
TDK DOLGOZAT

SZERZŐ:

POZSEGOVICS PÉTER

Óbudai Egyetem

Neumann János Informatikai Kar



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

ANYBOARD, KITERJESZTETT VALÓSÁG ALAPÚ EMBER-GÉP INTERAKCIÓ

Szerző:

Pozsegovics Péter

mérnök informatikus, MSc. II. évf.

Konzulens:

Dr. Vámosy Zoltán

egyetemi docens

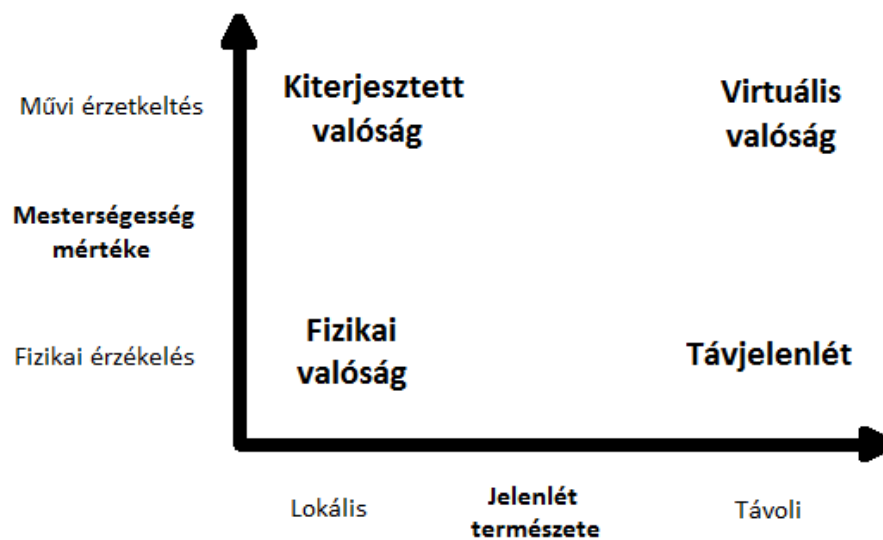
Tartalomjegyzék

1	BEVEZETŐ	4
2	ALKALMAZÁSOK	5
2.1	HADIIPAR	5
2.2	GYÓGYÍTÁS.....	5
2.3	SZÓRAKOZTATÁS.....	6
2.4	IPAR.....	6
3	TERVEZÉSI SZEMPONTOK.....	8
3.1	JELENETGENERÁLÁS.....	8
3.2	MOBILITÁS	9
3.3	3D REGISZTRÁCIÓ	9
4	SAJÁT RENDSZER.....	11
4.1	RENDSZERTERV	11
4.2	SZÍNKEZELÉS	11
4.3	KÉZFEJKÖVETÉS	12
4.4	JELŐLŐDETEKTÁLÁS	13
4.5	JELENETGENERÁLÁS.....	16
4.6	INTERAKCIÓ DETEKTÁLÁS	16
4.7	AFFIN KORREKCIÓ	16
5	ÖSSZEGZÉS.....	17
5.1	MEGVALÓSÍTÁS	17
5.2	TESZTELÉS	19
5.3	ÖSSZEVETÉS AZ EDDIGI TECHNIKÁKKAL, TOVÁBBFEJLESZTÉS	26
6	IRODALOMJEGYZÉK	28

1 Bevezető

A technológia fejlődésével és a rendelkezésre álló információ mennyiségének rohamos növekedésével egyre nagyobb az igény a hatékony perifériák kifejlesztésére is. Rengetegféle megközelítés létezik – az emberi agy ingereinek a közvetlen felhasználásától kezdve a végtagok vagy akár a szemgolyó mozgásával elért információtovábbításig – amelyek nem jöhettek volna létre a megfelelő modern hardver háttér nélkül. Ezen eszközök javarésze az ember mindennapi tevékenységeit hivatottak segíteni. Segítik a koordinációt, kapcsolódó információkkal látják el a használóját az adott szituációban, felhívják a figyelmet veszélyekre, érdekességekre. Elég csak az orvosi, a katonai, valamint civil alkalmazásokra gondolni, ahol az utóbbi időben rengeteg kutatás folyik.

Egy ígéretes megközelítés a virtuális technológiák használata, melynek az összefoglaló neve az úgynevezett kevert valóság. Ez magába foglalja a virtuális valóságot, a kiterjesztett valóságot és a távjelenlétet, melyet az 1. ábra szemléltet.



1. ábra – A kevert valóság osztályozása [1]

Az egyik véglet az távjelenlét, amely egy embert egy tőle távoli környezetbe képes helyezni egy ún. teleoperátor segítségével és ott úgy tud feladatokat elvégezni, mintha ténylegesen jelen lenne. Fontos megjegyezni, hogy ez nem azonos a virtuális valósággal, mivel itt a résztvevő érzése teljesen valódi, míg a virtuális valóságban teljesen művi. A kiterjesztett valóság a kettő ötvöze, tehát a valós világot láthatja a felhasználó virtuális objektumokkal kiegészítve vagy éppen valós objektumoktól megfosztva. A valós világ és a megfigyelő közé egy köztes réteget illeszt be és ezt a réteget általában arra használják, hogy számítógép generálta két- vagy háromdimenziós objektumokat illesszenek bizonyos helyekre. Az így létrehozott félvirtuális valóságban tulajdonképpen az ember érzékelését lehet feljavítani és az interakciós képességeit lehet kiterjeszteni. Egy kiterjesztett valóság alapú rendszer tervezésekor három fő szempontot kell figyelembe venni. Ezek a jelenet előállításának a módja, a hordozhatóság, illetve a 3D regisztráció. A következő részben a gyakorlati alkalmazásokon keresztül ismerhetjük ezek megvalósításait.

2 Alkalmazások

2.1 Hadiipar

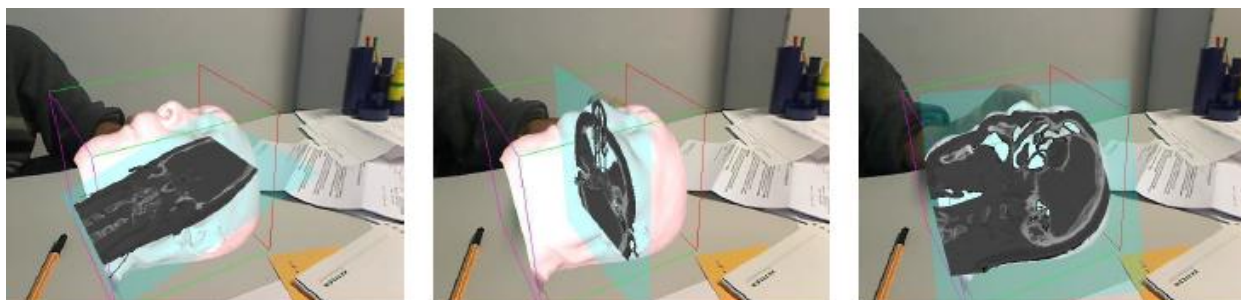
A történelem során a nagy felfedezések javarésze a hadiiparból került át civil életbe. Így van ez a kiterjesztett valósággal is. A legrégebbi alkalmazások a pilóták tevékenységeinek hatékonyságnöveléséhez köthetők, amilyen például a célzás vagy a manőverezés [1]. Azonban nem csak éles helyzetekben veszik hasznát a hadseregnek az ilyen technikáknak, hanem a kiképzés során is. Erre látunk példát a 2. ábrán. Gondoljunk csak bele, hogy tetszőleges harci helyzetet képesek szimulálni, ha az egész csatateret mesterségesen állítják elő. Ez egyfajta keveréke a korábban említett távjelenlétnek, a virtuális valóságnak és a kiterjesztett valóságnak.



2. ábra – Távjelenlét és kiterjesztett valóság a kiképzésen [2]

2.2 Gyógyítás

Ez az egyik legfontosabb alkalmazási terület, ahol rengeteg kutatás és fejlesztés folyik [3][4][5]. A különféle orvosi képalkotó eljárásokkal (MRI, CT) készített felvételek hatékonyabb vizualizálását már gyakran a kiterjesztett valóság technikákkal végzik [6]. Erre láthatunk egy példát az 3. ábrán. A felvételekből előbb egy 3D modellt állítanak elő, a beteg megfelelő testfelületére vetítik, így jobban meg tudják tervezni az orvosok a beavatkozás részleteit. Az előzőhöz hasonlóan lehetőség van az ultrahangos vizsgálat valósidejű vizualizációjára és 3D regisztrációjára. Ezáltal az orvos részletesebb képet kaphat a baba és az anya állapotáról. Léteznek már alkalmazások, amelyek nemcsak a beavatkozás megtervezésekor nyújtanak segítséget, hanem közben is. Segítségükkel a beavatkozó eszköz nem látható részeit illetve a beteg testrész viszonyát követheti nyomon az orvos [7].



3. ábra – Az MRI felvételek térbeli vizualizálása [6]

2.3 Szórakoztatás

A kiterjesztett valóság a szórakoztatásban is kiválóan alkalmazható. Gondoljunk mindjárt az időjárás jelentésre, ahol a bemondó egy homogén háttér előtt áll és a kép mögötte állandóan változik. Más műsorok esetén, például sportesemények közvetítése közben figyelhető meg, hogy játéktéren megjelenik egy-egy hirdetés, zászló vagy a játék szempontjából fontos információ. Erre látunk példát a 4. ábrán. Ezek is mind hasonló technikával kerülnek a képernyőre.



4. ábra – Hirdetés megjelenítése fizikai beavatkozás nélkül [2]

A médián kívül is lehet azonban alkalmazásokat találni. Egy népszerű alkalmazási terület például a társasjáték. A hagyományos táblajátékok esetén problémát okozhat az apró tartozékok elvesztése, illetve az anyaghasználat miatt fellépő rongálódás. Ezt ki lehet küszöbölni a kiterjesztett valósággal, ahol kellékek sokkal élethűbb hatást keltenek, animálhatóak és cserélhetőek, így nagymértékben fokozzák az játékélményt [8]. Erre láthatunk példát az 5. ábrán.



5. ábra – Monopoly és kiterjesztett valóság [9]

2.4 Ipar

Az iparban a különféle termékek tervezési fázisában hagyományosan papír alapú tervrajzokat használnak a tervezők. Ezek segítségével el tudják helyezni térben az adott munkadarabot, segítségükkel be tudnak vonni újabb tervezőket, ha szükséges. Ezzel megoldással a probléma, hogy

egy komplex terv esetén nehéz elképzelni a végterméket a valóságban, ezért a papír alapú megoldásokat felváltották az elektronikus tervezőeszközök, amelyek még komplexebb képet tudnak adni az adott munkadarabról. Ezek már segítik a tervezőt – tulajdonképpen kényszerítik – a fizikai paraméterek illetve a tervezési konvenciók betartására. Ez már kétségtelenül egy minőségi előrelépés a vizualizációban, azonban ezt még tovább lehet fokozni a kiterjesztett valóság alkalmazásával. A mérnökök munkáját nagyban segítheti, ha nem csak egy ablakon keresztül láthatja a munkadarabot lekicsinyítve, hanem akár életnagyságban úgy, hogy akár ő vagy a kollégái is interakcióba léphetnek az adott tárggyal. Ugyanezt lehet alkalmazni egy-egy javítási munkálat során is, ahol a javítandó részeket vizualizálhatják a megfelelő 3D regisztráció után a környezetbe illesztve. Ezzel lehetővé válik olyan alkatrészek beépítése is akár, amellyel a megbízott karbantartó még nem dolgozott előtte. Amint az a 6. ábrán is látszik, a bal alsó sarokban elhelyezett jelölő alapján a 3D regisztrációt követően vizualizálni lehet beépítendő elemek pontos helyét, illetve kapcsolódási pontjait a már meglévő részekhez.



