



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Híradástechnikai Tanszék

Horváth Kristóf Attila

Kinect alapú színpadi pozícionáló és világítás automatizálási rendszer

Konzulensek: Dr. Szabó Sándor
Lendvai Károly
Milánkovich Ákos

Budapest, 2012

Tartalomjegyzék

Összefoglaló.....	5
Abstract	6
1. Bevezető	7
1.1. Motiváció	7
1.2. Rendszerkomponensek.....	7
1.3. Lehetőségek a pozícionálásra.....	7
1.3.1. Háromszögelés, trilateráció	7
1.3.2. Távolságmérés	9
1.3.3. Inerciális navigáció.....	10
1.3.4. Képfeldolgozás alapú helymeghatározás.....	10
2. Kinect.....	11
2.1. A Kinect rövid áttekintése.....	11
2.2. Kinect for Windows	12
2.3. A készülék felépítése.....	12
2.3.1. Érzékelők	14
2.4. Az érzékelők működési elvei	15
2.4.1. Nyalábformálás.....	15
2.4.2. Mélységérzékelés.....	18
2.5. Kinect for Windows fejlesztői környezet.....	21
2.5.1. Rendszerkomponensek	21
2.5.2. Adatfolyamok	22
2.5.3. Csontvázkövetés	23
2.5.4. Arcok követése	26
2.6. Kinect tesztprogram	28
3. A Kinect pontossága	30
3.1. A mérés leírása.....	30
3.2. Mérés egyhelyben	31
3.3. Mélységi mérések.....	33
3.4. Nyers mélységi adatfolyam.....	37
3.5. Oldalirányú mérések	38
3.6. Összegzés, a Kinect színpadi alkalmazásának korlátjai.....	41

4.	Színpadvilágítás.....	42
4.1.	A színpadvilágítás rövid története.....	42
4.2.	Színpadi vezérléstechnika	44
4.3.	Lámpavezérlés.....	47
4.4.	Lámpavezérlő tesztprogram	50
5.	Követés	51
5.1.	Koordinátarendszerek	51
5.2.	Transzformációs modell.....	52
5.3.	Követő tesztprogram	53
6.	Kalibráció	54
6.1.	A kalibráció elve	54
6.2.	Kalibráció a gyakorlatban	55
7.	Integrált rendszer	56
7.1.	Felhasználói felület	57
7.2.	A program felépítése	58
8.	A rendszer tesztelése.....	60
8.1.	Késleltetés	60
8.2.	Érzékenység különböző fényviszonyokra.....	61
8.3.	Kalibráció.....	62
9.	Összegzés.....	63
10.	Továbbfejlesztési lehetőségek	64
	Forrásjegyzék.....	65
	Ábrajegyzék	67
	Táblázatok jegyzéke	68
	Rövidítések jegyzéke	69
	Függelék.....	70
	DTS Vinciscan fényvető színcsatorna kiosztása.....	70
	DTS Vinciscan fényvető gobócsatorna kiosztása	71

HALLGATÓI NYILATKOZAT

Alulírott **Horváth Kristóf Attila**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2012. 12. 04.

.....
Horváth Kristóf Attila

Összefoglaló

A szakdolgozat célja egy olyan rendszer megalkotása és bemutatása, mely a színpadon mozgó szereplő pozíciójának meghatározásával és robotlámpák vezérlésével képes a színpadtechnikában alkalmazott fejgépek manuális irányításának kiváltására, azok automatizálására. A dolgozat a rendszerkomponenseken végighaladva először a pozicionáló-, majd a lámpavezérlő alrendszert fejt ki, ezt követően mutatja be a két részt összekötő matematikai apparátust, a kalibrációt, valamint az integrált rendszert.

A pozíció meghatározására számtalan lehetőség ismert, ezek áttekintése után a választott eszköz, a Microsoft Kinect bemutatása, majd egy méréssorozat leírása következik, mely az érzékelő megfelelő pontosságának vizsgálatára irányult.

A lámpavezérlés kapcsán kitekintés történik a színpadvilágítás múltjára és a jelenleg alkalmazott világítástechnikai megoldásokra. A DMX512 a legelterjedtebb lámpavezérlő szabvány, szinte kivétel nélkül támogatja az összes világítástechnikai berendezés, így a megvalósított rendszerben is ennek segítségével történik a vezérlés. A szabvány bemutatása után a lámpavezérlés gyakorlati megvalósításának menete kerül leírásra.

A pozíció meghatározó- és lámpavezérlő alrendszert összekötő transzformációs modell, majd a modellparamétereket meghatározó kalibrációs eljárás ismertetésével zárul le a rendszerkomponensek tárgyalása. Mindegyik rendszerkomponens kipróbálására készült tesztprogram, ezek bemutatása az adott fejezeteknél található.

A már felhasználói felülettel is rendelkező, az összes funkcionalitást tartalmazó integrált rendszer bemutatása, tesztelése, végül a továbbfejlesztési lehetőségek taglalása zárja a dolgozatot.

Abstract

The objective of this paper is to create and present a system, which is able to replace the manual positioning of stage lighting follow spots by automatization. The system determines the position of the actor, and also controls the intelligent lights. The paper introduces the system components: first, the positioning- and the lamp control sub-system, second the mathematical background of the connecting of two sub-systems, the calibration, and then the integrated system are presented.

There are many ways for positioning, this paper gives an overview of these solutions. Then the chosen one, the Microsoft Kinect sensor is presented, and a series of measurements are done, to demonstrate the precision of the sensor.

A brief history of stage lighting and the current used solutions are presented. The DMX512 standard is the most widely used lamp control standard, almost every lighting equipment supports it, therefore my system also uses it for control. After the presentation of the standard, the progress of implementation is showed.

The transformation model between positioning- and lamp control sub-systems, and the calibration of model parameters close the discussion about the system components. Test programs are created to try every component, which are presented in the adequate chapters.

Then the integrated system, which has a graphical user interface and contains all functions, is presented and tested. Finally the opportunities of further development are discussed.

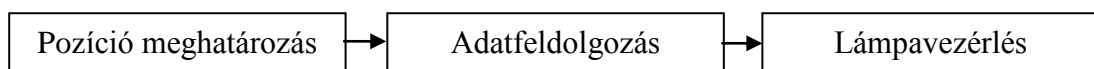
1. Bevezető

1.1. Motiváció

A színpadi világítástechnika gyakorta alkalmaz fejgépeket, mely berendezések célja, hogy a fontosabb színészeket egy fénykörrel megvilágítva kiemelje környezetükből. Manapság ezeket a lámpákat manuálisan, technikusok pozicionálják, és követik velük a színészek mozgását. A világítástechnikában azonban már évek óta alkalmaznak robotlámpákat, melyek automatikusan, a vezérlőpult utasításainak megfelelően képesek adott irányba fordulni, színt, effektet váltani, stb. Összevetve a fejgépeket és a robotlámpákat, jogosan merül fel az ötlet, hogy a ma már oly fejlett robotlámpák kiválthatnák a fejgépeket egy olyan rendszer segítségével, mely képes a színpadon mozgó színészek követésére, és a lámpák irányítására.

1.2. Rendszerkomponensek

Az 1. ábrán látható módon a fejlesztendő rendszer három részre bontható. Először detektálni kell a követendő objektum pozícióját. Ezt követően meg kell határozni a robotlámpák vezérléséhez szükséges vezérlőinformációkat. Végül ezen adatokat ki kell küldeni a lámpaparknak.

