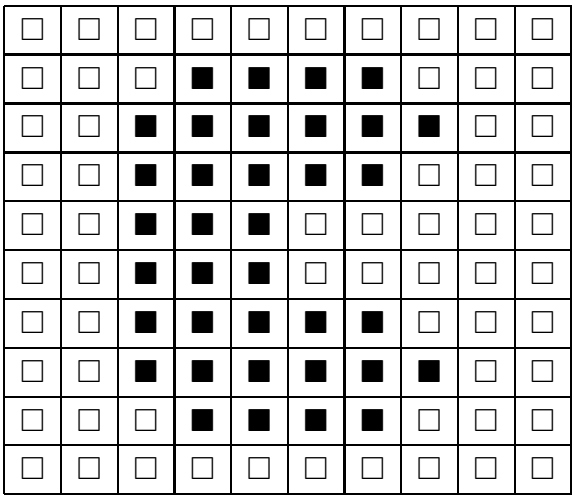
Ez a művelet a kiinduló A kép fekete pixeljeinek a számát "csökkenti" a B struktúra elemnek megfelelően. Ha ugyanazt a struktúra elemet használjuk mint a dilatációnál és a kiinduló kép a 20. ábrán látható, akkor az erózió eredményét a 21 ábra mutatja.

Az eróziót és a dilatációt általában párba szokták alkalmazni. A leggyakrabban használt műveletek a nyitás és a zárás. A nyitás eltűntet izolált pontokat a képről, amelynek a

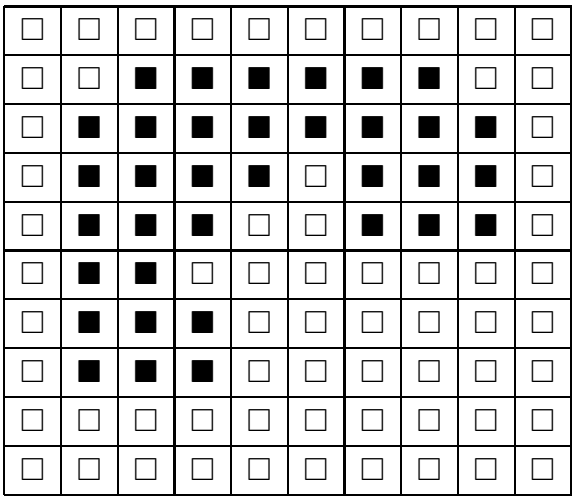
18. ábra. A kiinduló A bináris kép.



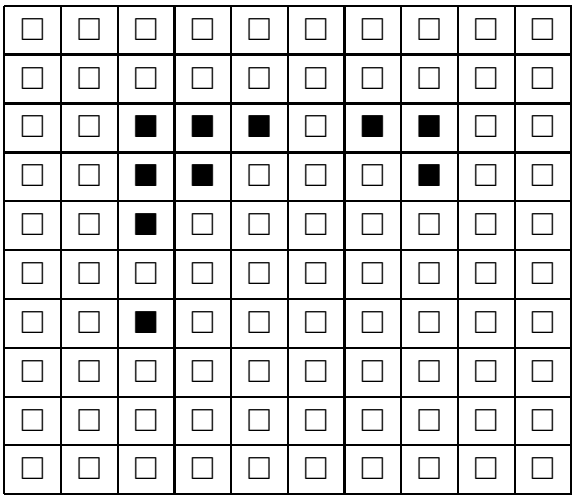
19. ábra. Az erózió műveletének eredménye a 18. ábrán látható képre. A struktúra elem a 17. ábrán látható.



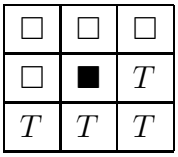
20. ábra. A kiinduló bináris kép az erózió műveletéhez.



21. ábra. Az erózió műveletének eredménye a 20. ábrán látható képre. A struktúra elem a 17. ábrán látható



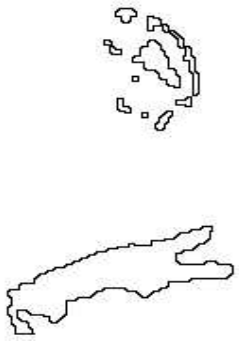
22. ábra. Bináris képen lévő alakzat kezdetének felismeréséhez szükséges maszk. A T-vel jelölt helyen a képpontok értéke tetszőleges lehet.



mérete kisebb mint a B struktúráló elem. A nyitási műveletnél először egy eróziót, majd egy dilatációt hajtunk végre. Az előző jelöléseket használva a nyitási művelet így írható le: $(A \ominus B) \oplus B$. A zárási művelet betölti a fekete pixelek hiányát (lyukakat), ekkor először egy dilatációt kell végrehajtani, majd az eredményül kapott képen egy eróziót. Képletben $(A \oplus B) \ominus B$.

A háttér levonás és bőrszín felismerés utáni képen még több alakzat található. Ezeket a blobokat azonosítanunk kell, és meg kell határoznunk a méretüket. Az egyszerűség kedvéért az alakzat kerületét választottam méret paraméterként. Az alakzat kerülete az alakzat éleit alkotó pixelek számával egyenlő. A blob azonosítás menete a következő. A [11]. műben ismertetett eljárást tekintettem alapnak. A bemeneti képből készítünk egy új képet, ami csak az éleket tartalmazza, hívjuk ezt élképnek. A kimenő kép (egészeket tartalmazó tömb) minden egyes pixelének helyére nullát írok. A blob analizálás következő lépésében a blobszámlálót nullára állítjuk. Ezután elindítunk egy raszteres pásztázást soronként. Amikor a 3.4. ábrának megfelelő maszk illeszkedik egy pixelre, akkor azt a pixelt egy blob kezdetének vesszük, és növeljük a blobszámlálót. A kimeneti képen az adott pixelre beírjuk a blobszámláló értékét. A megtalált pixel természetesen szerepel az élképen fekete pixelként, jelezvén, hogy ez a pixel része az élképnek. Ezután az élképet addig követjük, amíg vissza nem érünk a kiinduló ponthoz. Természetesen ahogy az élkép mentén haladunk a kimeneti kép azon pixeleire ahol végig haladtunk, beírjuk az blobszámlálót. Ha visszatértünk a kiindulási ponthoz, folytatjuk a soronkénti raszteres pásztázást addig, amíg újabb a 3.4. maszkra illeszkedő pixelt nem találunk. Ezzel az eljárással a blobokat és azok méreteit könnyen lehet azonosítani. Ha nem voltunk elég gondosak, akkor egy pixelből több él is kiindulhat, és ezeket az eljárás folyamán figyelembe kell venni. Ezt a problémát több módon is kiküszöbölhetjük, egyik módszer például az, hogy a bináris kép "hajait" (azon pixelek sorozata, amelyen hurok figyelése nélkül nem tudunk visszatérni a kiinduló pixelhez) megszűntetjük.