6. ábra – Szerelési útmutató kicsit másképp [10]

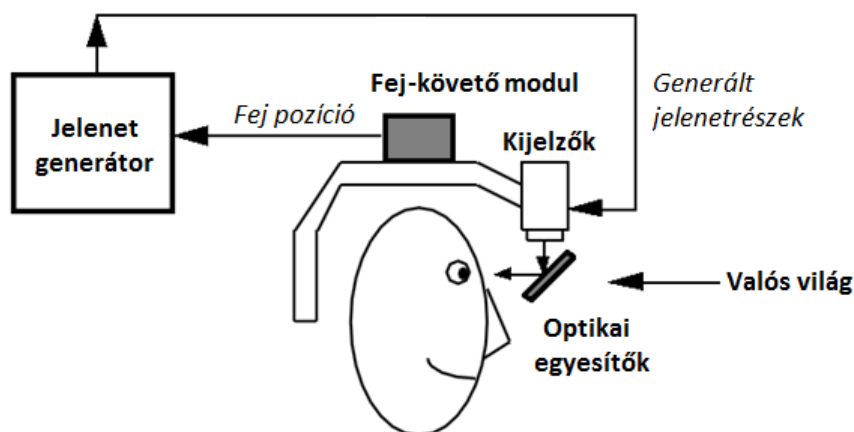
3 Tervezési szempontok

Az első szempont, amelyet meg kell fontolni, hogy hogyan keverjük a virtuális és a valós világ elemeit. Erre léteznek fejre rögzíthető kijelzők (HMD), melyek vagy átengedik a valós világból a szemünkbe jutó fényt és egy félig fényáteresztő kijelzőre vetítik a virtuálisan megjelenítendő részeket, vagy a valódi világ részeit is digitalizált formában továbbítják az ember felé [1].

3.1 Jelenetgenerálás

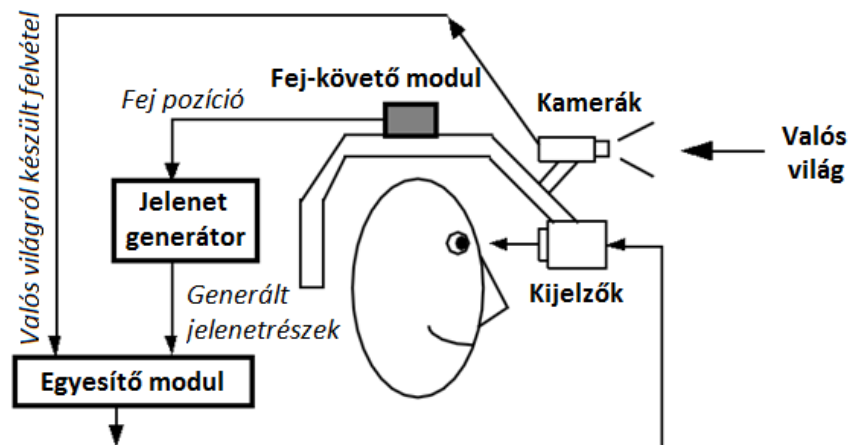
A két világ összemosásának egyik útja az úgynevezett optikai kombináció [2]. Az ilyen elven működő kijelzőket régóta használják a hadseregben a pilótáknál. Egy félig ezüstözött fényáteresztő tükrore használják, amely a fény egy részét átengedi a valós világból, de a rá irányított projektor fénysugarait is visszaveri. Ezt a 7. ábra szemlélteti.

A komolyabb eszközökön a fényt hullámhossz szerint is lehet akár szűrni. Ennek a megközelítésnek az előnye, hogy egyszerűbb konstrukció, mint a digitális és ennél fogva olcsóbb is. A késleltetése csupán néhány nanoszekundum és csak egyetlen videófolyamra kell koncentrálni. A felbontás tekintetében kétség sem férhet ahhoz, hogy a mai kijelzők még nem igazán tudják felülmúlni az emberi fovea felbontóképességét, így e tekintetben is ez jobb megoldás a digitálisnál. Ha a biztonsági szempontokat vizsgáljuk, a tápellátás megszűnése esetén az optikai megoldás használható marad, míg a digitális esetben a felhasználó elveszíti a vizuális kapcsolatot a külvilággal. Végül az optikai esetben nem lép fel eltolódás a kijelző és a megfigyelő eszköz orientációja között. Ezt különféle tükrök segítségével érik el, amely kissé megbonyolítja az ilyen típusú eszközök tervezését.



7. ábra – Az optikai jelenetvegyítés sémája [11]

Egy másik módszer, amikor a felhasználó teljes egészében digitális képet lát [2]. Ezt egy vagy két kamerával rögzítik és kiegészítik a szükséges generált részekkel, majd a kijelzőre továbbítják. Ezt a 8. ábra szemlélteti. Az összemosást többféleképpen is meg lehet valósítani. Egy lehetséges megoldás, ha olyan színű homogén háttér előtt rögzítjük a jelenetet, amelyet a fő téma nem tartalmaz. Ezután a háttérszínű pixeleket ki kell szűrni és a másik jelenet megfelelő részeit pedig megtartani, így a két jelenet szuperpozícióját lehet ezzel elérni. Egy kifinomultabb technika, amikor rendelkezünk mélységinformációval.



8. ábra – A digitális jelenetvegyítés sémája [11]

Ezzel elérhető, hogy virtuális objektumok eltakarjanak valós részeket és fordítva. Ez az egyik fő előnye az optikai megoldással szemben. További előnyt biztosíthatnak a különféle kamerára szerelt optikák, amelyekkel szélesebb látóteret biztosíthatunk a felhasználónak. Az optikai megoldásnál előforduló késleltetések közti különbségek itt kezelhetővé válnak, a két videófolyamot össze lehet hangolni. Mivel az optikai megoldás esetén csak fejre rögzíthető eszközbe épített szenzorok adataira támaszkodva lehet következtetni a megfigyelő orientációjára, ezért az a megoldás nyílt rendszernek tekinthető. A digitális esetben felhasználhatunk más regisztrációs stratégiákat – például jelölőket – annak eldöntésére, hogy milyen fej megfigyelő helyzete a világban.

3.2 Mobilitás

A második tervezési szempont a portabilitás. Az esetek többségében felmerül az igény, hogy a felhasználó mozgás közben használja az eszközt, ezért a képfeldolgozás gyorsasága és az eszköz mobilitása is meghatározó tényező [1].

3.3 3D regisztráció

A harmadik tényező a 3D regisztráció, amely a leginkább korlátozza a kiterjesztett valóság alapú megoldások hatékonyságát. A valós és virtuális tér világ elemeit úgy kellene összerendelni, hogy azok együtt egy konzisztens, hibrid egységet alkossanak. Rengeteg területen felmerül az igény erre, gondoljunk például az orvosi beavatkozásokat segítő alkalmazásokra, ahol pontosan kell tudni, hogy hol és milyen irányban kell a bőr alá hatolnia az orvosnak a különféle eszközökkel. Vizsgáljuk tehát meg milyen hibaforrásokkal kell számolni, ha 3D regisztrációt végre akarjuk hajtani.

Az egyik leggyakoribb az optikai torzítás, amely a felvevő kamera lencséinek illetve kijelző optikának a fizikai tulajdonsága. Ez a hiba tipikusan a nagylátószögű kameralencsék esetén fordul elő, mivel a torzítás általában valamilyen függvénye az optikai tengelytől mért távolságnak. A kép közepe ilyenkor torzítatlan, míg a szélei felé már egyre kivehetőbb a jelenség – például az egyenesek görbéké válnak. Mivel ez a fajta hiba általában könnyen észrevehető és van benne egyfajta rendszer, így könnyedén ki lehet küszöbölni. Ezt vagy további lencsék alkalmazásával oldják meg, amely plusz súlyt jelent, viszont nem terheli tovább a feldolgozó egységet; vagy digitálisan torzítják a képet különböző technikákkal.

Egy másik hibaforrás a követő és érzékelő rendszerben előforduló problémák. Ezeket a hibákat igen nehéz észlelni és mérni, mivel az előfordulásukban nincs rendszer.

Az utolsó statikus hibaforrás a megfigyelő kamera és a fényáteresztő kijelző eltérő helyzetéből adódik. Ezeket könnyedén lehet korrigálni a megfelelő kalibrálás alkalmazásával. A kalibrálási paramétereket olykor kézzel adják meg – ilyenkor elhelyeznek egy valós tárgyat és több nézőpontból megfigyelve addig hangolják az értékeket, amíg „jól nem néz ki”. Egy másik módszer lehet, ha további kamerákat helyeznek el pontosan oda, ahol a felhasználó szeme helyezkedne el és a kapott bemeneteket összehasonlítva megállapítják a szükséges kalibrálási értékeket. Végül használatos a jól bevált hagyományos kalibráció, amikor egy ismert tárgyról több irányból készítenek felvételeket, majd az azonos pontjainak az elmozdulásaiból megállapítják a szükséges paramétereket [11].

A dinamikus hibák a feldolgozási sebességből adódó késések miatt fordulhatnak elő. A rendszer teljes késleltetése a követő rendszerben való regisztrálástól a megjelenítésig eltelt időből adódik. Ez a késleltetés okozhat regisztrációs problémát, de csak akkor, ha mozgás jelentkezik a jelenetben. Könnyű elképzelni, hogy ha egy t_1 időpillanatban beérkezik egy kép miközben a megfigyelő kamera és/vagy a megfigyelt objektum (is) mozog, akkor mire a kijelzőn megjelenik a kép, az egy későbbi t_2 pillanatban lesz. Mire ez megtörténik, addigra –a további mozgás miatt – már nem a megfelelő időpillanathoz tartozó képet látja a felhasználó. Ez úgy mutatkozik a gyakorlatban, hogy a virtuális objektumok kissé le vannak maradva a kívánt helyükhöz képest. Ez meglehetősen illúzióromboló hatású lehet, mivel megfigyelő számára egyértelmű, hogy a látott jelenet két másik forrásból van összerakva, pedig az eredeti cél ennek pont az ellenkezője volt. Ennek a kiküszöbölésére négy lehetőség adódik.