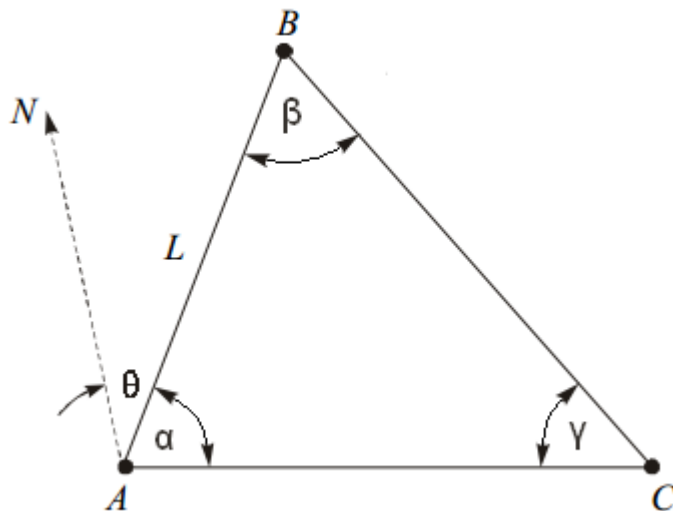


1. ábra Rendszerkomponensek

1.3. Lehetőségek a pozicionálásra

1.3.1. Háromszögelés, trilateráció

A háromszögelés módszere egy ismeretlen pont helyzetét szögek mérésével határozza meg. Kétdimenziós esetet feltételezve az alapelv a következő: a 2. ábrán látható módon adott egy alappont helyzete (A), és egy alapvonal (AB szakasz), melynek ismert a hossza (L) és iránya egy referenciáirányhoz képest (ami itt az N -nel jelölt irány). A háromszögelés során mérjük az ABC háromszög szögeit, és ezek alapján határozzuk meg C pont helyzetét.

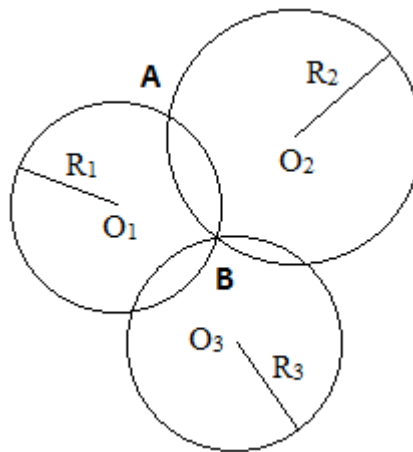


2. ábra Háromszögelés

A megoldás alapja a szinusz-tétel, AC szakasz hossza úgy aránylik L -hez, mint $\sin(\beta)$ $\sin(\gamma)$ -hoz. Mivel a szögek (így szinuszaik is) és L ismertek, AC szakasz hossza meghatározható. Az AC szakasz referenciáiránnyal bezárt szöge $\theta + \alpha$, e szög koszinuszával szorozva a szakasz hosszát az N referenciáirányba történő vetületének hossza, míg a szög szinuszával szorozva az N -re merőleges irányú vetületének hossza adódik. A pont koordinátáihoz az előbb kiszámított eltolásokat hozzáadva adódik C pont koordinátája. [1]

A világítótornyok alapján történő tengeri navigáció is az irányméréseken, így a háromszögelésen alapult, és a geodéziai alappont hálózatot is ezzel a módszerrel vették fel. Újabban az elektronikus mérőműszerek elterjedésével a szögek mérése helyett közvetlenül a távolságok mérése került előtérbe, a trilateráció ezt a módszert jelöli. A 3. ábrán látható módon, a keresett pont távolságát O_1 és O_2 pontoktól lemérve két körív vehető fel, e körök metszéspontja határozza meg a pont helyzetét. Érintőköröktől eltekintve két metszéspont jön létre, amik kijelölik az ábrán látható A és B pontokat. Amennyiben nincs kiegészítő információ arra vonatkozóan, hogy melyik metszéspont a keresett pont valódi helyzete, szükség van egy harmadik ponttól történő távolságmérésre is, ekkor az R_3 sugarú kör kiválasztja a megoldást, jelen esetben a B pontot.

A módszerek természetesen keverhetők, például radaralkalmazások az objektum helyzetének meghatározásához irány- és távolságadatokat is gyűjtenek. [2]



3. ábra Trilateráció

1.3.2. Távolságmérés

A trilateráció alapjául szolgáló távolságmérés megvalósítható fázisméréssel. A geodéziában alkalmazott távmérő berendezés modulált lézerfényt bocsájt a mérőpontban elhelyezett tükörrre. A visszaverődő fényből demodulált mérőjel fázisát összehasonlítja a kibocsájtottéval, a fáziskülönbségből számítható az úthossz, azaz jelen esetben a távmérő berendezés és a tükör távolságának kétszerese. Nagy pontosság eléréséhez nagyobb frekvenciát alkalmaznak, ennek hullámhossza azonban kisebb, mint a mérendő távolság, így a megoldás nem egyértelmű. A probléma megoldására több mérőjellel is elvégzik a mérést. [3]

A távolságmérés alapja a terjedéshez szükséges idő mérése is lehet, ebben az esetben a terjedési idő és a jelterjedési sebesség szorzataként adódik a távolság. Az ultrahangos távolságmérő berendezések többnyire ezt az elvet használják ki. Rádiós technológiában is megvalósítható ez az elv, azonban a rádióhullámok fénysebességgel terjednek, így nagyon pontos időmérés szükségeltetik ehhez. Az UWB (Ultra Wideband) technológia a költségek alacsony szinten tartása érdekében ezt úgy éri el, hogy a megfelelően nagy pontosságot nyújtó frekvencia helyett alacsonyabb frekvencián mintavételezi a beérkezett jelet, azonban a speciális jelalak miatt (amit a felhasznált nagy sávzsélesség okoz) keresztkorrelációs függvény számításával és interpolációs eljárás alkalmazásával a mintavételi frekvenciából adódó periódusidőnél pontosabb időkülönbségek is mérhetők, mellyel 1 méternél pontosabb távolságmérés is lehetővé válik. [4][5]

További lehetőség a jelerősség mérése, a jelterjedési modell alapján a vett jel erősségéből megállapítható a jelforrás távolsága. A módszer előnye, hogy rendkívül

egyszerű. Manapság a legtöbb mobiltelefon már rendelkezik beépített Wi-Fi-vel, így viszonylag olcsón telepíthető olyan hozzáférési pontokból álló infrastruktúra, melynek segítségével az ilyen telefonnal rendelkezők számára lehetőség nyílik a helymeghatározásra. Illetve a jelenleg kiépített Wi-Fi hozzáférési pontok és mobil bázisállomások feltérképezése is lehetséges közösségi alapon, ezt a lehetőséget használja ki a Google helymeghatározó szolgáltatása is. A jelerősség mérésének árnyoldala, hogy viszonylag kis pontosság érhető csak el vele, ugyanis a jelterjedés a valóságban rendkívül bonyolult, a közelítő modellek nem elég pontosak. [6][7]

1.3.3. Inerciális navigáció

Egyre több mobiltelefonba építenek giroszkópokat és gyorsulásmérőket. Ezen eszközökkel lehetővé válik az inerciális navigáció. A pozíció meghatározásának elve ez esetben a következő: a gyorsulásmérők által szolgáltatott adatsorok kétszeres integrálásával elmozdulás információkat kapunk a tér három irányában. A giroszkópok azért szükségesek, hogy a készülék aktuális orientációja ne befolyásolja a mérést – a giroszkópok szögelfordulásai alapján a gyorsulásmérők adatait egy előre rögzített koordináta-rendszerbe transzformáljuk. [8]

A módszer legnagyobb hátránya, hogy csupán elmozdulás információt szolgáltat, így pozíció meghatározáshoz a mérés elején ismerni kell az eszköz kiinduló pozícióját. Ezt követően pedig a helyzetmeghatározásra vonatkozó hibák halmozódnak, így az idő elteltével egyre nagyobb bizonytalansággal ismert csak az aktuális pozíció.