23. ábra. Az utófeldolgozás utolsó lépésével kapott kép összetett környezet esetén. A kiinduló kép a 16. ábrán található.



24. ábra. A bináris éldetektálás alkalmazása a 23. ábrán mutatott képre.

3.5. Utófeldolgozás

A háttérlevonás és bőrszín szegmentálás után kapott képen még további műveleteket hajtok végre azért, hogy a kisebb méretű blobokat eltávolítsam a képről. Erre azért van szükség, mivel a kéz helyzetének kijelölése egyszerűbb, ha csak néhány blob van a képen. Az utófeldolgozás során a [5] cikkben ismertetett eljárást követtem.

Az utófeldolgozás első lépésében az eredeti pixelrács helyett egy új, kisebb pixelrácsot vezetek be. Az eredeti pixelrácsot felosztom 8x8-as egységekre. Így kapok egy új, kisebb méretű pixelrácsot. Az első lépésben egy 8x8-as egység fekete vagy fehér voltát a következő szabállyal állapítom meg. Ha a 8x8-as területen van egyetlen fehér pixel, akkor az egész 8x8-as egység fehér színű lesz az eredeti pixelrácson és a kisebb méretű pixelrácson is. A 8x8-as pixelrács akkor lesz fekete, ha minden pixele fekete. Ha az utófeldolgozás első

lépését a 3. ábrán látható képre alkalmazom, akkor eredményül a 11. ábra baloldali részén látható képet kapom. Láthatjuk, hogy esetünkben a 8x8-as egységekre való felbontás nem működik. Megpróbáltam 4x4-es és 2x2-es egységekkel is végrehajtani az eljárást, az eredményeket a 11. ábra jobb oldali és középső része mutatja. Ezen ábra alapján nyilvánvaló, hogy az általam alkalmazott kamera és felbontás esetén csak a 2x2-es egységekre való felosztás használható.

Az utófeldolgozás második lépése a 2x2-es kisebb méretű pixelrácson dolgozik. Egy fekete 2x2-es egység fehér lesz, ha kevesebb, mint öt fekete 2x2-es egység veszi körül. Egy fehér 2x2-es egység fekete lesz, ha több mint két 2x2-es fekete egység veszi körül. A megadott számok itt az adott képpont 3x3-as környezetére vonatkoznak. Ezen lépés eredményeként kapott képet a 12. ábra mutatja.

Az utófeldolgozás végső, harmadik lépése a rövid függőleges és vízszintes vonalak eliminálását jelenti. Négynél rövidebb hosszúságú egybefüggő 2x2-es egységeket törlünk a képről. Az itt leírt lépéseket a [5] cikk ajánlotta. Ezeket a lépéseket még kiegészítettem egy zárási (closing) művelettel azért, hogy a bőrfelületek kisebb repedéseit, lyukait megszüntessem. Az utófeldolgozás végül a 13. ábrán látható képet szolgáltatta.

A 13. ábrán láthatjuk, hogy csak néhány egybefüggő fekete tartomány található a képen. Ezek közül kell kiválasztanunk azt, amelyik a kezet mutatja. Feltételezhetjük, hogy a háttérlevonás, bőrszín szegmentálás és utófeldolgozás után már valószínűleg a kéz a legnagyobb egybefüggő fekete terület a képen. Ezért a hátralévő feladat az, hogy az utófeldolgozás végső lépésében kapott képen lévő blobok közül kiválasszuk azt, amelyiknek a legnagyobb a kerülete. Ehhez szükség van a képen található alakzatok éleinek meghatározására. Bináris képen történő éldetektálás módszerét az előző fejezetben leírtam. Az ezzel a módszerrel kapott képet mutatja a 14. ábra.

A bonyolult, komplex háttér esetén az utófeldolgozás három lépése során kapott képeket a 15., 16. és 23. ábrák mutatják. Ezen ábrák kiinduló lépése a 9. ábrán látható kép volt. A komplex hátterű képen végrehajtott bináris éldetektálás eredménye a 24. ábrán látható.

A legnagyobb méretű alakzat kiválasztására az általam választott eljárást az előző fejezet legvégén ismertettem. Az eljárás felismeri, hogy melyik blobnak a legnagyobb a kerülete, és kijelöl egy téglalap alakú tartományt a képen, amely a kezet tartalmazza. A szegmentálásunk utolsó lépése, hogy az így meghatározott téglalap alapú területen újból végrehajtunk egy bőrszín felismerést. Az így nyert képet az eredményül kapott téglalappal együtt a 25. ábrán láthatjuk. A könnyebb összehasonlítás kedvéért ez az ábra a kiinduló képet is mutatja.

A szegmentálás ezzel véget ért, ha minden sikeres volt (a bemutatott esetben igen), akkor csupán egyetlen összefüggő fekete terület, a bekarikázott kézjel marad a képen. A bőrszín felismerés, háttérlevonás és utófeldolgozási lépések végrehajtásával tehát elértük a célunkat. A bonyolult hátterű 2. ábrán látható képből származtattunk egy olyan képet, amely csak egyetlen kézjelet mutat. A még hátralévő feladat, hogy végrehajtsuk a tulajdonképpeni kézjel felismerést.

25. ábra. A kiinduló összetett hátterű kép (bal oldal) és a szegmentálás végső eredménye (jobb oldal) amit a legnagyobb kerületű alakzat kiválasztásával nyertem a 23. ábrán látható képből.

4. Kézjel felismerés

Ha a felismerendő kézjelek előre meghatározott halmazokba esnek, akkor a példa-alapú megközelítést választhatjuk. Ez azt jelenti, hogy az alkalmazás két lépésből áll, tanulásból és végrehajtásból. A tanulási fázisban a felhasználó ismert (példa) kézjeleket mutat a számítógépnek, a gép a kézjeleket elemzi az eredményeket eltárolja. A végrehajtási fázisban a számítógép a felismerendő képet ugyanúgy analizálja, mint a tanulási fázisban, és összehasonlítja a jellemzőket a korábbi eltárolt értékekkel, majd dönt arról, hogy milyen alakot mutatott a felhasználó.

Az általunk alkalmazott alakfelismerés folyamata a következő. Minden képhez hozzárendelünk egy sajátság-vektort (feature vector). A sajátság-vektor magába foglalja a legfontosabbnak vélt információkat a képről. A feladat nehézsége abban rejlik, hogy megfelelően válasszuk a sajátság-vektort.

Ha például egyszerűen a pixelekhez tartozó intenzitások értékeit vennénk sajátság-vektornak, akkor ez nem lenne megfelelő. Például, ha a tanítóképen a kéz középen helyezkedik el, és a felismerendő képen ugyanaz a kéz a kép jobb oldalán van, akkor a két sajátság-vektor teljesen különböző lenne. Legyen most a kéz a tanító képen és a felismerendő képen is középen, de a megvilágítási viszonyok legyenek különbözőek, ekkor is a két sajátság-vektor teljesen különböző lenne.