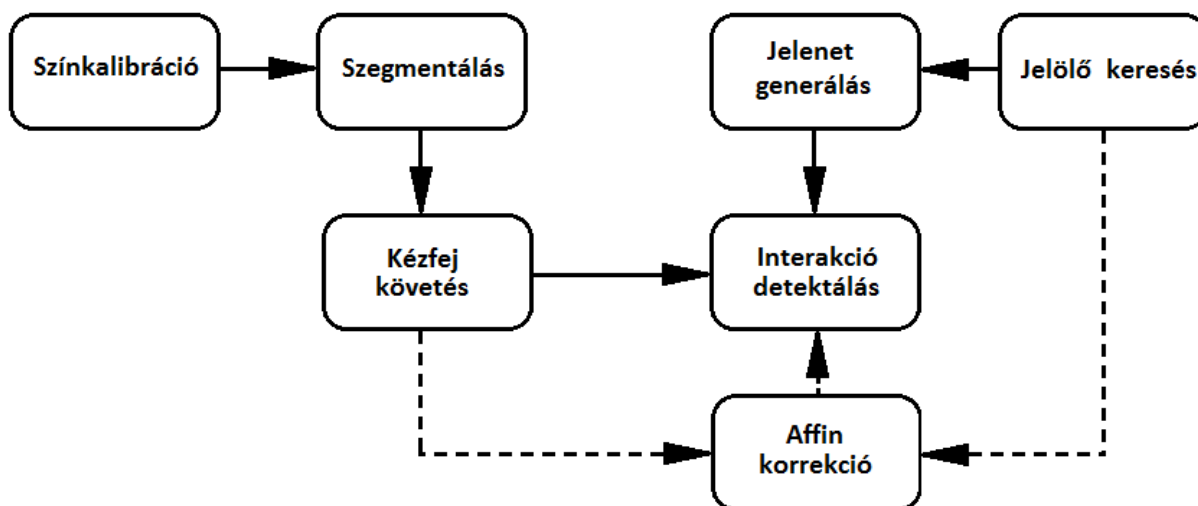
Az első és legtriviálisabb a rendszer teljes késleltetésének csökkentése. Ehhez olyan hardvert kell választani, amely inkább a sebességre, mint a minőségre van kiélezve. A feldolgozó szoftver végletekig való optimalizálásával is lehet értékes időt nyerni. Lehet trükközni a különféle affin transzformációkkal is. Ha például egy jelenet már kijelzésre kész és rendelkezésünkre áll a fej orientációja, akkor a kirajzolás előtt a legfrissebb orientációs adatok segítségével torzíthatjuk úgy a generált részeket, hogy az valóságghűbb összképet alkosson. Ehhez természetesen rendelkezni kell a megfelelő mélységinformációval minden egyes pixel esetében, illetve a transzformálást gyorsabban kell tudni elvégezni, mint az újrenderelést. Az utolsó módszer, amivel csökkenthetjük a dinamikus hibák zavaró hatásait, az a becslés. A jelenetben megfigyelhető mozgások alapján meg lehet becsülni adott részek jövőbeli pozícióját és akár orientációját. A becslést megvalósító algoritmust azonban úgy kell megválasztani, hogy az előforduló hiba minimális legyen, ugyanis meglehetősen illúzióromboló az a jelenség is, ha a generált jelenetrészek folyton „ugrálnak” a rossz becslés miatt.

4 Saját rendszer

Amint az előző fejezetekből kiderült, a kiterjesztett valóság témakörben rendkívül széleskörű kutatás folyik, hogy kiaknázzák a benne rejlő lehetőségeket. Rendszerem célja, hogy a felhasználóját tehermentesítse a különféle eszközök hordozása alól, illetve megkönnyítse az adatbevitelt például egy mobil eszközre, amelyhez a kiterjesztett valóság nyújtotta eszközöket kívánom felhasználni. Ez alatt azt kell érteni, hogy ha a felhasználó rendelkezik egy mobil, például a szemüvegére rögzíthető kivetítővel egybekötött kamerával és bizonyos mintájú jelölőkkel, ha a jelölőt bármilyen homogén színű és felületű térrészre helyezi, akkor képessé válik a teljes körű adatbevitelre, vagy kijelzésre. Ezzel lényegében bármilyen – a kritériumoknak megfelelő – felületet bemeneti, vagy akár kimeneti perifériává változtathatunk. Manapság nagy kérdés, hogy az felhasználóknak mekkora kijelzőre van szüksége a telefonjára, a laptopjára, vagy a táblagépére. Ezzel a megoldással, akár az összes felsorolt eszközt lehetne helyettesíteni és az egész működtető hardver elférne a zsebünkben.

4.1 Rendszerterv

A 9. ábrán látható a rendszer működésének elvi sémája. A rendszer, a bevezetőben említett digitális jelenetgenerálást fogja használni, ami azt jelenti, hogy valós világról készült felvételek lesznek összeszova a generált részekkel, majd e kép fog megjelenni a kijelzőn. Itt kiemelném a 3D regisztráció kapcsán felmerülő plusz lehetőségeket, amelyek a módszerrel elérhetővé válnak. Míg az optikai úton való egyesítés egy nyílt hurkú rendszernek tekinthető és a visszajelzésekről csak kevés információ áll rendelkezésre, addig a digitális úton való jelenetvegyítés egy zárt hurkú rendszerként fogható fel, amelyben több helyről is megerősítést kaphatunk a 3D regisztráció hatékonyságáról [1]. A kéz detektálása színinformáció alapján történik, míg az interakció követés bináris képfeldolgozással valósul meg. Az affin korrekció egy opcionális funkció, ami a megfigyelő szemszögéből adódó és az érintésdetektálást esetlegesen befolyásoló torzításokat hivatott kiküszöbölni.



9. ábra – A működés elvi sémája

4.2 Színkezelés

Ahhoz, hogy tetszőleges helyen használni lehessen az eszközt, szükség van a színkalibráció elvégzésére. Különböző megvilágítások – pl. természetes fény, volfrám izzó, neon izzó – mellett ugyanannak a tárgynak a színe meglehetősen eltérő lehet az érzékelés során. Ennek kiküszöbölésére

léteznek automatikus fehéregyensúly javító algoritmusok [12], melyek közül én a GrayWorld algoritmust használtam fel. Az RGB értékeket felhasználva, a 1.-4. összefüggések adják az adott pixel új R'G'B' értékeit.

$$K = \frac{R_{\text{átl}} + G_{\text{átl}} + B_{\text{átl}}}{3}$$

1. összefüggés

$$R' = R * \frac{K}{R_{\text{átl}}}$$

2. összefüggés

$$G' = G * \frac{K}{G_{\text{átl}}}$$

3. összefüggés

$$B' = B * \frac{K}{B_{\text{átl}}}$$

4. összefüggés

A bőr színének detektálására rengeteg módszert [13] lehet találni az irodalomban. A rendszer építése során fontos kritérium volt, hogy a módszer az időigényét tekintve valósidejű használatra alkalmas legyen. Három kategóriába lehet sorolni a módszereket, amelyekkel a problémát szerettem volna megoldani.

Elsőként az volt az ötlet, hogy egy ROI ablak segítségével kinyerem a kézfej egy jellemző részének a színeit, majd ezen színadatok különböző színterekben [14] ábrázolt színadatait külön átlagolva meghatározok egy referenciaértékeket, majd amely pixelek nem esnek ezen színérték egy bizonyos sugarú környezetébe, azokat háttérnek minősítem.

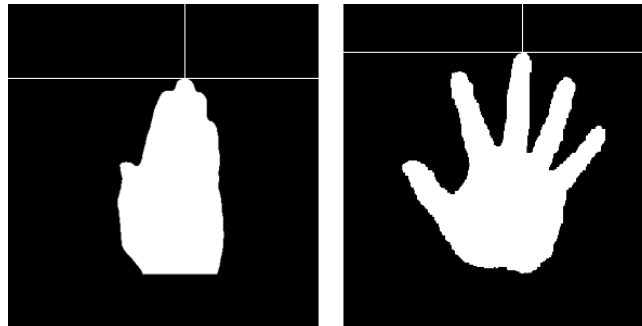
A második elgondolás az előző gondolat a továbbfejlesztett változata. Először a háttérpixelek Cb és Cr, illetve *u és *v csatornáiból vettem mintát, majd a ROI ablak segítségével a kézfej pixeleiből is építettem egy, az előzővel azonos típusú adathalmazt. Ezt követően SVM (támasz vektor gép) [15] módszerrel próbáltam elkülöníteni a háttér, illetve előtér pixeleket.

A harmadik módszer használata legegyszerűbb. Az irodalomban lehet találni empirikus úton meghatározott küszöbszámokat [13][16] az egyes színterek csatornáira vonatkozóan. Meglepőnek tűnhet, de a három megközelítés közül ez a módszer bizonyult a leghatékonyabbnak. A részletes paraméterezést a tesztelés fejezetben kerül ismertetésre.

4.3 Kézfejkövetés

Az interakció követés a szegmentált kép elemzésével történik. Mivel az ötujjas interakció követéséhez nem rendelkezünk elegendő információval, ezért korlátozni kell a használható ujjak számát egyre, vagy maximum kettőre a két kézen összesen. Ezért a felhasználótól legtávolabbi pontot – például kinyújtott ujjak esetén a középső ujj – fogjuk az interakciót végző eszköznek tekinteni. Erre láthatunk két példát a 10. ábrán, amelyen látszik, hogy bárhogy tartjuk a kezünket, mindig a

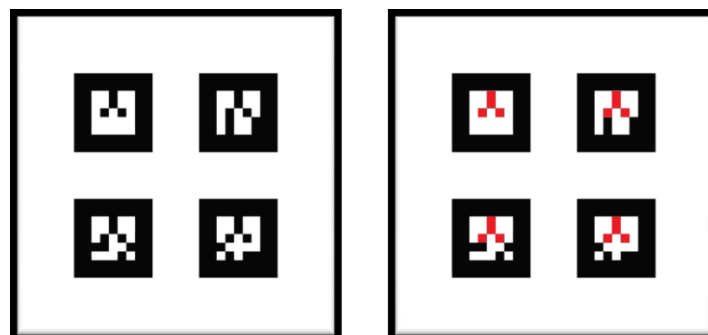
legtávolabbi pont lesz az interakciós pont. Továbbfejlesztési lehetőségként, meg lehetne határozni a szegmentált és binarizált kép nyomatéki főtengelyét és ebben az irányban keressük az objektum – itt kézfej – szélső pontját. Normál esetben a megfigyelő a saját kezét felülről látja bizonyos szögben, de az említett számítás elvégzésével az oldalról történő megfigyelés is lehetséges lenne.



10. ábra – Interakciós pont

4.4 Jelölődetektálás

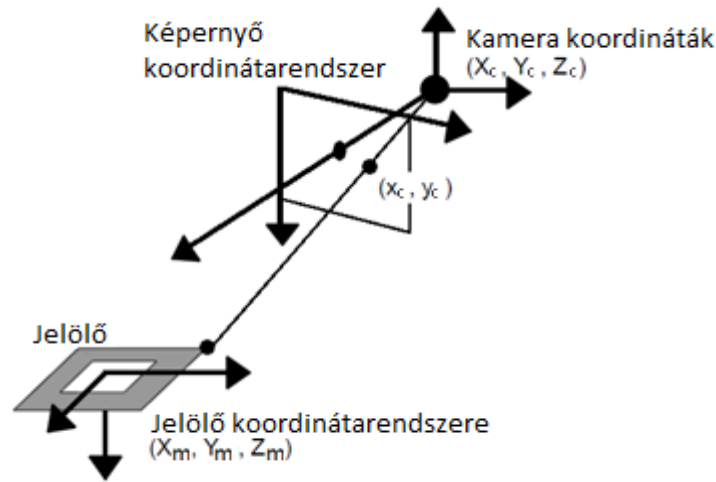
A rendszer egyik alapköve a jelölődetektálás. Ahhoz, hogy ezt hatékonyan meg lehessen valósítani, szükség lesz nulladik lépésként a kamera kalibráció elvégzésére. Ehhez segítséget nyújt az EmguCV illetve a Goblin XNA környezet is melyeket megvalósítás során felhasználtam. Mindkettő a szokványos sakktáblás módszert alkalmazza [17]. Ha rendelkezésünkre állnak a kamera torzítási paraméterei külső és belső paraméterei, akkor kezdetünk hozzá a jelölők detektálásához. A gyakorlatban egyre terjed az tulajdonság alapú 3D regisztráció [18], azonban a jelölő alapú rendszerek sokkal megbízhatóbbak. Ez utóbbi esetben a 3D regisztrációhoz egy – de inkább több – úgynevezett jelölőt használnak, amely hasonlít a jól ismert QR kódhoz [19]. Ez azt jelenti az esetemben, hogy egy fekete négyzet alakú keret fel lett osztva egyenlő négyzetes részekre és ezek egy bizonyos konvenció szerint feketére színezettek, vagy fehérek. A konvenció azért fontos, hogy meg lehessen állapítani, hogy az adott jelölőhöz képest a megfigyelő eszköz milyen orientációjú. A 11. ábrán megfigyelhető az általam alkalmazott megkötés, amely segíti a 3D regisztrációt. A bal oldali ábrán elsőre mindegyik minta véletlenszerűnek tűnhet, azonban ha tekintjük a jobboldali ábrát, akkor pirossal színezve látható, hogy mindegyik jelölő tartalmaz egy közös részmintát, amelynek segítségével egyértelműen meg lehet állapítani a jelölő irányát is.