1.3.4. Képfeldolgozás alapú helymeghatározás

A projekt megvalósítása során alkalmazott Microsoft Kinect képfeldolgozás útján határozza meg az előtte állók pozícióját. A módszer előnye, hogy nincs szükség a háromszögelés vagy trilateráció használatához szükséges infrastruktúra kiépítésére. A Kinecttel nagy pontosság érhető el (ezt a későbbiekben mérésekkel igazolom is), lényegesen nagyobb, mint azt jelterjedési modellből számolni lehetne, de az UWB néhány 10 cm-es pontosságát is felülmúlja – ez elengedhetetlen a megvalósítás szempontjából. Mindemellett az eszközhöz biztosított függvénykönyvtár viszonylag egyszerű fejlesztést tesz lehetővé.

2. Kinect

2.1. A Kinect rövid áttekintése

A Microsoft 2010-ben jelentette meg Kinect elnevezésű eszközét, melyet eredetileg az Xbox 360-as játékkonzoljukhoz fejlesztettek ki. A berendezés a 4. ábrán látható. Elnevezése a „kinetic” és „connect” szavak összetételéből fakad. A cél egy olyan periféria kialakítása volt, melynek segítségével a játékos kontrollerek kézbevétele nélkül irányíthatja a játékokat. [9]



4. ábra Kinect [10]

Korábban a Nintendo Wii-vel együttműködő Remote elnevezésű kontroller volt forradalmi újítás a játékiparban (5. ábra), ugyanis ezzel a kontrollerrel nyomógombok, joystick-ok helyett magával az eszköz mozgatásával lehetett a konzolnak utasításokat adni. Bizonyos játékoknál rendkívül hatékonyan kihasználható ez az újfajta irányítási metódus, azonban a kontroller jelenléte zavaró, a rendszer csak a kontroller mozgását képes érzékelni, a játékos többi testrészének mozgásait nem, ráadásul a játék hevében, a kelletténél hevesebben meglendített kontroller olykor a felhasználó kezéből kicsúszva a televízió képernyőjében landolt, hatalmas kárt okozva ezzel. [11] [12]



5. ábra Wii Remote [13]

A Microsoft Kinect elnevezésű berendezése a fenti problémáktól mentes, ugyanis a játékosok kezében egyáltalán nincsenek vezérlők, a készülék az előtte állók egész testét érzékeli, a testrészek megfelelő mozgásaival lehet a programokat irányítani. A kézben tartott kontrollereket elhagyva lehetőség nyílik a többi testrészt is bevonni a mozgásba, így természetesen a technológiát kihasználó legtöbb program sport tematikájú, a velük történő játék az egész testet megmozgatja – rafting, tenisz, ping-pong, gátfutás, gerelyhajítás, bokszt, röplabda, foci is megtalálható a játékkonzolra írt programok között, csak a legfontosabbakat említve. [14]

2.2. Kinect for Windows

A Kinectben rejlő lehetőségek azonban nem merülnek ki a játékkonzolok világában, ezt felismerve a Microsoft 2012 februárjában piacra dobta Kinect for Windows elnevezésű termékét, melyet a korábbi verzióval ellentétben már számítógépes használathoz készítettek. A Kinectre történő fejlesztést ösztönözvén ingyenes fejlesztői környezetet biztosít a vállalat, ami remélhetőleg rövid idő múlva számtalan program megszületését fogja eredményezni, ugyanis jelen pillanatban még nem érhetők el Windowson futtatható, a Kinect képességeit használó szoftverek, egyelőre a Kinect for Windows-t is csak fejlesztőknek, vállalatoknak kínálja a cég, a végfelhasználóknak történő értékesítés még várat magára. [15]

A Kinect for Windows, levetkőzve a játékkonzolok világának behatároltságát, már alkalmas lehet bizonyos speciális műszaki problémák megoldására, ilyen például a színészek követése is jelen projektben.

Mivel eredetileg az Xbox 360 játékkonzolhoz fejlesztették ki a Kinectet, és az abba beépített számítási kapacitás erősen korlátozott, ott elengedhetetlen volt a kiegészítő hardveres feldolgozás. A modern asztali számítógépek valószínűleg elboldogulnának a nyers adatfolyamok feldolgozásával is, de még itt is jelentős teljesítménynövekedés érhető el a készülék beépített hardveres megoldásainak köszönhetően. [9]

2.3. A készülék felépítése

A Kinect tulajdonképpen több érzékelő egybeépítve, kiegészítve a témérdek adat kiértékelését gyorsító hardveres feldolgozóegységgel. Éppen ezért talán meglepő a

berendezést egyszerűen szenzornak titulálni, azonban a hivatalos források is többször hivatkoznak így az eszközre.

Az eszköz egy fejrészből, egy talpazatból, és az abból kivezető kábelből áll, melyen keresztül egy USB porthoz kapcsolódhat. Bár a csatlakozó szabványos USB kábel, az SDK (Software Development Kit) dokumentációja felhívja a figyelmet arra, hogy a kommunikáció eltér a szabványban rögzítettektől. Az eszköz áramfelvétele is jelentős, így a Kinectből kilépő kábel a számítógéppel, és egy tápon keresztül az elektromos hálózattal is kapcsolatban áll. A talpazat és a fej bezárt szöge egy motorral változtatható, így a fej elevációs szöge tetszőlegesen beállítható $\pm 27^\circ$ között, aminek aktuális értéke a fejben található eszközökkel mérhető is. Ennek az a jelentősége, hogy az érzékelő 60 centiméter és 2 méter közötti magasságban tetszőlegesen elhelyezhető, így ennek megfelelően szoftverek fejlesztése során gondoskodni kell arról, hogy a szenzor magasságától függetlenül, az elevációs szög állításának segítségével a kamerák látótere mindig lefedje az előtte állókat. Tulajdonképpen minden fontosabb elem, az érzékelők és a feldolgozó egység is a fejben található, melynek belseje látható a 6. ábrán. [16]



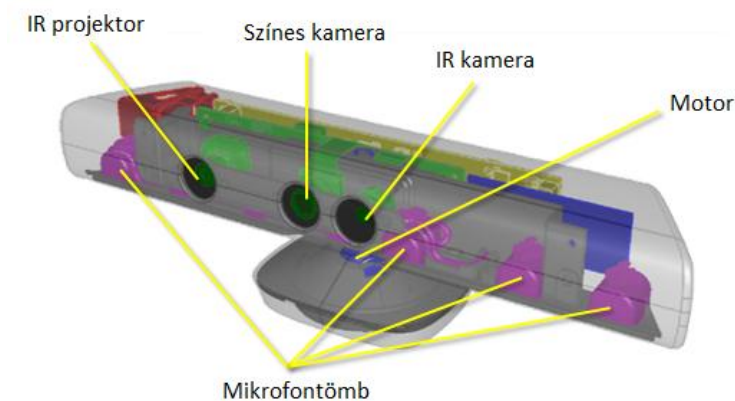
6. ábra A Kinect belseje [17]