Tegyük fel, hogy döntöttünk arról, milyen sajátság-vektort választunk. A tanítási fázisban, ha N_t számú tanítóképet készítünk, akkor kapunk N_t darab sajátság-vektort. A tanulási fázis során létrejön egy adatbázis amely a sajátság-vektorokat tartalmazza

$$\text{adatbázis} = \text{sajátság-vektorok halmaza} = \{\phi_1, \phi_2, \ldots \phi_{N_t}\}. \tag{4.33}$$

.

A felismerendő képhez tartozó sajátság-vektor legyen ϕ. A felismerendő képet azzal a tanító képpel azonosítjuk, amire

$$||\phi_i - \phi|| \tag{4.34}$$

minimális. Ez egy nagyon egyszerű eljárás, de esetünkben, az általam végzett kísérletek szerint jól működik. Ezt az egyszerű felismerési eljárást használták a [14, 15] munkákban is. Bonyolultabb esetben használhatjuk például a neurális hálózatokat.

4.1. Lokális orientációs hisztogram technika

A lokális orientációs hisztogram technika a megvilágítási viszonyoktól eléggé független és eltolás invariáns sajátság-vektort szolgáltat. A lokális orientációs hisztogram technikát, Connell publikálta 1986-ban és ötletét szabadalmaztatta is [13].

A módszert eredetileg szürkeskálás képre fejlesztették ki, ekkor a kép egy $I(x, y)$ kétváltozós valós értékű függvényként fogható fel. Az $I(x, y)$ függvény megadja az (x, y) koordinátájú képpont világosságkódját. Minden képponthoz meghatározhatjuk a gradiens vektort

$$v(x, y) = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \\ \frac{\partial I}{\partial y} \end{pmatrix}, \tag{4.35}$$

illetve annak hajlásszögét

$$\alpha(x, y) = \arctan\left(\frac{\frac{\partial I}{\partial y}}{\frac{\partial I}{\partial x}}\right). \tag{4.36}$$

A gradiens vektorok hajlásszögeinek értékeiből megkonstruáljuk a sajátság-vektor komponenseinek értékeit

$$\phi_k = \sum_{x,y} \begin{cases} 1 & \text{ha } \left|\alpha(x, y) - \frac{360^0}{N} k\right| < \frac{360^0}{N}, \\ 0 & \text{egyébként,} \end{cases}$$

ahol N a ϕ sajátság-vektor komponenseinek száma. A sajátság-vektor k-adik komponense, azaz ϕ_k megadja, hogy hány olyan (x, y) képpont van, hogy az ottani gradiens vektor hajlásszögére teljesül, hogy

$$\frac{360^0}{N}(k - \frac{1}{2}) < \alpha(x, y) < \frac{360^0}{N} - (k + \frac{1}{2}). \tag{4.37}$$

A ϕ_k számokból készítünk egy hisztogramot, és ezt nevezik lokális orientációs hisztogramnak. Az irodalom alapján [14, 15] a programomban a hisztogram elkészítéséhez

N értékének 36-ot választottam. Az így definiált orientációs hisztogramot célszerű még simítani. Erre az 1, 4, 6, 4, 1 súlyokkal definiált maszkot [14] használtam. Ez azt jelenti, hogy ϕ_k új értéke a következő lesz:

$$\phi_k \rightarrow \frac{1}{16}(\phi_{k-2} + 4\phi_{k-1} + 6\phi_k + 4\phi_{k-1} + \phi_{k-2}). \tag{4.38}$$

A gradiens vektor definíciójában természetesen a parciális deriváltat csak közelítően tudjuk számolni, amelyet a következőképpen határozunk meg:

$$\frac{\partial I}{\partial x} \approx \frac{I(x + \Delta x, y) - I(x - \Delta x, y)}{2\Delta x}. \tag{4.39}$$

Hasonlóképpen az y szerinti parciális derivált a következőképpen kerül meghatározásra:

$$\frac{\partial I}{\partial y} \approx \frac{I(x, y + \Delta y) - I(x, y - \Delta y)}{2\Delta y}. \tag{4.40}$$

A lokális orientációs hisztogram technikát sok területen alkalmazzák. Például kép hátterének eltávolítására [16] vagy szervomechanikus rendszerek irányítására [17]. A módszert több irányba is továbbfejlesztették pl. relatív képkoordináták bevezetésével [18].

4.2. Momentumok módszere

A bináris képen szereplő adott B alakzat momentumait a következő képlettel számíthatjuk ki [12] :

$$m_{rs} = \sum_{p \in B} x(p)^r y(p)^s. \tag{4.41}$$

A fenti képletben az összegzés az adott B alakzat összes p képpontjára terjed ki. Itt $x(p)$-vel a p pixel x irányú koordinátáját és $y(p)$-vel a p pixel y irányú koordinátáját jelöltem. Azaz ha $p = (i, j)$, akkor $x(p) = i$ és $y(p) = j$. Az r és s indexek értékei természetes számok lehetnek.

Speciális r és s értékeknek nagyon szemléletes jelentése van. Ha $r = 0$ és $s = 0$, akkor m_{00} megadja az alakzathoz tartozó képpontok darabszámát. Az előbbiekben definiált momentumok nagyon függnek a koordináta rendszer kezdőpontjának megválasztásától. Ezt kiküszöbölendő, vezessük be az alakzat x és y irányú tömegközéppontját $\bar{x}$-et és $\bar{y}$-ot. A tömegközéppont a momentumok segítségével a következő formában írható:

$$(\bar{x}, \bar{y}) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right). \tag{4.42}$$

A centralizált momentumok már eltolás függetlenek, mivel a tömegközépponthoz viszonyítanak. Definíciójuk a következő:

1. táblázat. A 27. ábrán látható jelek Hu-féle momentumai.

	kő	papír	olló
I_1	0.198194	0.204003	0.230450
I_2	0.009907	0.015485	0.019138
I_3	0.001123	0.000091	0.000428
I_4	0.000131	0.000015	0.000567
I_5	0.000000	0.000000	0.000000
I_6	0.000012	0.000002	0.000070
I_7	0.000000	0.000000	0.000000

2. táblázat. A 27. ábrán látható jelek kilencven fokos elforgatásával nyert kézjelek Hu-féle momentumai

	kő	papír	olló
I_1	0.198980	0.204642	0.229759
I_2	0.010050	0.015745	0.019133
I_3	0.001194	0.000086	0.000402
I_4	0.000142	0.000013	0.000528
I_5	0.000000	0.000000	0.000000
I_6	0.000014	0.000001	0.000067
I_7	0.000000	0.000000	0.000000

$$\mu_{rs} = \sum_{p \in B} (x(p) - \bar{x})^r (y(p) - \bar{y})^s. \tag{4.43}$$

Azért, hogy skála invariáns, méret független momentumot kapjunk, bevezették a normalizált momentumokat

$$\eta_{rs} = \frac{\mu_{rs}}{\mu_{00}^{\gamma}}, \tag{4.44}$$

ahol $\gamma = \frac{r+s}{2} + 1$ és $r + s \geq 2$.