11. ábra – Szabályszerűség a jelölők előállításában [20]

Az imént bemutatott jelölők mérete állandó és előre ismert, így a pozícióbecslésnek jó alapjául szolgálnak. A jelölő koordinátarendszere és a kamera koordinátarendszere közötti kapcsolatot a 12. ábra szemlélteti, ahol az X_c, Y_c, Z_c a kamera koordinátarendszerében vett koordinátákat, az x_c, y_c a

képernyő koordinátarendszerében a pixelkoordinátákat, az X_m, Y_m, Z_m pedig a jelölő koordinátáit jelölik a jelölő koordinátarendszerében nézve [21].



12. ábra – Szemléletes kapcsolat a koordinátarendszerek között [21]

A transzformációt az 5. összefüggés fejezi ki, ahol az X_c, Y_c, Z_c a kamera koordinátarendszerében vett koordinátákat, a $V_{3 \times 3}$ elemei a homogén transzformációs mátrix rotációs almatrixát, a $W_{3 \times 1}$ a homogén transzformációs mátrix translációs almatrixát, az X_m, Y_m, Z_m pedig a jelölő koordinátáit jelölik a jelölő koordinátarendszerében nézve. A skálázó faktort egyre választjuk, illetve a perspektivikus torzítást nem tételezünk fel.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & W_x \\ V_{21} & V_{22} & V_{23} & W_y \\ V_{31} & V_{32} & V_{33} & W_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{V}_{3 \times 3} & \mathbf{W}_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \mathbf{T}_{cm} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

5. összefüggés

A jelölők helyét egy küszöbölés után úgy kapjuk meg, hogy elvetjük a kép azon részeit, amelyeket nem lehet négy egyenessel határolni. A négy határoló egyenes paramétereit illetve a négy sarokpontot a metszéspontjaikból lehet kiszámítani. Az immár rendelkezésünkre álló régiókat ezután normalizáljuk, amely egy perspektív transzformációt jelent, majd mintaillesztéssel keresünk egyezést a rendszernek előzőleg megadott minták és ezen megtalált régiók tartalma között. Ezt a 6. összefüggés szerint tudjuk megtenni, ahol a transzformációs mátrix változói a képernyő koordinátáinak és a jelölő koordinátáinak behelyettesítéséből adódnak. Skálázó faktorként megjelenik a h , amelynek a Z koordináták esetén az egyes értéket adjuk.

$$\begin{bmatrix} hx_c \\ hy_c \\ h \end{bmatrix} = \begin{bmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ 1 \end{bmatrix}$$

6. összefüggés

Ha a jelölő négyzet két párhuzamos oldala a képre van vetítve, akkor a két egyenes egyenlete a kamera koordináta rendszerében a 7. összefüggésben leírtak szerint néz ki.

$$a_1x + b_1y + c_1 = 0, \quad a_2x + b_2y + c_2 = 0$$

7. összefüggés

Ezeket az értékeket minden egyes jelölőre kiszámítjuk. A 8. összefüggésben látható a P perspektív projekciós mátrix, amelyet a kamerakalibrálás során határoztunk meg, illetve azon síkok egyenlete, amelyek tartalmazzák ezt a két oldalt. Ezen síkok kifejezhetőek a kamera koordináta rendszerében is behelyettesítve az x_c és az y_c értékeket, így kapva az 8. összefüggést.

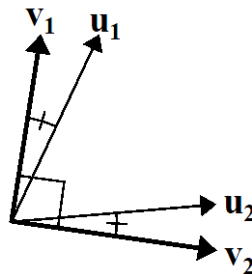
$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & 0 \\ 0 & P_{22} & P_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

$$a_1P_{11}X_c + (a_1P_{12} + b_1P_{22})Y_c + (a_1P_{13} + b_1P_{23} + c_1)Z_c = 0$$

$$a_2P_{11}X_c + (a_2P_{12} + b_2P_{22})Y_c + (a_2P_{13} + b_2P_{23} + c_2)Z_c = 0$$

8. összefüggés

Mivel a síkok n_1 és n_2 normálvektorait ismerjük, ezért a két párhuzamos oldal irányvektorát is meghatározhatjuk a két normális keresztszorzataként. Továbbá ismert a két vektor egymásra elméletben merőleges irányvektor u_1 és u_2 , amelyet a két párhuzamos oldalpárból nyertünk. Mivel a gyakorlatban képfeldolgozás során keletkező hibák miatt ezek a vektorok szinte sosem pontosan merőlegesek egymásra, ezért bevezetünk egy v_1 és egy v_2 vektort rendre az u_1 és az u_2 síkjában, ahogy az a 13. ábrán látható. Ha megadjuk a hozzá a v_3 vektort is, amelye egyaránt merőleges a v_1 -re és a v_2 -re, akkor megkapjuk a $V_{3 \times 3}$ rotációs komponenst.

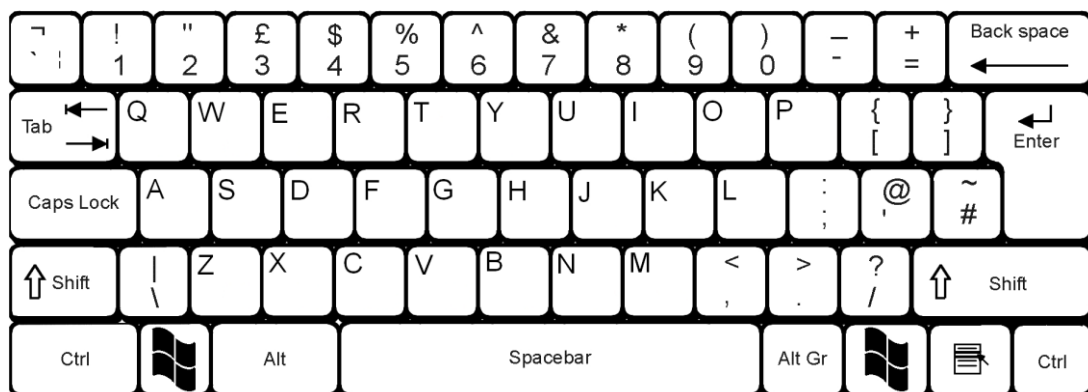


13. ábra – Az u és v vektorok viszonya

Ha a $V_{3 \times 3}$ -t ismerjük, akkor a jelölő négy sarokkoordinátáját a jelölő koordinátarendszerében és képernyő koordinátarendszerében felírva kapunk nyolc egyenletet, amelyekből a W_x , W_y és W_z komponensek kiszámíthatóak.

4.5 Jelenetgenerálás

Ha a jelölőt megtalálta a rendszer, akkor ismerjük az orientációját és a pozícióját, amelyet a billentyűzettérkép kivetítésére tudunk felhasználni. Ezt úgy tesszük, hogy például a billentyűzettérkép jobb felső sarkát a megtalált jelölő bal alsó sarkához igazítjuk. Ez azért fontos, hogy bevitel közben a felhasználó ne takarja el a kezével a jelölőt, különben eltűnik a billentyűzettérkép. A 14. ábrán egy lehetséges billentyűtérképet láthatunk, azonban ez csak egy példa, mivel térképet tetszés szerint lehet változtatni.



14. ábra – Angol kiosztású billentyűtérkép

4.6 Interakció detektálás

Mivel az interakciós pont már ismert, a billentyűtérkép struktúrája szintén ismert és a jelölő biztos pontjai is ismertek, így minden adott, hogy az interakció megvalósuljon. Ezt úgy lehet elérni, ha például a jelölő bal alsó sarkához viszonyítva megvizsgáljuk az interakciós pont elmozdulását és ezt levetítjük a billentyűtérképre. Ekkor több lehetőség adódik az „érintés” meghatározására. Az egyik módszer, ha egy kis ideig – például három-négy képkockán keresztül – a kívánt helyen tartjuk az ujjunkat és ezt tekintjük érintésnek. Ez ugyan lassúnak tűnhet, viszont így kisebb a tévesztés esélye, illetve a felhasználó testre is szabhatja a várakozás idejét. A gyorsabb módszer, ha az egyik ujjunk ide-oda húzogatásával a megtett útvonal töréseit, görbületeit elemezzük.

4.7 Affin korrekció

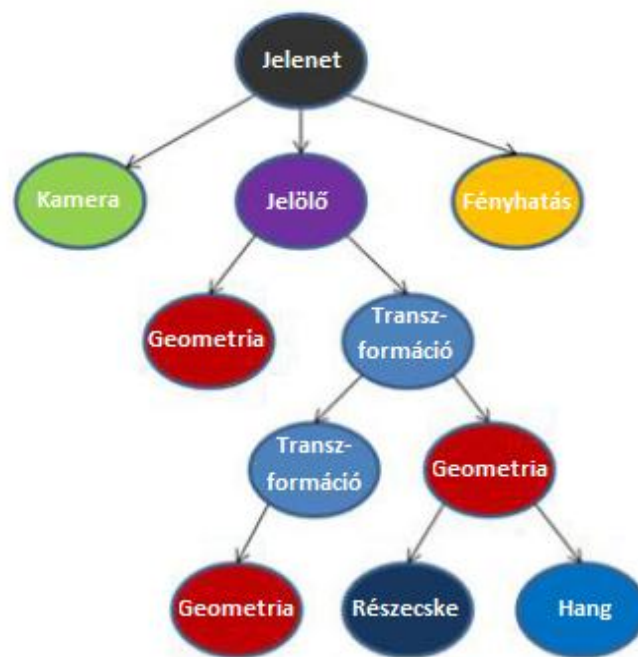
Az affin korrekció egy opcionális beállítás a rendszerben. Ha például a nézőpont miatt a kivetített térkép kezdene zavaró mértékben torzulni, akkor a kiszegmentált kézfejet is lehetne affin transzformációval a jelölő torzulásából meghatározott módon módosítani.

5 Összegzés

5.1 Megvalósítás

A megvalósítás során két keretrendszert használtam fel – ezek a Goblin XNA [22] és az EmguCV. A továbbiakban e két rendszerről lesz röviden szó

A Goblin XNA – egy szabad felhasználású gyűjtemény, amely segítségével 3D interfészeket lehet megvalósítani kiterjesztett illetve virtuális valóság technikák felhasználásával egyaránt. E rendszer részeit használtam fel a jelölőkeresés és a jelenetgenerálás feladatok megvalósítására. Egy jelenet felépítése hierarchikus módon történik, amelyet egy fa struktúrán keresztül biztosítanak, amit preorder bejárás szerint értékelnek ki. Egy példa erre a 15. ábrán látható.