A szenzoroktól származó adatfolyamokat feldolgozó hardveres egység lelke az izraeli PrimeSense által kifejlesztett PS1080-A2 kódú chip, melyhez egy 64MB-os, Hynix gyártmányú DDR2 memóriablokk kapcsolódik. Az eszköz hűtésére egy

ventilátor is helyet kapott a többi alkatrész mellett – az Xbox 360-as játékkonzolok esetében túlmelegedési problémák léptek fel, azonban a Kinect esetében a hűtés remekül működik. [17]

2.3.1. Érzékelők

A 7. ábrán látható, különböző érzékelők képi- és hanginformációkat is rögzítenek. A képi jelfeldolgozás alapja az előlapon található két kamera. Az egyik egy teljesen közönséges, színes kamera (640*480 pixeles VGA felbontással). A belátott tartomány vízszintesen 43°, míg függőlegesen 57°, a képfrissítés gyakorisága 30 kép másodpercenként. [18]

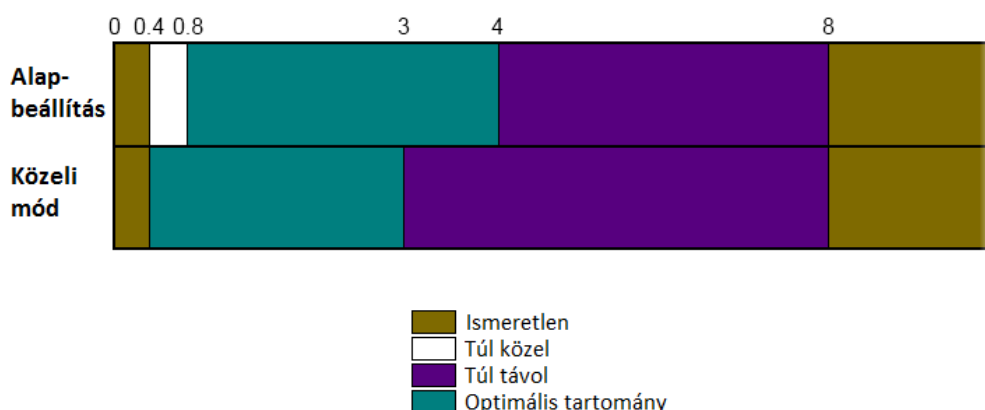


7. ábra Érzékelők és a motor elhelyezkedése [18]

A másik kamera is hasonló, azonban ez az infravörös tartományban érzékel, és a három színinformáció rögzítése helyett monokróm üzemmódban működik. Ehhez tartozik a készülék előlapjának átellenes oldalán található IR projektor, ennek segítségével a készülék a környezetére apró infravörös pontokat vetít egy pszeudorandom minta alapján. Az infravörös kamerát sokszor csak mélységérzékelő kamerának hívják a különböző leírásokban, és valóban, olyan bitfolyam érkezik a berendezés felől, melyben minden pixelhez egy-egy távolságérték van rendelve.

A mélységérzékelés a belátott szögtartomány korlátjain kívül további megszorításokat tartalmaz, melyet a 8. ábra szemléltet. A 40 cm-nél közelebbi és 8 m-nél távolabbi objektumok teljesen értelmezhetetlenek, és optimális feldolgozás csak 80 cm és 4 m között lehetséges. A Kinect for Windows egyik újítása, hogy bevezették a közeli módot, ugyanis rengeteg visszajelzés érkezett arról, hogy kis lakásban rendkívül

körülményes a megfelelő távolság kialakítása. Ebbe az üzemmódba átkapcsolva az optimális tartomány 40 cm-től 3 m-ig terjed.



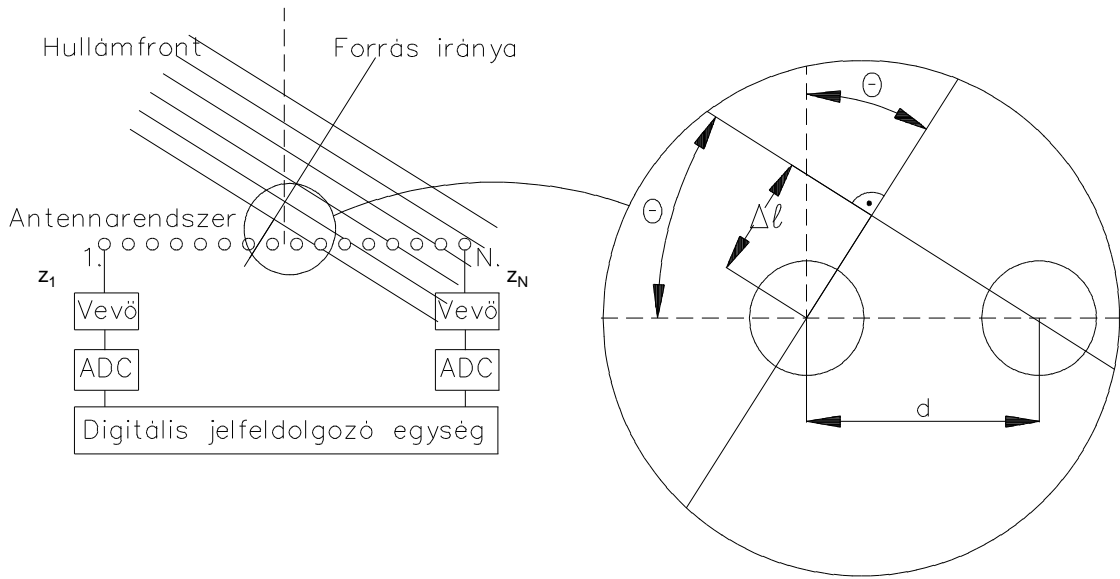
8. ábra Működési tartományok [18]

A hanginformációkat négy mikrofon dolgozza fel. Ezek mindegyike 24 bites A/D átalakítóhoz kapcsolódik. A mikrofon sor konstrukciójából fakadóan képes nyalábformálásra, ami egyrészt a visszhang- és zajelnyomáshoz, másrészt pedig a forrás irányának felderítésére alkalmas, ami több játékos nyomon követése esetén elengedhetetlen. Nyalábformálással 11 előre rögzített iránykarakterisztika közül lehet választani $\pm 50^\circ$ között 10° -os lépésekben, valamint lehetőség van az optimális karakterisztika automatikus felderítésére. A Microsoft Speech API (Application Programming Interface) használatával lehetőség nyílik beszédfelismerésre is. [18]

2.4. Az érzékelők működési elvei

2.4.1. Nyalábformálás

A Kinect esetében a mikrofon sor akusztikus nyalábformálásra képes. Nagyfrekvenciás rendszerek esetében szintén szükséges lehet a nyalábformálás – körsugárzó antennák helyett valamilyen irányítottsággal rendelkező antennák sokhelyütt előfordulnak. Erre megoldást jelenthet egy megfelelő geometriával rendelkező apertúra antenna, például egy forgásparaboloid antenna, azonban antennarendszer használatával is kialakítható irányítottság. A különböző antennarendszerek leírása gazdag irodalommal rendelkezik, mikrofon sorok analóg módon tárgyalhatók, azonban ekkor elektromágneses hullámok helyett hanghullámok képezik a vizsgálat tárgyát. [19]



9. ábra θ szögből érkező síkhullám [19]