Hu javasolta az ortogonális momentum invariánsakat, amelyek már forgás függetlenek is. Ezeket a momentumokat gyakran használják alakfelismerésre. Az ortogonális momentumok a normalizált momentumokkal fejezhetők ki:

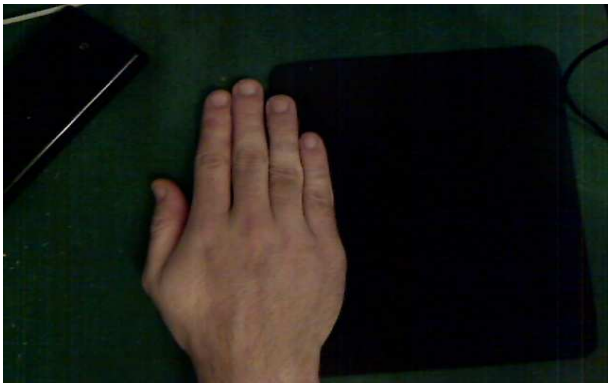
$$\begin{aligned}
I_1 &= \eta_{20} + \eta_{02} \\
I_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_1 1^2 \\
I_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
I_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
I_5 &= (\eta_{30} - 3\eta_{12}(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\
&- 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
I_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \\
&+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})^2] \\
I_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\
&- 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned} \tag{4.45}$$

A fenti hét képlettel definiált hét szám adja meg a sajátság-vektor komponenseit a momentumok módszerében.

5. Kísérleti eredmények

Korábbi fejezetekben bemutattam a szegmentálás folyamatát. A háttérlevonás, bőrszínfelismerés és utófeldolgozás után az eredményül kapott bináris kép már csak a kézjelet tartalmazza. Az előző fejezetben a sajátság-vektorok meghatározásának két fajta módszerét ismertettem.

Az orientációs hisztogram technika esetén nem kell a szegmentálást elvégezni. Az eredeti kiinduló színes képet kell szürkeskálás képpé konvertálni. Ezután a hisztogram az intenzitás értékekből kiszámítható. Ebben a fejezetben megvizsgálom, hogy mitől függ az orientációs hisztogram és hogyan célszerű azt meghatározni. Az orientációs hisztogram technikát két probléma megoldása céjából vezették be. Olyan módszert akartak, amely



26. ábra. Tesztkép egyszerű háttér esetén. A háttér eltér a 27. ábrán látható tanítóképek hátterétől.

27. ábra. Tipikus képek a tanulási fázisból.

a megvilágítási viszonyoktól lehetőleg nem függ, valamint a sajátság-vektor legyen eltolás invariáns. Ez utóbbi tulajdonságot egyszerűen úgy érték el, hogy a hisztogram meghatározásában a teljes kép vett részt. Az orientációs hisztogram módszer nagy előnye, hogy bonyolult képfeldolgozási módszerek alkalmazását nem igenyli.

A momentumok módszere esetén alapvetően fontos a szegmentálás, mivel a momentumok a geometriai alakzatot jellemzik. Ez a módszer eltolás, forgás és skála invariáns sajátság-vektort eredményez.

A két fajta módszer abban is különbözik egymástól, hogy a momentumok módszere esetén időben teljesen elválhat a tanítóképek készítése a felismeréstől. Ezzel szemben az orientációs hisztogram módszer esetén, mivel a hisztogram kiszámolása az egész képre vonatkozik, a tanítóképeket minden új háttér és minden új felhasználó esetén el kell készíteni.

Ebben a fejezetben megvizsgálom mindkét módszer teljesítőképességét. Ehhez készítettem egy tanítóképekből álló adatbázist, és egy tesztképeket tartalmazó adatbázist. A módszerek teljesítőképességét azzal mértem, hogy a tesztképek adott sorozata mellett hány képet ismer fel jól az eljárás. Hogy a kapott eredmény megbízható legyen, ahhoz nagy számú tanítóképet és nagy számú tesztképet kellett volna használnom. Sajnos mindkét adatbázisom csak kb. 50-50 képet tartalmazott.

5.1. Az orientációs hisztogram és momentumok módszerének vizsgálata

Az orientációs hisztogram technikát szürkeskálás képekre fejlesztették ki. A webkamera azonban színes képet készít, ezért először az RGB színtérről áttérek a szürkeskálás színtérre a következő formula segítségével:

$$I(x,y) = 0,3R(x,y) + 0,59G(x,y) + 0,11B(x,y), \tag{5.46}$$

ahol $R(x,y)$, $G(x,y)$ és $B(x,y)$ az (x,y) képponthoz tartozó RGB komponenseket jelöli.

28. ábra. A 27 ábrán látható képek szürkesklálás transzformáltjai.

Mind a tanulási, mind a végrehajtási fázisban, amikor egy képet analizálunk, az első lépés az, hogy minden képpontban kiszámítjuk a gradiens vektort. Vagyis minden pixelhez hozzárendelünk egy kétdimenziós vektort. A 29. és 30. ábrákon látható képek a gradiens vektorokat mutatják. Minden egyes pixel esetén az ottani gradiens vektort ábrázoltam. A 29. ábrán azok a gradiens vektorok láthatók, ahol a gradiens vektor nagysága nagyobb, mint 10, a 30. ábrán pedig azok a gradiens vektorok látszódnak, amelyeknek a nagysága nagyobb, mint 2.

A kézjel felismerés következő lépése, hogy előállítom az orientációs hisztogramot. Ekkor felmerül az a kérdés, hogy mely képpontok vegyenek részt a hisztogram elkészítésében. Korábbi munkákban [14, 15] azt javasolták, hogy ne minden pixel vegyen részt a hisztogram

29. ábra. A 28. ábrán látható képhez tartozó gradiens vektorok, amikor $||v|| > 10$.

30. ábra. A 28. ábrán látható képhez tartozó gradiens vektorok, amikor $||v|| > 2$.

elkészítésében, hanem csak azok a pixelek, ahol a gradiens vektor nagysága nagyobb, mint a gradiens vektorok átlagos nagysága. Tehát csak azok a pixelek vesznek részt az orientációs hisztogram kialakításában, ahol teljesül, hogy

$$||v|| > cM. \tag{5.47}$$

3. táblázat. Momentumok módszere esetén a sikeres felismerések száma a tanítóképek számának függvényében. A tesztkézjelek száma 10 volt. A felismerendő kézjel papír volt. Tipikus tanító és tesztkézjelek a 27. ábrán láthatók.

tanító képek száma	1	2	3	4	5
sikeres felismerések száma	4	6	9	10	10

A fenti egyenlőtlenségben M a gradiens vektorok nagyságának átlaga a teljes képre nézve

$$M = \frac{1}{\text{pixelek száma}} \sum_{x,y} ||v(x,y)||. \tag{5.48}$$

Ha a c-t megfelelően választjuk meg, akkor a felismerési hatékonyság javulhat.