15. ábra – Hierarchikus jelenetépítés a Goblin XNA keretrendszerben [22]

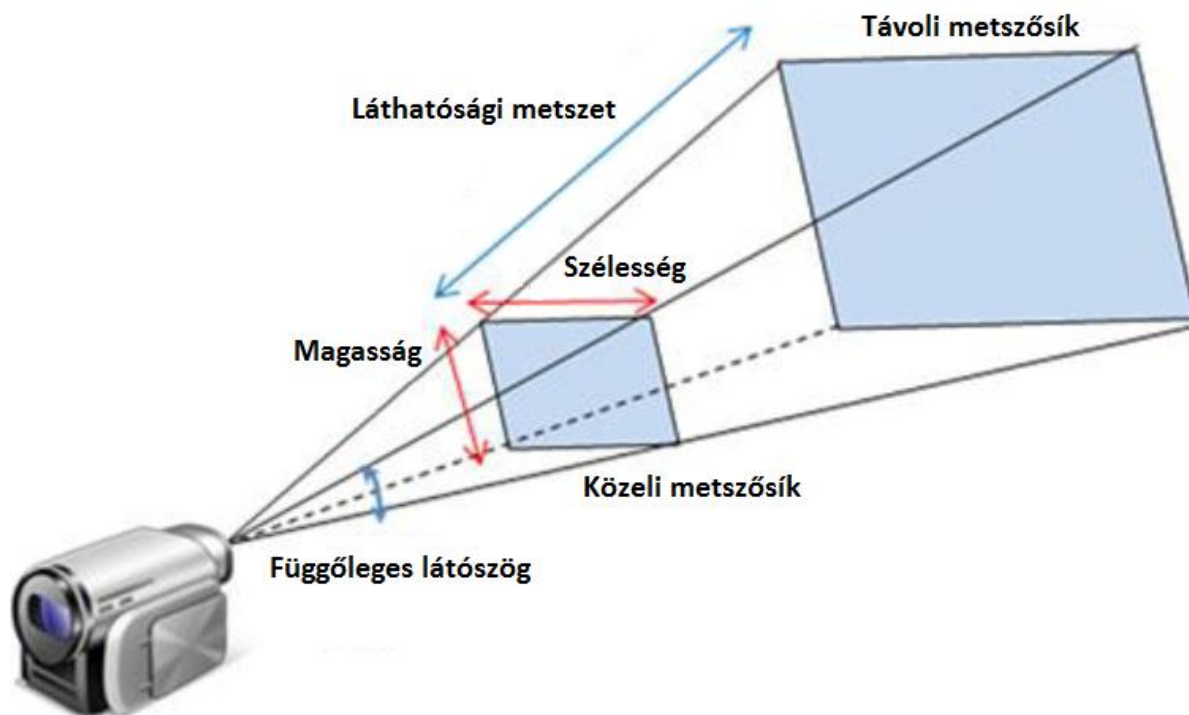
Alapvetően tíz féle csomópont típust lehet megvalósítani, amelyek a következők: geometria, transzformáció, fényhatás, kamera, részecske, jelölő, hang, választó, felbontás, követő. Azonban megkötések is definiáltak, amelynek értelmében csak a geometria, a transzformáció, a jelölő, a választó, a felbontás és a követő típusú csomópontoknak lehetnek leszármazottaik. Nézzük sorban, mit is jelentenek az egyes típusok.

A geometria típusú csomópontok tartalmazzák a generált jelenetrészek 3D modelljét. Egy ilyen típusú csomópontozhoz mindig csak egyetlen egy modell állományt lehet társítani. Ez azt jelenti, hogy ha több dolgot szeretnénk megjeleníteni, akkor több ilyen csomópontot kell a fa struktúrába illeszteni. Vigyázni kell azonban, mert ha például egy transzformációs csomópontból ágazik le egy geometriai csomópont, akkor a transzformáció hatására módosulhat, illetve a további csomópontok is tranzitíven módosulnak ezen az ágon. Ezzel a típussal például a gombokat valósíthatjuk meg.

A transzformációs csomópontok tudják módosítani a hierarchiában alattuk elhelyezkedő olyan csomópontokat, amelyek tartalmazznak bármilyen térbeli információt. A transzformáció tulajdonságait vagy megadjuk egyenként (eltolás, skálázás, forgatás), vagy leírjuk a konkrét transzformációs mátrixot. Ezzel a csomóponttal lehet animálni a gombokat.

A fényhatás típusú csomópont segítségével be lehet állítani, a fényforrás helyét és szögét, amely megvilágítja az adott objektumot. Mivel a keretrendszernek ezen része a jelen projekt funkcionalitásához nem kapcsolódik, ezért a részleteket a dolgozat nem tárgyalja.

A kamera típusú csomópont definiálja a kijelzőn látható részeket. Lényegében ezt egy gúlával lehet szemléltetni, ahogy a 16. ábrán is látható. A gúlát a függőleges látószög, a képarány, a közeli metszősík és a távoli metszősík határozza meg.



16. ábra – Látható tartomány meghatározása [22]

A részecske típusú csomópontok különféle effektek – például tűz, füst, robbanás, fröccsenés – megjelenítésére valók, hasonlóan a fényhatás típusú csomópontokhoz. A jelölő típusú csomópontok képesek módosítani a leszármazott csomópontjaik transzformációit, hasonlóan a transzformációs csomópontokhoz. Ez egy hat szabadságfokú helyzeti és orientációs mátrixon alapul, amelyet egy vagy több jelölőre számítanak ki. Egy-egy jelölő lényegében egy speciális geometriai alakzat, amelyen gyakran papírra nyomtatnak és ezeket a mintákat keresik a korábban ismertetett eljárás segítségével, hogy megállapítsák a pozícióját és az orientációját a kamerához képest. A Goblin XNA jelenleg az ALVAR jelölő-követő módszerét használja, amely egyezik a korábban ismertetett módszerrel. Ehhez létezik egy generátor állomány, amely előállít a közel 65000 féle lehetőség közül annyit, amennyire éppen szükség van és ezeket egyedi azonosítóval látja el. Ezt követően a rendszer az azonosítók alapján fogja elfogadni, vagy elvetni a megtalált vagy vélt jelölőket.

A különféle hanghatásokért felelős hang típusú csomópontot egyfajta visszacsatolásként lehetne beilleszteni a tárgyalt rendszerbe. Például, ha felhasználó megnyom egy gombot, akkor erről – a vizuális változáson kívül – egy rövid visszajelző hang formájában tud értesülni. Ez azért hasznos, mert gyakorlatilag nem a felületek fizikai érintésével kap visszajelzést a felhasználó, hanem a jelenben szereplő részek megfelelő fedésbe hozásával, amely nem feltétlen jár fizikai kontaktussal.

A kapcsoló típusú csomópont a jelenetépítés szempontjából fontos fabejárási sorrendet képes módosítani.

A felbontás típusú csomóponttal egy modellekből álló listából lehet meghatározni azt, hogy melyik modell kerüljön megjelenítésre. Ez igazából a geometria típusú csomópont kiterjesztése, a különbség csupán a listás megoldás. A listában elfoglalt hely determinálja a felbontás minőségét, tehát minél előrébb helyezkedik el egy modell a listában, annál élesebb lesz a jelenet generálásakor.

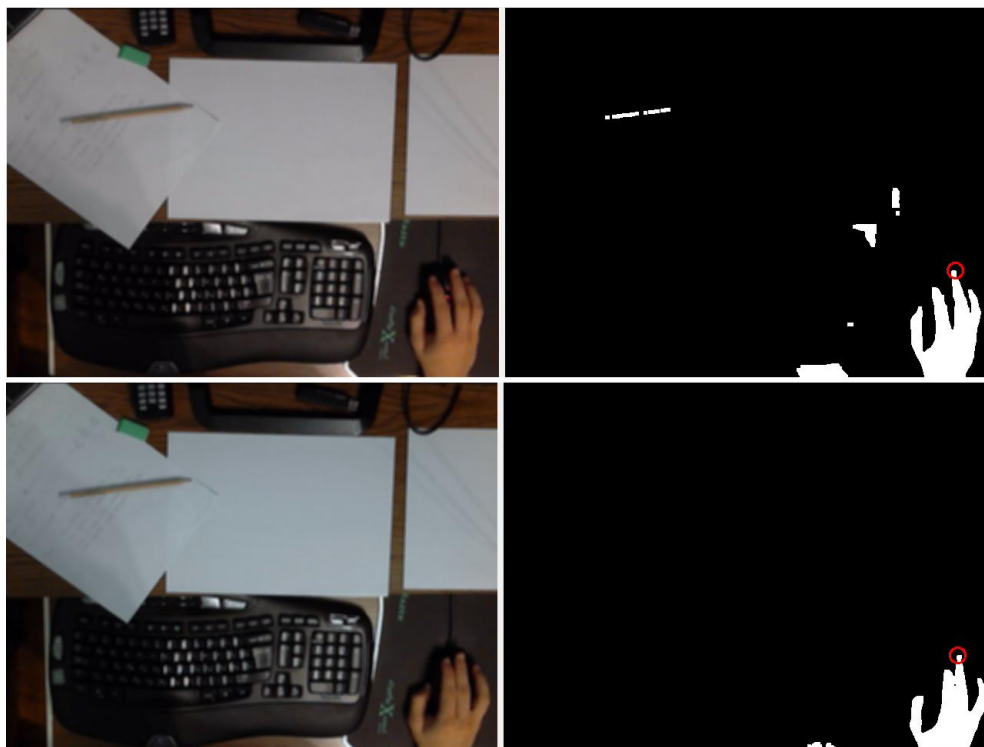
Végül tekintsük a követő típusú csomópontot, amely nagyon hasonló a jelölő típusú csomópontéhoz. Az egyetlen paraméter, amit meg kell adni, az az eszköz azonosítója, amely meghatározza a hat szabadságfokú követő típusát. Ha ez megvan, akkor a koordináta transzformációk automatikusan működésbe lépnek és módosítják a leszármazott csomópontok tulajdonságait.

A színkezelést, az interakciós pont meghatározását és követését, illetve az affin korrekciót megvalósító alrendszerek megvalósítását az OpenCV közvetett felhasználását biztosító EmguCV wrappert használtam fel, melynek segítségével elérhetővé válnak C# nyelven a C++-ban hatékonyan implementált képfeldolgozó metódusok.