Induljunk ki egy lineáris ekvidisztáns antennarendszerből, mely N elemű. (Tehát N darab antenna egy vonalban, egymástól egyenlő távolságra helyezkedik el.) A 9. ábrán látható módon érkezzék θ szögből egy síkhullám. Így az egyes antennákon adott időpillanatban a vett jel csupán fázisában különbözik. A fáziskülönbség az úthosszkülönbségre vezethető vissza, mely a

$$\Delta l = d \cdot \sin \theta$$

képletből számítható. Ebből adódik az elemek közötti fáziskülönbség:

$$\Delta \Phi = \Delta l \cdot \frac{2\pi}{\lambda} = d \cdot \frac{2\pi}{\lambda} \cdot \sin \theta$$

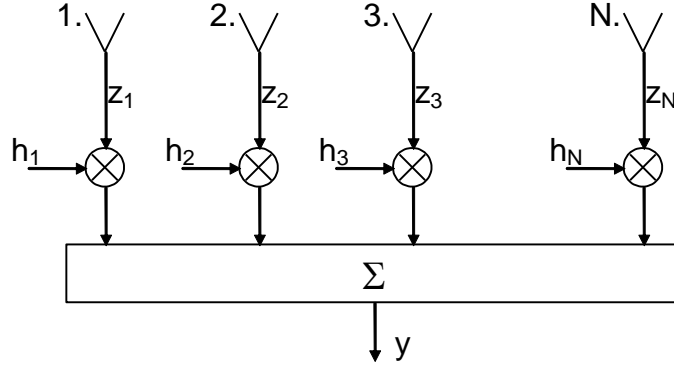
Így a k . elem fázisa:

$$\Phi_k = k \cdot \Delta \Phi = k \cdot 2\pi \cdot \frac{d}{\lambda} \cdot \sin \theta$$

A vett jel vektora az eddigiek alapján a következő:

$$\mathbf{s}^T(\theta) = [1 \quad e^{-j\Delta\Phi} \quad e^{-j2\Delta\Phi} \quad e^{-j3\Delta\Phi} \quad \dots]$$

A cél egy olyan vevőberendezés létrehozása, mely a θ szögből érkező síkhullámokra a legérzékenyebb. Ez a vett jel fázismenetének kompenzálásával oldható meg. Erre a 10. ábrán látható elrendezés lehet alkalmas.



10. ábra Antennarendszer [19]

Az antennákról érkező jelet egy-egy konstans állandóval szorozzuk, majd egy összegzőn keresztül kapjuk a kimenetet. Azonos szorzók esetén a merőlegesen beérkező síkhullám okozza a kimeneten a legnagyobb jelet, hiszen ekkor fázismódosítás nélkül, fázisban adódnak össze az egyes antennára beérkező jelek. θ szög felé irányítottág esetén a szorzóknak úgy kell viselkedniük, hogy az ebből a szögből beeső síkhullámok a szorzókkal kompenzálva az összegző bemenetére már fázisban érkezzenek. Ebből a kompenzáló súlyvektorra a következő adódik:

$$h(\theta) = s^*(\theta)$$

Mivel $s^T(\theta)$ a fentiek alapján

$$s^T(\theta) = [1 \quad e^{-\Delta\Phi} \quad e^{-2\Delta\Phi} \quad e^{-3\Delta\Phi} \quad \dots],$$

így a kompenzáló súlyvektor:

$$h^T(\theta) = [1 \quad e^{+\Delta\Phi} \quad e^{+2\Delta\Phi} \quad e^{+3\Delta\Phi} \quad \dots]$$

A Kinect esetében lényeges különbséget az antennák mikrofonokra cserélése nem jelent. Az viszont a fenti számítási lépésekhez képest újdonság, hogy az eszköz nem ekvidisztáns mikrofonsort alkalmaz. Ebben az esetben a probléma megoldása természetesen némileg bonyolódik, azonban létrehozható olyan konstrukció, mely egy

ekvidisztáns mikrofonosor tulajdonságaival bír, azonban „foghíjas”, és ebből fakadóan kevesebb elemet tartalmaz. A nagyfrekvenciás irodalom ezt a konstrukciót minimum redundanciájú antennának nevezi. [20]

2.4.2. Mélységérzékelés

Mint sok más technikai megoldás esetében, a Kinect mérnökei is biológiai ismeretek alapján oldották meg az eszköz mélységérzékelését. Ebből kifolyólag érdemes röviden kitérni az emberi mélységérzékelésre.

Optikai mélységészlelés

Az optikai mélységészlelés csoportjába a két szem konvergenciája és a szemlencse akkomodációja tartozik. Bár egyik sem tudatosul bennünk, agyunkba eljutnak ezen információk, és távolságérzetet okoznak. A konvergencia esetén minél párhuzamosabb a két szem tengelye, annál messzebbre nézünk, közelebbi pontokra történő fixálás esetén a két szem egyre inkább az orr irányába fordul. Az akkomodáció a szem fénnytörő rendszerének egyetlen változtatható elemének, a szemlencse domborúságának alkalmazkodását jelenti a fókuszban tartani kívánt távolsághoz. [21]

Képi mélységészlelés

A látás során a szem optikai rendszerén túl agyunk magából a képi információból is következtet a különböző objektumok távolságára.

- Atmoszférikus perspektíva: a távolban lévő tárgyakat elmosódottabbnak látjuk
- Takarás: az eltakart tárgy az eltakaró tárgy mögött található
- Relatív nagyság: a közelebbi tárgyak nagyobbak
- Relatív függőleges pozíció: a horizonthoz közelebbi tárgyak messzebb helyezkednek el
- Lineáris perspektíva: távolodó párhuzamos vonalakat összetartónak érzékelünk

Rengeteg optikai csalódás a fenti jelenségeken alapszik, többek között a 11. ábrán látható Ames-szoba, melyet a Csodák Palotájában is felépítettek. A szoba a perspektíva eszközével azt a hatást kelti, mintha az oldalirányba mozgó személy magassága folyamatosan változna. [22]



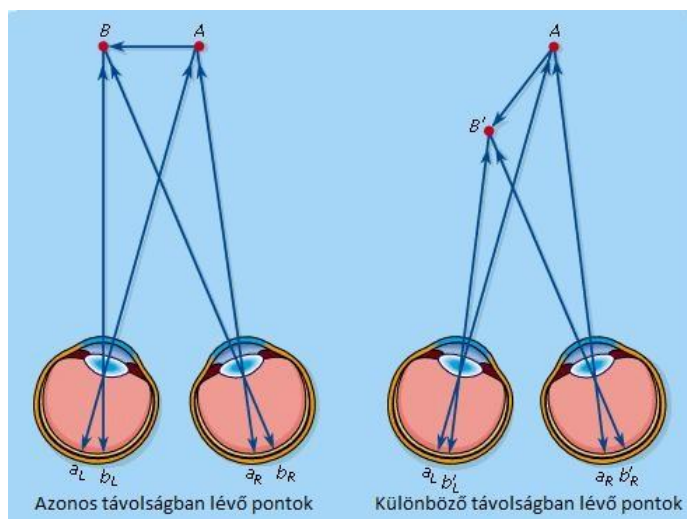
11. ábra A Csodák Palotájában található Ames-szoba [22]