Ahhoz, hogy megválaszthassam a c értékét, megvizsgáltam a gradiens vektorok nagyságának eloszlását. Meghatároztam a gradiens vektorok átlagos nagyságát, valamint a legnagyobb értékét. Ez utóbbira azért volt szükség, hogy megadhassam azt az intervallumot, ahol a hisztogramot el kell készíteni.

A következőket találtam: A 27. ábrán látható képek esetén a gradiens vektor nagyságának átlaga 2,4, 2,7 és 2,2. A maximális hossza a gradiens vektoroknak rendre 87,8, 75,5 és 58,3. A gradiens vektorok nagyságának eloszlását a 31. ábra mutatja. Ebből az ábrából láthatjuk, hogy a gradiens vektorok nagysága szinte mindig a $[0, 10]$ intervallumba esik.

4. táblázat. A sikeres felismerések száma. A tanítóképek és tesztkézjelek száma 10-10 volt. Tipikus tanítókézjelek a 27. ábrán láthatók. Egy tipikus tesztkézjelet pedig a 26. ábra mutat.

	kő	papír	olló
orientációs hisztogram	0	3	6
momentumok módszere	9	6	10

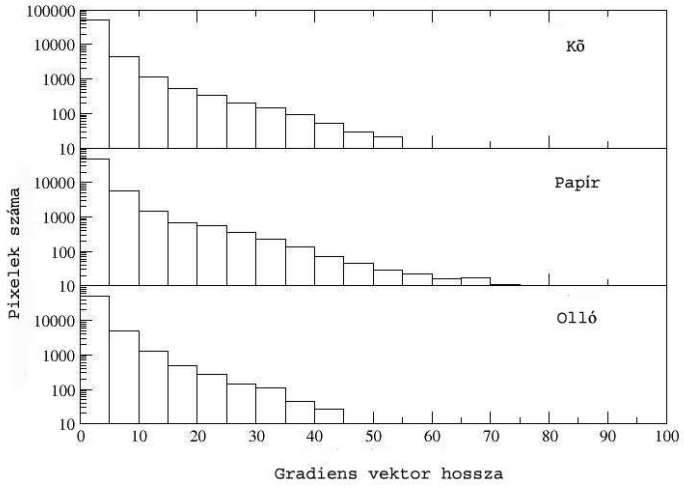
Ebből arra következtethetünk, hogy azok a pixelek, amelyek gradiens vektorának nagysága kisebb mint 10, valószínűleg a hátteret adják.

Így, ha azt szeretnénk, hogy csak azok a képpontok kerüljenek be az orientációs hisztogramba, ahol a gradiens vektorok nagysága nagyobb mint 10, akkor c értékét 5-nek kell választani ($||v|| > cM$ és $M \approx 2$).

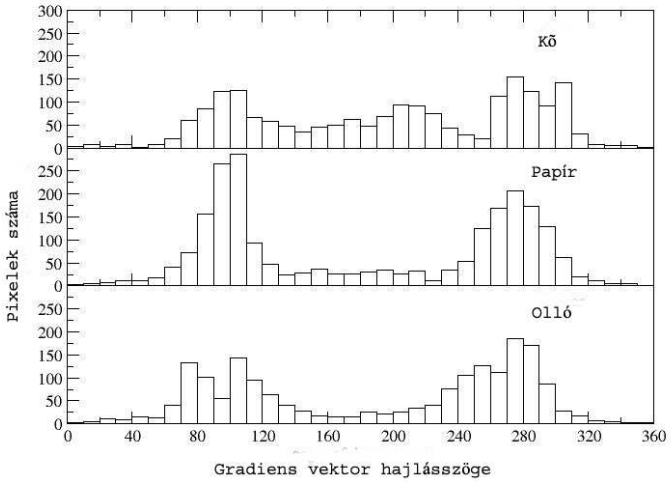
Az 27. ábrán látható mindhárom kézjel orientációs hisztogramját a 32., 33. és 34. ábrák mutatják. Az ábrák között az a különbség, hogy a 32. ábrán csak azok a pixelek vettek részt az orientációs hisztogram kialakításában, ahol $||v|| > 10$, míg a 33. ábrán csak azok a képpontok adtak járulékot az orientációs hisztogramhoz, ahol $||v|| > 2$. Végül a 34. ábrán nincs korlátozás a gradiens vektor nagyságára $||v||$.

A 32. ábrán látható, hogy a három kézjel orientációs hisztogramja jól elkülöníthető még szabad szemmel is. A 33. ábrán már alig lehet látni a különbséget a kézjelek hisztogramjai között, ellenben a 34. ábrán szabad szemmel már nem lehet különbséget tenni az orientációs hisztogramok között.

Annak eldöntésére, hogy a program milyen eredményesen ismeri fel a kézjeleket, a következő kísérletet végeztem. Először mind a három kézjelről készítettem tíz-tíz darab tanítóképet. Majd az ezekhez tartozó 36 elemű sajátság-vektorokat eltároltam. Ezután ugyanazon felhasználó, mint aki a tanítóképeket készítette el, mutatott tíz-tíz új kő, papír és olló képet ugyanolyan egyszínű háttér mellett. A továbbiakban ezeket az új kézjeleket (tesztképeket) arra használtam, hogy teszteljem a felismerés jóságát.



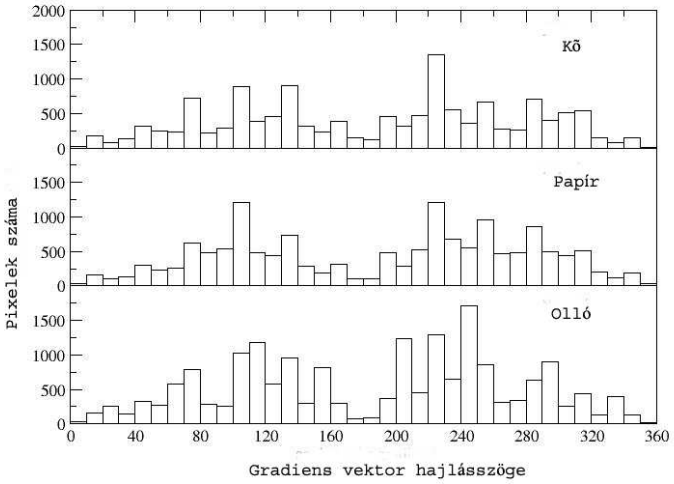
31. ábra. A 28. ábrán látható képek elemzése. A gradiens vektorok nagyságának hisztogramja.



32. ábra. A 28. ábrán látható képek orientációs hisztogramjai, amikor $||v|| > 10$.