5.2 Tesztelés

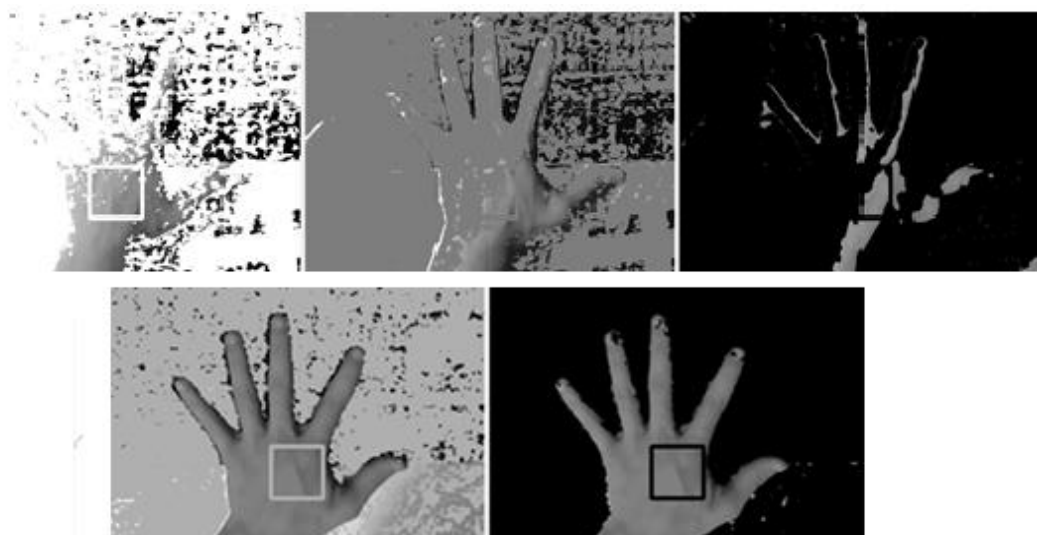
Mivel a Goblin XNA megvalósít egy olyan funkciót, amely a képernyő sík egy pontját – például egérgattintás – képes levetíteni a jelölő síkjába és képes megállapítani, hogy az eltalált-e egy objektumot, így a teszteléskor csak a kéz szegmentálására fektettem hangsúlyt. A megoldandó feladat tehát az volt, hogy a kéz egy jellemző pontját – itt leghosszabb ujj vége, interakciós pont – kellett tudni meghatározni, illetve egy bizonyos ideig egy meghatározott pozícióban tartani. A detektált interakciós pontot piros kör jelzi. A leütési szándékot akkor detektálom, ha a kör középpontja ötnél több egymást követő képkockán keresztül nem mozdult az euklideszi távolság szerint három egységnél távolabbra.

A színkorrekció GrayWorld algoritmus által elért eredmény a 17. ábrán látható. A felvétel készítésekor a megvilágítást meleg fehér színárnyalatú tartós izzó biztosította, így a bal felső képen jól látható – főként a fehér papíron – a színek enyhe torzulása a vöröses-sárgás tartomány felé. Amint az megfigyelhető, ennek hatására a színszegmentálás elvégzése után a ceruza, illetve a fa asztal egy része is bőr tónusú objektumpixelként detektált, amely adott esetben elronthatja az interakciós pont érzékelését. A bal alsó, korrigált képen – összevetve a bal felsővel – a papírlap színe ezúttal a hidegebb, kékes tartomány felé torzult el, azonban ez már jóval közelebb áll a valósághoz. Erre a bizonyíték rögtön a jobb alsó képen látszik, ahol a teljes ceruza és szinte az összes asztalrészlet eddig objektumként detektált része eltűnt, viszont a kéz megmaradt. A továbbiakban, ahol nincs külön említés a GrayWorld-ről, ott az előfeldolgozás részeként felhasználtam azt.



17. ábra – GrayWorld színekorrektció

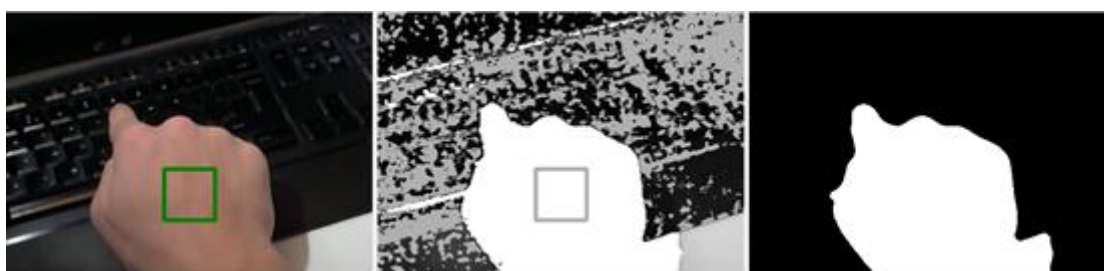
A szegmentálás során háromféle megközelítést alkalmaztam. Az első módszer során a referenciaszín meghatározásához több olyan színteret teszteltem, ahol a luminancia komponens elkülönül a színinformációtól. Ezzel kívántam elérni, hogy a megvilágítás minél kevésbé befolyásolja az eredményt. A 18. ábrán a HSV, az Lab, az Xyz, az YCbCr és az Luv színterekkel elért eredmények láthatóak. Az első három színtér eredményei a felső sorban követhetők, melyekkel láthatóan nem sikerült érdemi eredményt elérni. Azonban az YCbCr és a Luv viszonylag sikeres szegmentációt szolgáltatott.



18. ábra – Különböző színterekben végzett kísérletek eredményei. A sorrend fentről jobbra haladva: HSV, Lab, Xyz.
Alsó sor: YCbCr, Luv

A szűrés a ROI ablakban meghatározott színter komponensek átlagát veszi referenciaként, majd ennek alapján hajtódik végre a szűrés az egész képen. A szűrést több tolerancia küszöbvel is elvégeztem, de

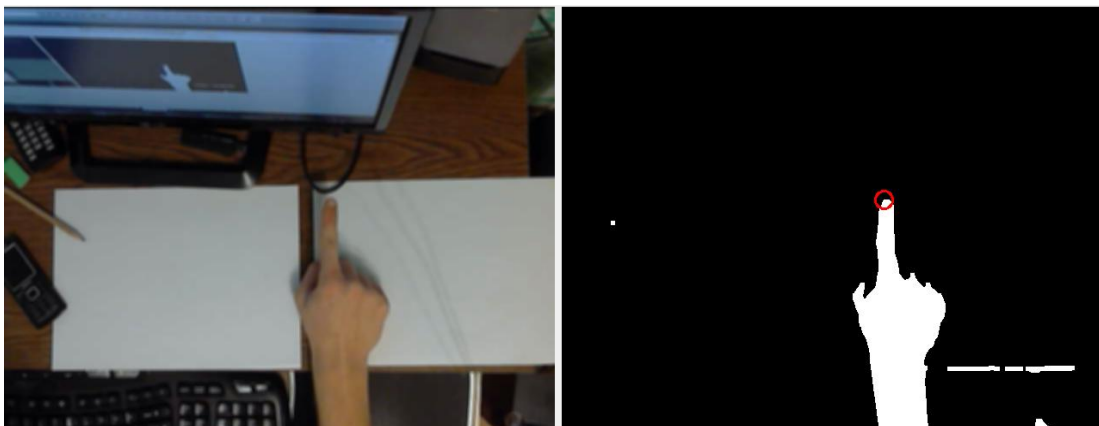
az eredményen ez érdemben nem változtatott. A HSV színtér esetén – mivel itt csak egy csatorna tárol szín információt – a tolerancia egy szám volt a 0-30 intervallumban, míg a többi színtér esetén ezt euklideszi távolsággal definiáltam szintén ezen az intervallumon, mivel itt több komponens tárolja a színinformációt. A 19. ábrán látható, a maszk kialakításának folyamata. Előfeldolgozásként mediánszűrést használtam 3-as ablakmérettel, a szegmentálási tolerancia mindkét színtér esetén 20-20 volt, a binarizálást Otsu módszerrel végeztem, végül utőfeldolgozásként pedig szintén mediánszűrést használtam 13-as ablakmérettel. Érdekességgé felmerült, hogy egyes színterek (például YCbCr) bizonyos színeket (tipikusan kék és zöld) jobban detektáltak, mint másokat (például piros). A jelenség nyitja valószínűleg a színtér felépítéséből adódik. Ezt a tulajdonságot egyszínű kesztyűvel ki lehetne használni, de nélküle is működött a rendszer. További tapasztalatként megemlítem, hogy a megvilágítás enyhébb változásaira is igen érzékeny volt ez a módszer, mivel a referenciaszín egy adott megvilágításhoz készül. Ezen a GrayWorld sem tudott érdemben segíteni.



19. ábra – Eredeti kép, szegmentált kép, binarizált kép

Az előző ötletből következett az a gondolat, hogy sokkal robusztusabbá lehetne tenni a szegmentálást, ha a színinformációk által meghatározott térben egy hipersíkkal választanám el a háttérret az objektumtól. Mintát vettem tehát a háttérből – kéz nem volt a képen – illetve a ROI ablak segítségével a kézfejről, érdemi eredményt azonban nem sikerült elérni. Érdekes módon a tanítóminták száma nem befolyásolta számottevően a betanítási folyamatot, az mindig lefutott 1-3 másodperc alatt. Ám a betanított rendszer predikciós sebessége közel sem volt valós idejű alkalmazásra használható. A 640×480 felbontású kép teljes osztályozása több, mint tíz percig tartott és ennek fejében még értékelhetetlen eredményt is szolgáltatott. Rengeteg paraméterezést kipróbálva, jelentősen csökkentett méretű tesztképen sem produkált használható eredményt, így ezt a módszert elvettem. A tanítás feljavítására esetleg egy igen nagyméretű címkézett adatbázist lehetne felhasználni, amelyben a címkék a képen előforduló bőr objektumrészeket jelölik. Ezzel előre tanítva a rendszert talán jobb eredmény érhető el, azonban nekem nem állt ilyen adatbázis a rendelkezésemre.

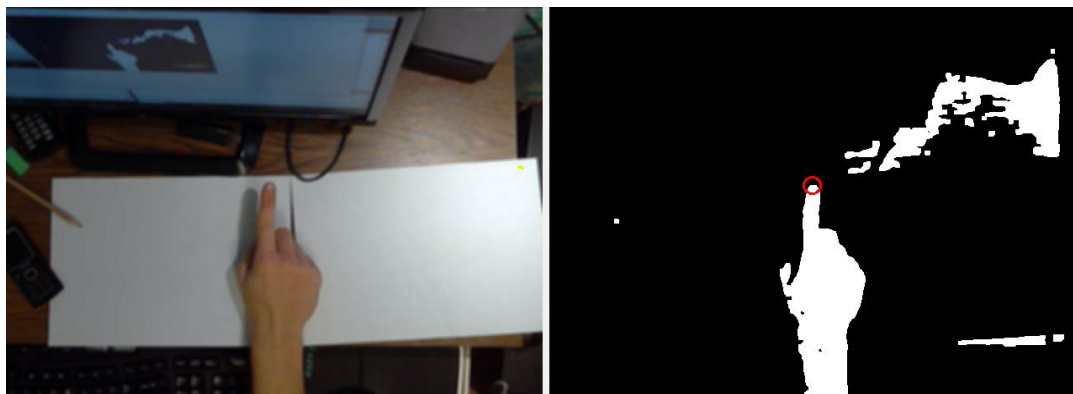
Harmadik módszerként az empirikus úton meghatározott mérőszámokat próbáltam ki. Az értékek minden esetben byte adatformátumhoz lettek igazítva. A HSV színtérre vonatkozóan a $0 \leq H \leq 50$, a $23 \leq S \leq 68$ és a $89 \leq V \leq 229$ paraméterezést javasol az irodalom. Az szórt fényű megvilágításnál készült teszteredmény a 20. ábrán látható.



20. ábra – HSV szegmentálás szórt fénynél

Bár maradtak nem kívánatos részek a képen, azonban jól azonosítható marad az interakciós pont. Ezzel a módszerrel szórt fényű megvilágítás mellett az interakciós pont stabilan 3 pixeles sugarú tartományban tartható, amely alkalmassá teszi a leütési szándék azonosítását.