Mozgásparallaxis

Fejünket mozgatva is mélységinformációhoz juthatunk. Legjobb szemléltető példa erre az, hogyha kinézünk egy mozgó autóból, a fixált ponttól közelebb elhelyezkedő tárgyak az autó haladási irányával ellentétesen, míg a távolabb elhelyezkedő tárgyak azzal megegyező irányba mozognak. [21]

Binokuláris látás

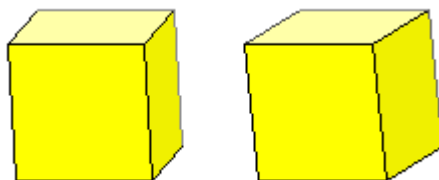
A legfontosabb mindközül azonban a két szemmel történő mélységérzékelés, térérzékelésünk leginkább erre támaszkodik. A jelenség alapja a retinális diszparitás. A 12. ábrán látható, hogy a szemlélőtől azonos távolságban lévő tárgyak képe a bal és jobb szem retináján is egymástól azonos távolságban váltanak ki ingerületet. Amennyiben az egyik tárgy közelebb van, úgy ez a szimmetria sérül, az eltérés nagyságából számítja ki agyunk a tárgyak egymáshoz képesti viszonyát. Amennyiben a fixált pont távolságától a másik tárgy távolsága egy bizonyos mértéknél jobban eltér, a tárgy két retinán létrejött képei már nem is fuzionálhatók, két képet látunk. [23]



12. ábra Retinális diszparitás [21]

A jelenség könnyen demonstrálható: helyezzük egyik ujjunkat a következő sorra, majd a sor olvasása közben (ez biztosítja a fixáció megtartását) közelítsünk ujjunkkal az arcunk felé! Szinte azonnal kettőt fogunk látni ujjunkból, és a közelítés mértékétől függően a két kép távolsága egyre növekszik. Bár zavarónak tűnhet a jelenség, a mindennapok során ritkán vehető észre, ugyanis az éppen feldolgozandó részletet mindig fixálja látórendszerünk, a retinális diszparitás szerepe pedig a térérzékelésben mutatkozik meg.

Amennyiben a két szemünk számára eltérő képet tudunk mutatni, lehetővé válik a fenti jelenséget kihasználva háromdimenziós érzet kiváltása. A 13. ábrából is egy térhatású kép alakul ki, amennyiben sikerül a kocka két nézetét külön-külön szemmel néznünk. Ehhez mindössze annyit szükséges tenni, hogy a távolba kell meredni, azaz a kép mögött, egy képzeletbeli távoli pontra kell fixálni, egészen addig, míg a két nézet fedésbe kerül – ekkor valójában egyik szemünk az egyik, míg a másik szemünk a másik képre fog tekinteni, agyunk pedig a látottak alapján összeállítja a térhatású kockát.



13. ábra A binokuláris látás segítségével térhatású kocka

A manapság egyre népszerűbb 3D filmvetítési technika is a binokuláris látást használja ki. Kétségkívül ez a legmarkánsabb térérzékelési rendszerünk, azonban nem szabad megfeledkeznünk arról, hogy a fent leírt egyéb módszerek segítségével is mélységinformációhoz jutunk. A moziban állandóan a vászonra fókuszálunk, a fixálás csak a rendező által fókuszban tartott képrészletnél hozhat kielégítő eredményt, nem működik a mozgásparallaxis, stb. Ezek mind ellentétes információkat küldenek az agyunknak, és bár a binokuláris látás elnyomja ezeket az ingereket, így nem fog megszűnni a ránk erőltetett térérzékelés, ennek ellenére hányingert, fejfájást okozhat. [24]

A Kinect mélységérzékelése

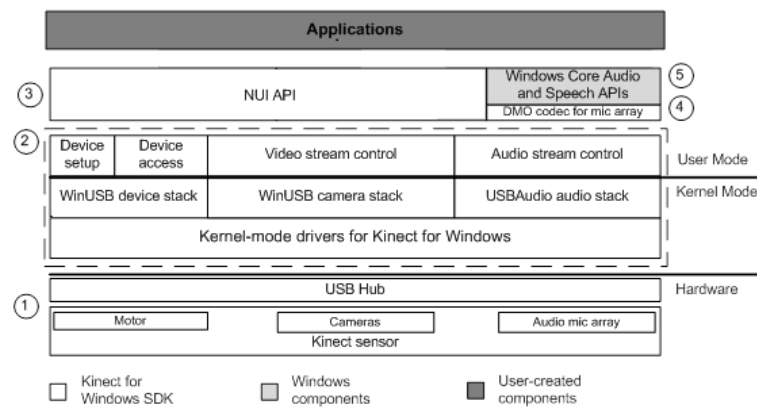
A távolságérzékelés tulajdonképpen a Kinect esetében is a binokuláris látáson alapszik. A jelenség két kamerával is szimulálható, a két képen előforduló azonos részletek pozíciójának különbségéből mélységinformáció nyerhető. A Kinect mélységérzékelésében azonban még egy csavar található: csupán egy kamera képe áll rendelkezésre, legalábbis ami a mélységérzékelést illeti. Azonban a sztereo feldolgozáshoz szükséges másik képet éppen az infravörös kamera vetíti a környezetre, így ez is rendelkezésre áll a mélységinformáció számításához. [25]

2.5. Kinect for Windows fejlesztői környezet

2.5.1. Rendszerkomponensek

A Microsoft által biztosított ingyenes fejlesztőkörnyezet számtalan módon segíti a programozók munkáját a Kinectet használó alkalmazások fejlesztése közben. Letöltése és telepítése két részletben történik. A Kinect for Windows SDK tartalmazza a driver-eket, valamint az eszközzel kapcsolatot tartó függvénykönyvtárakat, interfészeket. A Kinect for Windows Developer Toolkit további hasznos eszközöket nyújt a fejlesztéshez: példa forráskódokat, a Kinect Studio elnevezésű monitorozó eszközt, valamint a Face Tracking SDK-t, mellyel elérhetővé válik a későbbiek során bemutatásra kerülő arckövető funkció.

A szenzor felől érkező adatfolyamokat a különböző programok az úgynevezett Natural User Interface (NUI) segítségével érhetik el. Az SDK teljes architektúráját a 14. ábra szemlélteti.



14. ábra SDK architektúra [18]

1. Hardver komponensek: Ebbe a csoportba tartoznak az érzékelők tárgyalásánál bemutatott kamerák és a mikrofonosor, az elevációs szög beállító motor, valamint a számítógéppel kapcsolatot teremtő USB csatoló.
2. Driver-ek: Lehetővé teszik a hang és videó adatfolyamok kezelését, biztosítják a mikrofonosor elérését a Windowsban található szabványos audio API-k segítségével, valamint lehetővé teszik, hogy egy alkalmazás több Kinecthez is kapcsolódhasson egyidejűleg. (Azonban egy Kinect egyszerre csak egy alkalmazással állhat kapcsolatban.)
3. Natural User Interface: Függvénykönyvtár az adatfolyamok kezeléséhez, valamint a csontvázkövetéshez.
4. DirectX Media Object: Ez a komponens valósítja meg a mikrofontömb nyálábformálását, és határozza meg a hangforrás irányát.
5. Windows 7 függvénykönyvtárak: A Windows 7 SDK, illetve a Microsoft Speech SDK segítségével szintén elérhetők az eszköz bizonyos funkciói.