Az egyértelmű, hogy a tanítóképek számának növekedésével javul a felismerés hatékonysága. Meglepett az az eredmény, hogy olyan egyszerű háttér mellet mint amilyen a 27. ábrán látható egy, két tanítókép esetén már minden tesztképet felismert az az orientációs histogram módszer. Ezzel szemben a momentumok módszere esetén azt találtam, hogy papír jelet mutató tesztképek esetén egy, két tanítókép nem elegendő. Ezt mutatja az 3. táblázat is, ahol a tanítóképek számának függvényében a sikeres felismerések számát mutatom be. A táblázat szerint momentumok módszere esetén még ilyen egyszerű háttér estén is legalább négy tanítókép szükséges hogy a felismerés hatékonysága olyan jó legyen mint amilyen jó volt az orientációs hisztogram módszer.

Azt tapasztaltam, hogy papír tesztkép esetén azt szinte minig kőként azonosította a program. Hogy megértsem ezen tévedés okát megvizsgáltam három tipikus kézjel momentumait. Ezeket a számokat mutatja a 1. táblázat. Láthatjuk, hogy az olló kézjel I_2, I_4 és I_6 momentumai jelentősen eltérnek a kő és a papír kézjelhez tartozó momentumoktól. Ezzel szemben a kő és a papír kézjel momentumai nagyon hasonlóak. Ez érthető is hiszen a momentumok az alakzat geometriai formájától függnek, és ez a kő és papír jelek esetén eléggé hasonló. Megvizsgáltam azt is, hogy az elméletileg bizonyított forgatás invariancia mennyire teljesül. Ehhez azt a legegyszerűbb utat választottam, hogy a 27. ábrán látható képeket kilencven fokkal elforgattam az Adobe Photoshop programmal. Az így keletkezett kepek Hu-féle momentumait mutatja a 2. táblázat. A 1 és a 2 táblázatokat összehasonlítva láthatjuk, hogy a momentumok gyakorlatilag nem változtak. A kis eltérés oka valószínűleg az, hogy az eredeti képek tipusa bmp volt az elforgatás után pedig a kép tipusa jpg volt.



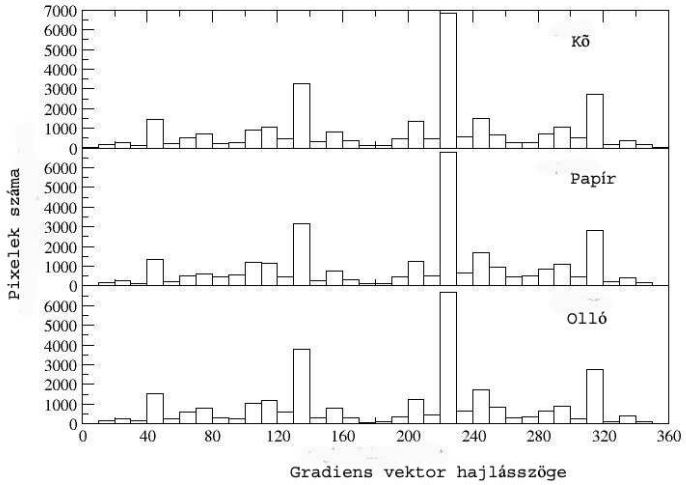
33. ábra. A 28. ábrán látható képek orientációs hisztogramjai, amikor $||v|| > 2$.

A momentumok módszerének nagy előnye, hogy a sajátság-vektor forgás invariáns. Ha 27. ábrán látható képek kilencven fokos elforgatottjait használtam felismerendő képnek, akkor a momentumok módszere négy tanítókép esetén azokat sikeresen felismerte. Ezzel szemben, mivel az orientációs hisztogram technika által adott sajátság-vektor nem forgás invariáns, így az elforgatott képeket elvileg nem lehet és a gyakorlatban sem tudtam felismertetni ezzel a módszerrel.

A különböző módszerek teljesítőképességének mérésére még egy kísérletet is elvégeztem. A tanítóképek halmazát nem változtattam meg. Az adatbázist tíz-tíz kő, papír és olló kép alkotja. A tipikus példák a 27. ábrán láthatók. Készítettem viszont tíz-tíz új tesztképet egyszerű környezet, háttér esetén. A teljesen egyszínű háttér helyett most egy kicsivel összetettebb a háttér, és a fényviszonyok is eltérőek voltak a tanítóképekhez képest. Egy tipikus új tesztképet mutat a 26. ábra. A sikeres felismerések számát mutatja a 4. táblázat. A kísérletet mindkét módszerrel elvégeztem. Az eredmények azt mutatják, hogy a momentumok módszere sokkal jobb, mint az orientációs hisztogram technika.

5.2. A számítógépes program ismertetése

Több funkciós programot készítettem C++ nyelven, Windows operációs rendszerben. Egy grafikus keretrendszert használva a felhasználónak lehetősége van arra, hogy egy adott adatbázisból betöltse a régebbi tanulóképeket leíró sajátság-vektorok komponenseit, vagy új tanulóképeket készítsen, ezáltal frissítve a meglévő adatbázist. A felhasználó arról is



34. ábra. A 28. ábrán látható képek orientációs hisztogramjai, amikor nincs korlátozás $||v||$-re.

dönthet, hogy játszani akar, vagy a helyi lemezen lévő, már előre elkészített képeket analizáljon, ismerjen fel.

Most röviden ismertetem az egyes grafikus felületek használatát. Először a felhasználó a "webkamera keresés" gombra kattintva megtudhatja, hogy milyen webkamerák vannak a rendszerben. Ezután egy legördülő menüből ki kell jelölnie, hogy melyik kamerát akarja használni. Az "indít" gombbal aktivizálódik a legördülő menüben kiválasztott webkamera, és egy ablakban megjelenik a kamera által mutatott felvétel. Az "állj" gomb segítségével megszűnik a kiválasztott kamera működése. A teljesség kedvéért ez a felület tartalmaz egy "fényképezz" gombot is, ellenőrizve azt, hogy jól fényképez-e a kamera, ami a későbbi élő játékban kritikus fontosságúvá válik. Az elkészített fényképek egy adott könyvtárban találhatók. A következő grafikus felület aktivizálásához a "játék indítása" gombot kell megnyomnia a felhasználónak.

Az új grafikus felületen, adatbázis betöltése esetén, nem szükséges megadni a tanuló képek számát, ellenkező esetben, azaz, ha vagy a merevlemezen lévő tanuló képeket akarjuk felhaszálni, vagy új tanuló képeket akarunk készíteni, akkor viszont a legördülő menü segítségével muszáj megadnunk a tanuló képek számát.

Ha a "tanuló képek betöltése" gombra, vagy az "adatbázis betöltése" gombra kattintottunk jöhet a következő lépés, de ha új tanuló képeket akarunk készíteni, akkor a következnünk kell a számítógép által adott szóbeli és írásbeli utasításokat, hogy elkészíthessük az új tanuló képeket. Mind a három esetben, ha rátérhetünk a következő lépésre, akkor erről

szöveges üzenetet kapunk.