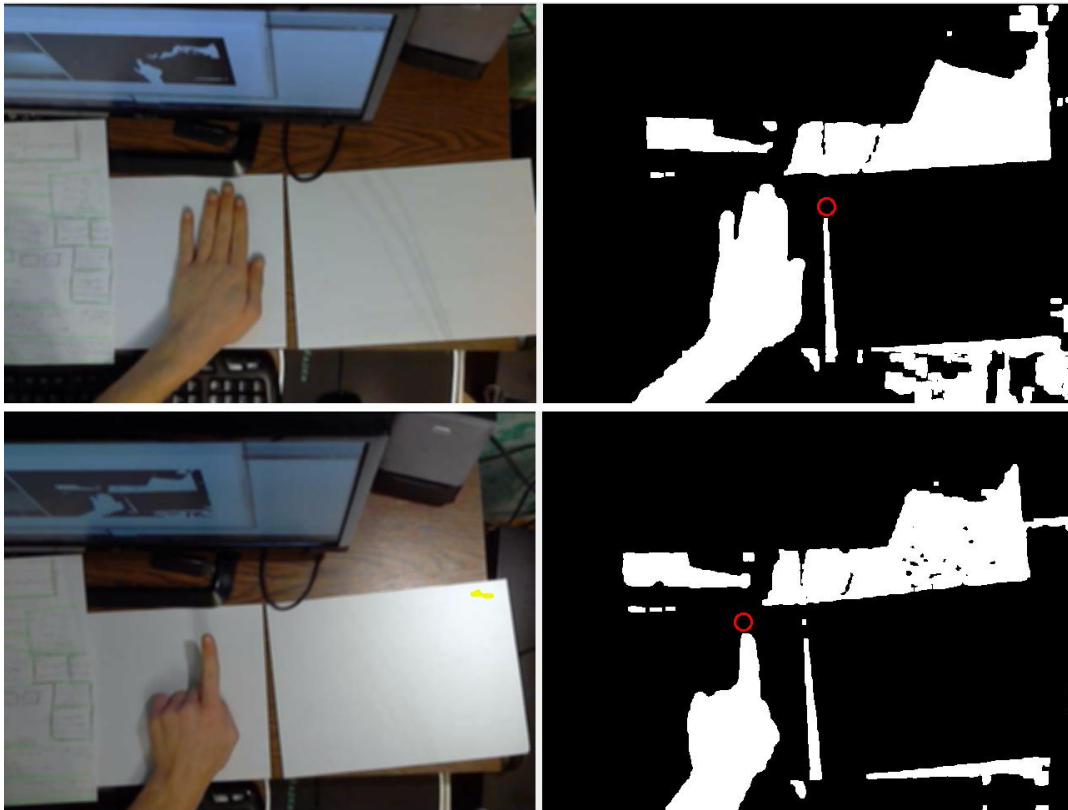
Elvégeztem a tesztet direkt megvilágítás esetére is, amelynek az eredményét a 21. ábra szemlélteti.



21. ábra – HSV szegmentálás direkt megvilágítással

Az bal oldali képen jól látszik, hogy a lámpából jövő fény az asztal jobb oldalán koncentrálódik. Ez sajnálatos módon előidéz nemkívánatos részeket a bináris maszkon, amely így olykor hibás interakciós pontot detektál. Ami ennél is nagyobb probléma, hogy a jól detektált interakciós pont nem marad a 3 pixeles tűréshatáron belül, így alkalmatlan a leütési szándék detektálására. Ez a kamera által a gyenge fényviszonyok miatt felvett zajnak tudható be.

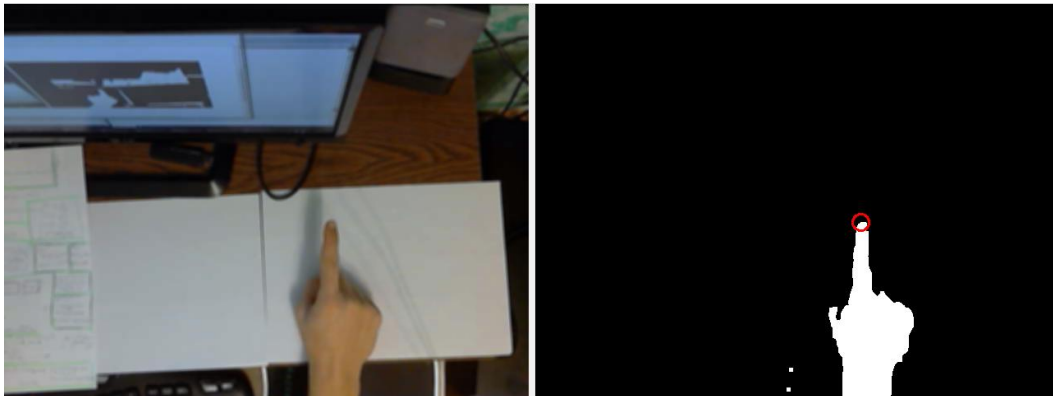
További színterekre is léteznek empirikus küszöbszámok, például az YCbCr-re. Itt a $77 \leq Cb \leq 127$ és a $133 \leq Cr \leq 173$ szabályokat alkalmaztam, amelynek az eredménye a 22. ábrán látható.



22. ábra – YCbCr szegmentálás szórt (felső sor) és direkt (alsó sor) megvilágítás esetén

Érdekes összevetni a szórt és a direkt megvilágításnál kapott maszkokat egymással, illetve a HSV színtér esetén kapott maszkokkal. Két fontos dolog figyelhető meg ennél a tesztnél. Az egyik, hogy ebben a színtérben ezekkel a paraméterekkel nem lehet megkülönböztetni a bőr és a fa színeit, amely így hibás maszkot és rossz interakciós pontot eredményez. A másik, hogy ebben az esetben jobb eredményt kapunk direkt megvilágítással, mint a szórt fénnel. Jól látható a két bináris képen, hogy direkt megvilágítással kevesebb fals pozitív objektumpixelt kapunk, amely ellentmond a HSV esetén tapasztaltakkal. Az eredmények igen tanulságosak, azonban az alkalmazás szempontjából ez a módszer nem használható.

A következő próba a hagyományos RGB színtér használatával történt. Az javasolt paraméterezések $R > 95$, $G > 40$, $B > 20$, $\max(R, G, B) - \min(R, G, B) > 15$, $|R - G| > 15$, $R > G$, $R > B$ használatával kapott eredmény szórt megvilágítás mellett a 23. ábrán látható.



23. ábra – RGB szegmentálás szórt fénynél

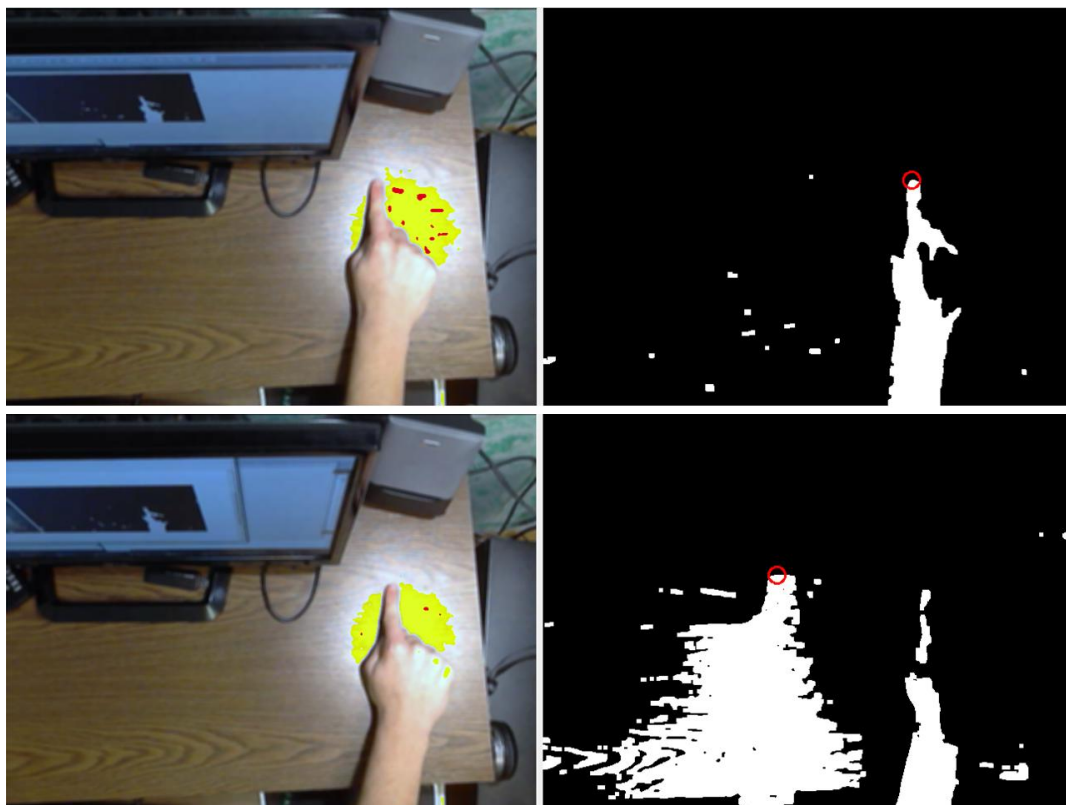
Ezzel a módszerrel szinte tökéletesen el lehetett különíteni a fa és a bőr részeket, amelyeket az előző két színtér esetén nem sikerült.



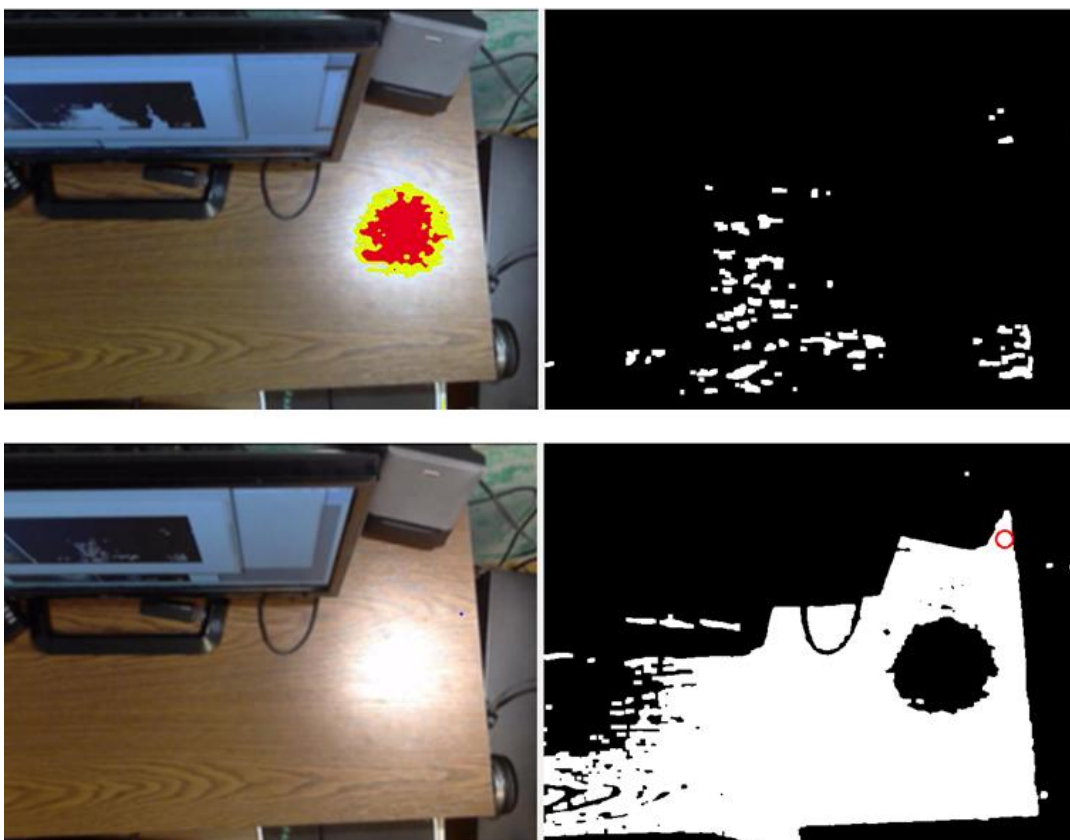
24. ábra – RGB szegmentálás direkt megvilágítással

Direkt megvilágítás akadnak fals pozitív részek, azonban az interakciós pont mindkét fényviszony esetén stabilan benntartható a 3 pixeles tűrési zónában, így alkalmas arra, hogy a leütési szándékot detektáljuk.