A függvénykönyvtárak C++, C# és Visual Basic nyelvekre implementáltak. [18]

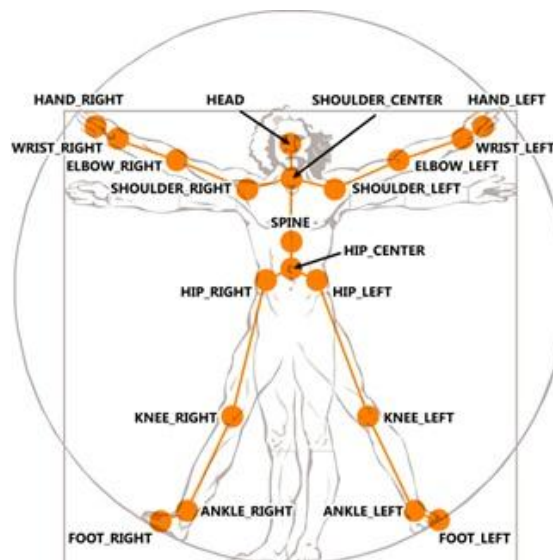
2.5.2. Adatfolyamok

A Natural User Interface segítségével elérhetők a Kinect által szolgáltatott különböző adatfolyamok. A NUI API inicializálásakor meg kell határozni a használni kívánt adatfolyamokat, a későbbiekben másik adatfolyamot elérni csak az eszköz újrainicializálása után lehet. Az érzékelőktől hang-, színes kép- és mélységi adatfolyam kérhető le. A függvénykönyvtár rengeteg eszközt nyújt az adatfolyamokkal történő manipulálásra is.

Az adatfolyamok egyes adatkeretei szelektív hívással vagy eseménymodell segítségével érhetők el. Előbbi esetben a forráskódban elhelyezett megfelelő függvény a keret rendelkezésre állásáig felfüggeszti a program végrehajtását, míg utóbbi esetben eseménykezelő függvények társíthatók az adatfolyamokhoz, melyek a keretek elkészültekor automatikusan hívódnak meg. [18]

2.5.3. Csontvázkövetés

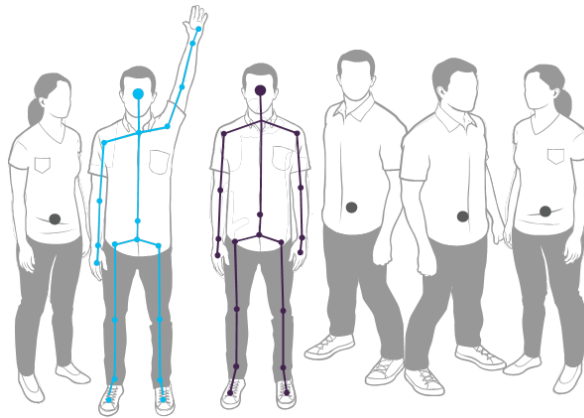
A Kinect talán legnagyobb sikerű funkciója, hogy képes az előtte állókat automatikusan detektálni, és különböző testrészeit követni. A rendszer egy-egy játékos 20 testrészének pozícióját határozza meg minden képkocka alapján, ezek az alábbiak: fej, jobb kézfej, bal kézfej, jobb csukló, bal csukló, jobb könyök, bal könyök, jobb váll, bal váll, vállvonal közepe (nyak), gerinc, csípő közepe (köldöktájék), csípő jobboldala, csípő baloldala, jobb térd, bal térd, jobb boka, bal boka, jobb lábfej, bal lábfej.



15. ábra A csontvázkövetés által támogatott testrészek [18]

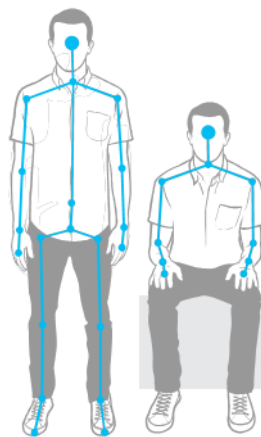
A 15. ábra a megfelelő függvényekben közvetlenül alkalmazható elnevezéseket tartalmazza. Fontos megjegyezni, hogy az ujjak pozíciójáról semmiféle információt nem szolgáltat a szenzor.

A Kinect számítási kapacitása is véges, éppen ezért a követhető emberalakok száma limitált, nem csak a kamerák által belátott terület geometriai tulajdonságai szabnak határt ennek. Egyidejűleg 6 csontváz követhető, azonban ebből aktívan csupán kettő. Passzív követés esetén a csontváz egészéről kapunk csak információt, az egész csontvázat egyetlen pont pozíciója írja le csupán, ezt mutatja a 16. ábra.



16. ábra Aktív és passzív követés [18]

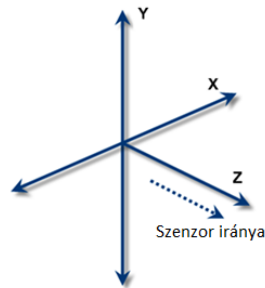
Aktív követés esetén érhető el a fentebb felsorolt 20 testrész egyedi meghatározása. A csontvázkövetés funkció csak álló emberalakokat képes teljesen felismerni, fekvő személyek követése egyáltalán nem támogatott, ülő testhelyzetben pedig a felsőtestről, karokról kapunk információt 10 pont erejéig, a 17. ábrán látható módon. További megkötés, hogy a készülék fele kell nézni, teljesen oldalvást, vagy háttal állva a működés bizonytalan.



17. ábra Álló és ülő mód [18]

A testrészek követése során a pozíciók egy a készülékhez rögzített koordinátarendszerbe képződnek le, melyet a 18. ábra szemléltet. A koordinátarendszer Descartes-féle, az x , y , z tengely ebben a sorrendben jobbsodrású rendszert alkot. A z tengely a szenzor érzékelési irányába néz, az x tengely a két kamerát összekötő egyenesen helyezkedik el, és a készülékkel szembe fordítva jobb oldalt található a pozitív tartománya. Nem szabad megfeledkeznünk arról, hogy egy motor segítségével a berendezés fejének elevációja változtatható, ekkor az x tengely körül elfordul az y és a z

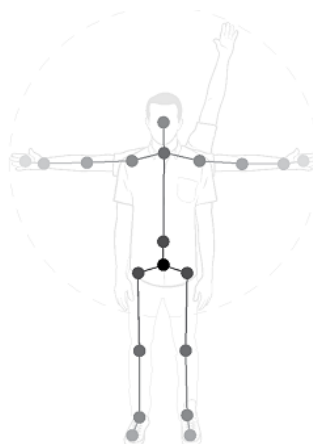
tengely. A csontvázak és testrészek pozícióinak értékei várhatóan az x tengely mentén -2,2 m-től 2,2 m-ig, az y tengely mentén -1,6 m-től 1,6 m-ig, míg a z tengely mentén 0-tól 4 m-ig terjedhetnek az érzékelő korlátjaiból fakadóan.



18. ábra A Kinect koordinátarendszere

A testrészek pozícióján túl lehetőség van a csontok orientációjának lekérdezésére is. A reprezentált csontok nem valódi, biológiailag értelmezhető csontok, hanem a megfelelő testrészeket összekötő szakaszok, azonban az ezek segítségével felépíthető csontváz tökéletesen alkalmas egy virtuális test felépítéséhez, a játékok során alkalmazott avatárok létrehozásához nem szükséges ennél részletesebb, biológiailag pontosabb leírás.

A csontok reprezentációjához a korábban bemutatott 20 testrész egy fá gráfba rendezett, melynek gyökere a csípő középrésze, a csontok pedig a gráf élei. A gráfot a 19. ábra mutatja be. Ülő testhelyzetben történő követésnél a gráf gyökere a vállvonal középrésze.