A további lépések ugyan ebben a grafikus felületben történnek. Most a felhasználónak két lehetősége van. Az egyik, hogy a merevlemezen lévő képeket akar felismertetni, vagy el akarja indítani az élő játékot. Az első esetben a "kézjel felismerés" gombra, második esetben a "játék indítása" gombra kell kattintania. Kézjel felismerés esetén a program kiírja, hogy a felhasználó által kiválasztott képen milyen jelet ismert fel. Ha a felhasználó a második esetet, azaz a játék indítását választotta, akkor mielőtt ezt a gombot megnyomná, a legördülő menü segítségével meg kell adnia, hogy hány darab játékot akar játszani. Ezután követnie kell a számítógép által adott szóbeli és írásbeli instrukciókat. A játék során az élőben mutatott jelet a program analizálja, majd véletlenszerűen választ a gép számára egy kézjelet (követ, papírt vagy ollót) és összehasonlítja a felhasználó által mutatott jellel, és végül eredményt hirdet. Ez a folyamat a felhasználó által beállított játékok számáig fog tartani.

A játék indítása és az új tanuló képek készítése gomb megnyomása után megjelenő grafikus felület folyamatosan mutatja a webkamera által mutatott képet, valamint azt a fényképet, amit a videófolyamból kinyertünk aktuális kézjelként. Ezen grafikus felületek aktiválása után a következő utasítások hangozhatnak el: mutass egy jelet, mutass egy követ, stb. , majd pedig egy hang háromig számol és háromra a felhasználónak az általa kívánt jelet jól látható módon mutatnia kell. Ez lesz az aktuális kézjel.

A program több mint 6500 soros, ebből körülbelül 700 sor a webkamera működéséért felelős. Ez utóbbiba értendő a rendszerben lévő webkamerák megkeresése, azok inicializálása, valamint a videófolyamból való egy adott keret (kép) kinyerése. Ezeket a programokat a Microsoft weboldaláról, valamint egyéb webhelyekről töltöttem le. A több mint 5500 sort a hivatkozás jegyzékben lévő cikkek, valamint saját ötletek alapján írtam meg.

A program szerkezete gyakorlatilag követi azt a szerkezetet, amit a dolgozatomban leírtam. Először a háttérképeket készítem el és kiszámolom a (3.19, 3.20 és 3.22) mennyiségeket. Ezután következik az aktuális kép kinyerése a videófolyamból, majd pedig a háttér levonás (3.23) képlet segítségével. Az utófeldolgozás után történik a blob analizálás, ahol meghatározom azt a területet, ahol a kézjel elhelyezkedik. Ezután azon programrész aktivizálódik, amely elkészíti az adott képhez tartozó sajátság-vektort. Végül ezt a vektort a program összehasonlítja az adatbázisban tárolt sajátság-vektorokkal, és a gép meghozza a döntést a kézjel típusáról.

6. Összefoglalás

Az elképzelések szerint a kő-papír-olló játék úgy zajlana, hogy a felhasználó a számítógépe előtt ül és "beszédbe" elegyedik a géppel. A számítógéphez rögzített webkamera – a gép "szeme" – érzékelné a játékos által mutatott kézjelet. A Tanszék korábbi fejlesztésének eredményeként meglévő beszélő fej pedig közölné a gép által választott kézjelet, érzelmeket fejezne ki és utasításokat is adhatna a játék menetéhez. Például a beszélő fej számolna

háromig, és akkor kellene a játékosnak megmutatnia és kimondania a választott kézjelet.

A kő-papír-olló játék ilyen multi-modális megvalósításához szükséges egy olyan képfeldolgozást végrehajtó szoftver, amely bonyolult, összetett háttér előtt is meg tudja állapítani a játékos által mutatott kézjelet. Ezt az alap szoftvert sikerült megcsinálnom.

Mivel a kő-papír-olló játékhoz nem szükségesek játéktábla és játékfigurák, ezért a Tanszéken kifejlesztett sakkozógép robot karjára nincs szükség. Arra lenne csupán szükség, hogy a beszélő fejet a programomhoz integráljuk, és ekkor lényegében elkészülhetne egy multi-modális megvalósítása a kő-papír-olló játéknak.

A kézjel felismerés első része a szegmentálás. Ezt két lépésben végeztem el. Mielőtt a játékos mutatna egy kézjelet, elkészítek öt darab háttérképet, és ezek alapján előállítok egy háttérmodellt. Miután a webkamera által készített videófolyamból kinyertem a játékos által mutatott aktuális kézjelet, végrehajtom a háttér levonást. A levonás után már feltételezhetően a háttér nagy része eltűnik a képről, és csupán a felhasználó keze, néhány arcbőr foszlány és esetlegesen kis méretű bőrszínű háttér maradványok maradnak a képen.

A szegmentálás második részében a kéz által elfoglalt területet kell megjelölnöm. Erre a célra bőrszín felismerést alkalmazok. Pixel alapú bőrszín felismerést használok. Minden egyes pixelről eldöntöm hogy bőrszínű-e vagy sem. A bőrszín azonosításra az YC_bC_r színteret választottam. A C_b, C_r koordináta térben egy téglalap belseje jelöli ki azokat a C_b és C_r értékeket, amelyek bőrszínnek felelnek meg. Az interaktív alkalmazás miatt gyors bőrszín felismerésre volt szükségem, azért választottam ezt a fajta bőrszín felismerő eljárást. Úgy találtam, hogy a korrekt bőrszín felismerés igen fontos esetemben. Egy adott színtéren kijelölt bőrszín tartomány függ a megvilágítási viszonyoktól. Erre nézve találtam cikkeket az irodalomban, amelyek azt vizsgálták, hogy a bőrszíntartomány hogyan változik a fényt kibocsátó forrás hőmérsékletétől. Az általam választott módszerek azonban nem tesznek különbséget természetes napfény és belső, szobai világítás között. Azért, hogy a programom megbízhatóan és stabilan működjön, minden (de nem extrém) megvilágítás között, ahhoz a bőrszín felismerést kell javítani.

Miután sikerült a kéz helyét megtalálni a háttérlevonás és bőrszín szegmentálás segítségével, a hátralévő feladat a kézjel felismerése volt. Az általam választott módszer két lépésből áll. Először az úgynevezett tanulási fázisban elkészítek és elemzek lehetőleg minél több ismert kézjelet. A jellemzésre használt mennyiségeket (a sajátság-vektor komponenseit) egy adatbázisban eltárolom. A játék folyamán az aktuális kézjelre szintén meghatározom a sajátság-vektort, majd a döntési fázisban összehasonlítom a tanulási fázisban és a felismerési fázisban kapott sajátság-vektorokat. A felismerendő képet azzal a tanító képpel azonosítom, amire nézve a két (a tanító képhez és a felismerendő képhez tartozó) sajátság-vektor különbségének nagysága minimális. Ez az egyszerű felismerési eljárás esetemben jól működik. Valószínűleg azért, mivel csak három kézjelet kell felismerni. Bonyolultabb esetekben az általam használt módszernél jobb eljárásokat célszerű alkalmazni, például neurális hálózatokat.