Végző összehasonlításként kipróbáltam a HSV és az RGB módszert a fehér háttér nélkül is (25. ábra), valamint a GrayWorld algoritmust az RGB szegmentálásra szintén fehér háttér nélkül (26. ábra). A bal oldali képeken a sárga és piros foltok GrayWorld algoritmus implementációk különbségeiből adódnak, azonban a további feldolgozást nem befolyásolják hátrányosan, ahogy az a jobboldali képekből is látszik.



25. ábra – RGB (felső sor) és HSV (alsó sor) szegmentálás direkt megvilágítás esetén komplex háttérrel



26. ábra – GrayWorld + RGB szegmentálás komplex háttérrel (felső sor),
GrayWorld nélküli RGB szegmentálás (alsó sor)

Összegzésként elmondható, hogy az eredmények változatosak, illetve meglepőek. A GrayWorld módszerről bebizonyosodott, hogy igen hasznos az alkalmazása az előfeldolgozás során, különösen, ha a színinformáció a későbbiekben kritikus. A szegmentálási módszerek vizsgálata során a referenciaszínnel operáló megoldás szűkös keretek között, de működött, viszont nem hozta a kívánt eredményt. Az SVM osztályzó „kudarcat” a hiányos tanításnak lehet betudni, azonban kiváló tanítás mellett is sokkal gyorsabb predikcióra lenne szükség. Az empirikus küszöbök alkalmazása esetén egy olyan színtérrel szemben, amely külön csatornán tárolja a megvilágítás komponensét, az elvárás az lenne, hogy a megvilágítás változásaitól valamelyest függetlenül hasonló eredményt adjon a szegmentálás során. A HSV-ről és az YCbCr-ről azonban bebizonyosodott, hogy valós idejű detektálásra nemigen alkalmas, azonban pozitív csalódás az RGB módszer meglepő hatékonysága a fényviszonyok és háttér változása ellenére is. Ennek a magyarázata az lehet, hogy ez utóbbi esetben több paraméter alapján hoztunk döntést, s így biztosabban el lehetett különíteni a bőr színeit a az RGB színtérben.

A felvetett problémát tehát sikerült megoldani a GrayWorld algoritmus és az RGB szegmentáló együttes alkalmazásával. A maszkok fals pozitív találatait könnyedén ki lehet szűrni, ha egy előzetes háttérvizsgálat során regisztráljuk a kritikus régiókat, illetve ha a momentumok alapján osztályozzuk a régiókat.

Továbbfejlesztésként megfontolandó a kevert Gauss modellek módszere [13], amelyet szintén gyakran használnak az irodalom szerint bőrszín detektálásra és igen jó eredményeket [23] lehet elérni vele offline feldolgozásban. Valós idejű alkalmazásáról, illetve a számítási időigényéről jelenleg nem áll rendelkezésemre információ.

5.3 Összevetés az eddigi technikákkal, továbbfejlesztés

A dolgozat egy meglehetősen jövőbe mutató megoldást kínál a mai érintőképernyős szövegbeviteli interakció helyettesítésére, kiegészítésére. Ma még ugyan nem terjedt el széles körben a fejre rögzíthető kijelző [24], azonban megfontolandó tény, hogy például a Google is ebben látja a jövőt [25]. Ugyan az erős fényviszonyok képesek megbonyolítani a helyzetet, azonban ez a probléma még a mai érintőképernyős készülékek esetén is gondot okozhat. Megfontolandó továbbá az is, hogy az ujjak számának korlátozása sem válik a koncepció rovására, mert ugyan az érintőképernyővel rendelkező telefontulajdonosok is az esetek többségében két hüvelykujjal használják a virtuális billentyűzetet, a második leggyakoribb beviteli mód az egy mutatóujjas módszer [26]. Ezt ki lehetne küszöbölni, ha egy úgynevezett „time of flight” elvű [27] kamerával is kiegészítenénk a rendszert, mert így már nem csak vizuális visszacsatolás lenne, de rendelkeznenék mélységinformációval is, melynek segítségével jobban lehetne követni az ujjak térbeli mozgását. Egyes fórumokon [28] rendszeresen előfordulnak, negatív visszhangok, hogy a fekete-fehér jelölők nem túl esztétikusak, azonban léteznek már formatervezett jelölők is, amelyek megtörik az egyhangúságot. Ezekre láthatunk példát a 27. ábrán.



27. ábra – Formatervezett jelölők [29]

Jelen dolgozatból kiderül tehát, hogy ez a rendkívül kurrens kutatási terület, hogyan szakít magának egyre nagyobb szeletet a virtuális technikákra alapuló alkalmazások tortájából. A bemutatott alkalmazás a megfelelő peremfeltételek mellett – hardverháttér – egy kényelmes alternatívát kínál az ember-gép interakciók megvalósítására. A kutatások és workshopok száma egyre növekszik, így a jövőben egyre az ergonomikusabb, a mindennapi életünket megkönnyítő alkalmazások megjelenésének rohamos ütemben való növekedésére lehet számítani.

6 Irodalomjegyzék

- [1] **D.W.F van Krevelen, R. Poelman**, "A Survey of Augmented Reality Technologies, Applications and Limitations", *The International Journal of Virtual Reality*, 9(2), pp. 1-20, Delft, Netherlands, 2010
- [2] **G.A.Giraldi, R.Silva, J.C. Oliveira**, "Introduction to Virtual Reality", *LNCC Research Report*, Vol. 6., National Laboratory for Scientific Computation, Quintadinha, Brazil, 2003
- [3] **J. Fischer, A. del R'io, M. Neff, S. Flossmann, F. Duffner, J. Hoffmann, W. Straßer, D. Bartz**, "The ARGUS System: Surgical Mixed Reality based on Image Guided Surgery, Extensions, and applications", *In Proc. of Computer-Aided Surgery around the Head (CAS-H)*, 2007
- [4] **R. Lapeer, M. Chen, G. Gonzalez, A. Linney, G. Alusi**, "Image-Enhanced Surgical Navigation for Endoscopic Sinus Surgery: Evaluating Calibration, Registration and Tracking", *The International Journal of Medical Robotics and Computer Assisted Surgery*, 4(1):32–452, 2008
- [5] **F. Wacker, S. Vogt, A. Khamene, J. Jesberger, S. Nour, D. Elgort, F. Sauer, J. Duerk, J. Lewin**, "An Augmented Reality System for MR Image-guided Needle Biopsy: Initial Results in a Swine Model", *Radiology*, 238(2):497–504, 2006
- [6] **S. Zein, B. Preim, Georg Rose**, "An approach for enhanced slice visualization utilizing augmented reality: algorithms and applications", *Proc. of 3rd Palestinian International Conference on Computer and Information Technology (PICCIT)*, Hebron, USA, 2010
- [7] **G. Zheng, M. Caversaccio, R. Bchler, F. Langlotz, L. Nolte, R. Husler**, "Frameless Optical Computer- Aided Tracking of a Microscope for Otorhinology and Skull Base Surgery", *Archives of Otolaryngology - Head and Neck Surgery*, 127(10):1233–1238, 2001
- [8] **O. Oda, L. J. Lister, S. White, S. Feiner**, "Developing an augmented reality racing game" *In Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive enterTAINment*, p. 2., New York, USA, 2008
- [9] **E. Molla, V. Lepetit**, "Augmented Reality for Board Games" *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 1-8, 2010
- [10] **Metaio** [Online] <http://www.metaio.com/projects/engineering/man-replacement-of-a-ship-engine-in-cramped-spaces/>, utolsó látogatás: 2012.11.05.
- [11] **R.T Azuma**, "A Survey of Augmented Reality, *Presence: Teleoperators and Virtual Environments*", Vol. 4., pp. 355-385, Malibu, USA, August 1997
- [12] **Kyung, Wang-Jun, Kim D. C., Ha H. G., Ha Y. H.**, "Color enhancement for faded images based on multi-scale gray world algorithm", *2012 IEEE 16th International Symposium on Consumer Electronics (ISCE)*, pp. 1-4., Vilnius, Lithuania, 2012
- [13] **Kakumanu, Praveen, Makrogiannis S., Bourbakis N.**, "A survey of skin-color modeling and detection methods", *Pattern recognition* 40, Vol. 3., pp. 1106-1122, 2007
- [14] **M. H. Asmare, V. S. Asirvadam, L. Iznita**, "Color space selection for color image enhancement applications", *IEEE ICSAP 2009. International Conference In Signal Acquisition and Processing*, pp. 208-212, Kuala Lumpur, Malaysia, 2009

- [15] **Chang, Chih-Chung, Lin C. J.**, "LIBSVM: a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, Vol. 3., p. 27., 2007
- [16] **Cheddad, Abbas, Condell J., Curran K., Mc Kevitt P.**, "A skin tone detection algorithm for an adaptive approach to steganography", *Signal Processing* 89, Vol. 12., pp. 2465-2478, 2009
- [17] **A. De la Escalera, J. M. Armingol**, "Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration" *Sensors* 10, Vol. 3., pp. 2027-2044, 2010
- [18] **R. B. Rusu, N. Blodow, M. Beetz**, "Fast point feature histograms (fpfh) for 3d registration", *ICRA'09 IEEE International Conference on Robotics and Automation*, pp. 3212-3217, Kobe, Japan, 2009
- [19] **H. Yoon, N. Park, W. Lee, Y. Jang, W. Woo**, "QR Code Data Representation for Mobile Augmented Reality", *In The International AR Standards Meeting*, pp. 17-19, 2011
- [20] **S. Cawood, M. Fiala, D. H. Steinberg**, "Augmented reality: A practical guide", *Pragmatic Bookshelf*, pp. 1-27, 2008
- [21] **H. Kato, M. Billinghurst**, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System", *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality '99*, pp. 85-94, San Francisco, USA, 1999
- [22] **O. Oda, C. MacAllister, S. K. Feiner**, "Goblin XNA User Manual", *Columbia University Department of Computer Science*, New York, USA, 2012
- [23] **Ghazali, Bin K. H., Ma J., Xiao R.**, "An Innovative Face Detection based on Skin Color Segmentation." *International Journal of Computer Applications* 34, Vol. 2., 2011
- [24] **Brother** [Online] <http://www.brother.com/en/news/2011/airscouter/>, utolsó látogatás: 2012.11.09.
- [25] **Google** [Online] <http://plus.google.com/+projectglass>, utolsó látogatás: 2012.11.09.
- [26] **S. Azenkot, S. Zhai**, "Touch Behavior with Different Postures on Soft Smartphone Keyboards", *Proc. Mobile HCI 2012, San Francisco, USA, 2012*
- [27] **S. Schuon, C. Theobalt, J. Davis, S. Thrun**, "High-quality scanning using time-of-flight depth superresolution", *CVPRW'08. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1-7, Anchorage, USA, 2008.
- [28] **Neonia** [Online] www.neonia.hu, utolsó látogatás: 2012.11.09.
- [29] **S. Siltanen**, "Theory and applications of marker-based augmented reality", *Espoo 2012 VTT Science* 3, 2012