Egy fontos kérdés annak az eldöntése is, hogy minek válasszuk a sajátság-vektorokat.

Dolgozatomban két lehetőséget is megvizsgáltam. Az első esetben a lokális orientációs hisztogramot készítettem el. Ekkor a webkamera színes képét szürkeskálássá konvertáltam. Minden egyes képponthoz tartozik egy, az intenzitásokból számolt gradiens vektor. A gradiens vektorok hajlásszögéből kell elkészíteni a hisztogramot. A leírásból látszik, hogy itt az egész kép részt vesz a hisztogram létrehozásában. Ez azt jelenti, hogy a tanítóképeket minden egyes új háttér és minden egyes új felhasználó esetében el kell készíteni. Egy másik hátránya az orientációs hisztogram technikának, hogy csak az eltolással szemben invariáns. A módszer nagy előnye viszont az, hogy lényegében nem igényel képfeldolgozást. A képpontok intenzitás értékeiből el tudjuk készíteni a hisztogramot.

A momentumok módszere esetén időben elválhat a tanítóképek elkészítése és játék (kézjel felismerés). A momentumok módszere által definiált sajátság-vektor már eltolás, forgatás és skála invariáns. Mivel a módszer a geometriai alakzat fizikai momentumait használja, ezért igen fontos, hogy nagyon pontosan ki tudjuk jelölni a kéz helyzetét. Nem elég csak egy téglalapot kijelölni, amely a kézjelet tartalmazza, hanem a momentumok kiszámításakor csak azok a képpontok vehetnek részt, amelyek tényleg a kézjelhez tartoznak. A momentumok módszere miatt volt szükségem a háttérlevonás, bőrszínfelismerés és blob-analizálás technikájának alkalmazására. Ez a módszer sokkal több képfeldolgozási módszert alkalmaz, mint az előző.

Vizsgálataim azt mutatták, hogy a momentumok módszere sokkal jobb, mint az orientációs hisztogramot alkalmazó eljárás. A módszer hatékonyságára csak egy elég durva becslést tudok adni. Ennek az az oka, hogy a legnagyobb adatbázisom sem volt túl nagy, csupán 50-50 tanítóképet tartalmazott minden kézjelről A hibás azonosítások számát (vizsgálataim alapján) olyan 10-20 százalékra becsülöm. Biztos vagyok benne, hogy sokkal nagyobb adatbázis esetén a hibás azonosítások száma tovább csökkenthető, és így a momentumok módszerével a kő-papír-olló játék multi-modális módon megvalósítható.

7. Köszönetnyilvánítás

Elsőként Dr. Fazekas Attilának szeretném megköszönni, hogy a képfeldolgozás témakörébe fokozatosan vezetett be. Kezdeti egyszerű feladatoktól (Tux-racer irányítása és bőrszínfelismerés) eljutottam a kő-papír-olló játék multi-modális megvalósításáig. Neki köszönhetem, hogy a TDK versenyen elindultam és sikeresen szerepeltem. Köszönöm neki segítségét, hogy nyári ösztöndíjat nyerhettem és, hogy egy évig mellette demonstrátorként taníthattam. Kovács Györgynek is köszönöm segítségét, aki megkönnyítette beilleszkedésemet és segített munkámban. Ezen kívül, családomnak, barátnőmnek Zsófinak, valamint barátaimnak köszönom támogatásukat és türelmüket.

Hivatkozások

[1] Juan P. Wachs, A Gesture-based Tool for Sterile Browsing of Radiology Images, Journal of the American Medical Informatics Association 15, 321 o. 2008, DOI 10.1197/jamia.M24 és http://www.medicalnewstoday.com/articles/111816.php

[2] O.M.Foong, T.J.Low és S.Wibowo Hand Gesture Recognition: Sign to Voice System (S2V), International Journal of Electrial, Computer and Systems Engieering, 3, 198 o. 2009

[3] S.Funck Video-Based Handsign Recognition for Intuitive Human-Computer-Interaction, Lecture Notes in Computer Science 2449, 1611 o., 2002

[4] P.Peer, J.Kovac és F.Sholina, Human Skin Colour Clastering for Face Detection, EUROCON1993, Ljubljana, Slovenia 144, 2003

[5] D.Chai és K.N.Ngan, Face Segmentation Using Skin Color Map in Videophone Applications, CirSysVideo(9), No. 4, 551.o, 1999

[6] W.Nabiyev és A.Günay, Towards a Biometric Purpose Image Filter According to Skin Detection, The Secont Internation Conference "Problems of Cybernetics and Informatics", 2008

[7] N.A.bin Abdul Rahman, K.C.Wei és J.See, RGB-H-Cb-Cr Skin Colour Model for Human Face Detection MMU International Symposium on Information and Communicationss Technologies 2006

[8] J.C. Terrillion, M.N. Shirazi, H. Fukamachi és S. Akamatsu, Comparative performance of different skin chrominance models and chrominance spaces for automatic detection of human faces in color images, Proc. of the International Conference on Automatic Face and Gesture Recognition, 54-61 o. 2000.

[9] M.Störing, H.J.Andersen és E.Granum, Skin colour detection under changing lighting conditions,7th Symposium on Intelligent Robotics Systems, 1999.

[10] I.Harritaoglu, D.Harwoodés L.S.Davis W^4: Real-Time Surveillance of People and Their Activities, IEEE Transactions on pattern analysis and machine intelligence vol 22. no.8, 809 o., 2000

[11] B.Batchelor és F.Waltz, Intelligent Machine Vision Techniques, Impelentations and Applications, Springer

[12] S.Conseil, S.Bourenane és L.Martin, Comparison of Fourier Descriptors and Hu moments for hand posture recognition, 1960 o., EUSIPCO, Poznan 2007

[13] R.K. McConnell, U.S. Patent No. 4,567,610 Jan. 1986.

[14] M. Roth, K. Tanaka, C. Weissman, W. Yerezunis, Computer Vision for Interactive Computer Graphics, IEEE Computer Graphics and Applications, May-June, 42. o., 1998.

[15] W. T. Freeman, M. Roth, Orientation Histograms for Hand Gesture Recognition, IEEE, Intl. Wkshp. On Automatic Face and Gesture Recognition, Zurich, June, 1995.

[16] D.H. Jang, X. Jin, Z.J. Choi, T.Z. Kim, Lecture Notes in Computer Science, 222. o., kötet. 5068/2008.

[17] J.M. Tavares, R. Ferreira, F. Freitas, Control a 2-Axis Servomechanism by GEsture Recognition using a Generic WebCam, International Journal of Advanced Robotic Systems, Volume 2, Number 1, 39. o. (2005)

[18] H. Zhou, D.J. Lin, T.H. Huang, Computer Vision and Pattern Recognition Workshop, 2004. CVPRW apos;04. Conference on Volume , Issue , 27-02 June 2004 Page(s): 161.