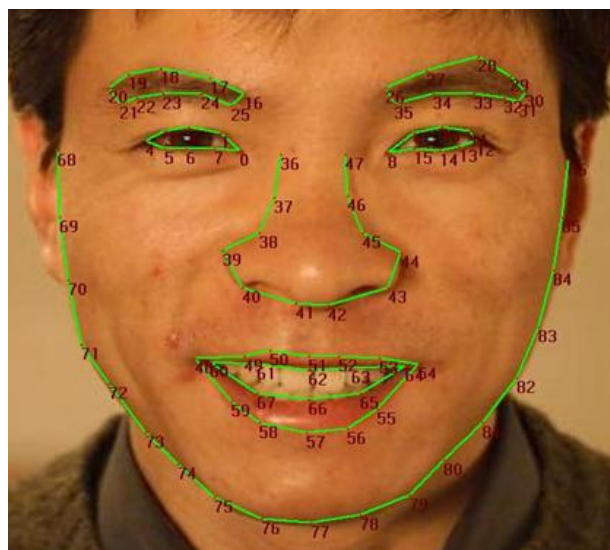


19. ábra A testrészek hierarchikus szervezése [18]

A csontok orientációját a gráf megfelelő éléhez tartozó két pont közül a gyermekelem tárolja. A szülőelem orientációjához képesti eltérés kódolt. A játékos abszolút orientációját a gyökérelem (a csípő közepe, ülő testhelyzet esetén a váll közepe) tartalmazza, a kamerával tökéletesen szembefordult állapothoz képesti eltérés kódolt. [18]

2.5.4. Arcok követése

A Face Tracking SDK segítségével lehetőség nyílik nemcsak az érzékelő előtt állók csontvázát, hanem az arcukat is nyomon követni. A függvények itt is a csontvázkövetésnél már bemutatott koordinátarendszert alkalmazzák. A különböző metódusok a Kinect színes kép és mélységi adatfolyamával dolgoznak, így ezek minősége jelentősen meghatározza a végeredményt. A jó eredmény érdekében az alakoknak jól megvilágítottaknak, a képnek élesnek kell lennie. Természetesen annál pontosabb lesz az arc egyes részleteinek detektálása, minél nagyobb területet foglal el a képből.

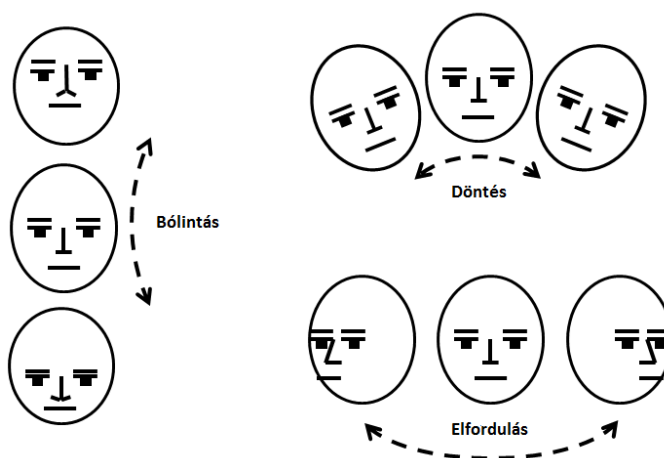


20. ábra Arckövetés által meghatározott pontok [18]

Az arc feldolgozása során számos információ kérdezhető le. 87 pont kerül detektálásra, ezek egy tömb elemeiként egyenként elérhetők. Ezen pontok csak síkbeli információval rendelkeznek, a tömb elemei a szenzortól megkapott RGB kép egy-egy pixelére mutatnak. (A 20. ábrán 13 pont nincs jelölve.) A fej bólintása, döntése és elfordulása szintén lekérdezhető (a 21. ábra szemlélteti a megfelelő mozgásokat), az eredmény fokokban értelmezendő.

	Pozitív irány	Működési tartomány	Optimális tartomány
Bólintás	Felfelé	20°	10°
Döntés	Bal váll felé	90°	45°
Elfordulás	Baloldalra	45°	30°

1. táblázat A fej különböző mozgásai



21. ábra A fej különböző mozgásai [18]

A Candide3 modell alapján a száj nyitottsága, az ajkak összeszorításának mértéke, a szemöldökök helyzete, és még néhány dinamikus paraméter szintén egyből a programozó rendelkezésére áll, éppúgy, mint a szem méreteire, távolságára, az orr pozíciójára, stb. vonatkozó statikus paraméterek. Ezen statikus paraméterek alkalmasak arra, hogy a kamera elé később visszatérő személyt arcvonásai alapján felismerhesse a rendszer.

Ezen kívül – szintén a Candide3 modell alapján – egy háromdimenziós modell is megalkotható az arcról, bár ez a modell nem jön létre automatikusan az arckövető függvények hívásakor, hanem a korábban említett dinamikus és statikus Candide3 paraméterekből számítható, az API tartalmazza az ehhez szükséges függvényeket is. [26]

2.6. Kinect tesztprogram

A projekt megvalósítása során a Kinect csontvázkövetés funkciója a legkézenfekvőbb megoldás. A Kinect kapcsolódó funkcióinak feltérképezésére egy tesztprogramot írtam, mely egy csontvázat, és annak egy adott testrészét monitorozza, az eredményeket vagy szöveges módban, vagy a testrész pozícióját grafikus úton jeleníti meg.

A Kinect for Windows SDK használatához a Visual Studio program projektjének beállításain néhány változtatást kell alkalmazni. Egyrészt a megfelelő header fájlok eléréséhez az „Include Directories” mezőhöz hozzá kell adni a \$(KINECTSDK10_DIR)\inc bejegyzést. Másrészt az SDK programkönyvtárát is láthatóvá kell tenni a linker számára, így az „Additional Dependencies” mezőhöz hozzá kell adni a \$(KINECTSDK10_DIR)\lib\x86\Kinect10.lib fájlt.

A Natural User Interface használatához ki kell adni az `#include <NuiApi.h>` fordítói direktívát. A `Windows.h` header fájljal hasonlóképpen kell eljárni (akkor is, ha csak interfészen belüli függvények szerepelnek), egyébként a fordítás sikertelen lesz.

A program először a `kinect.config` fájlból konfigurációs adatokat olvas ki. Minden indulás alkalmával bekérni ezen adatokat hosszadalmas procedúra volna, forráskódban történő rögzítésük pedig a gyakori módosítások miatt a fordítási idő miatt körülményes, ezért hoztam létre a konfigurációs fájlt. Mivel az adatok beolvasása a Kinect programozása szempontjából nem releváns, így ennek lépéseit nem részletezem.

A szenzor használatához először inicializálni kell azt. Több szenzor is használható egy programmal, azonban egy készüléket csak egy program használhat. Éppen ezért az inicializálás erőforrásként le is foglalja az eszközt, így más program ezután már nem képes végrehajtani rajta az inicializációt, míg fel nem szabadítja az előző program. Amennyiben csak egy Kinectet használunk a rendszerben, lehetőség lenne *INuiSensor* objektum nélkül megoldani a feladatot, azonban számomra átláthatóbb kódot eredményezett az objektumhoz rendelés, és az IntelliSense használatát is megkönnyítette. Éppen ezért az inicializálás előtt a *NuiCreateSensorByIndex* (0,&szenzor) függvény segítségével a „szenzor” változóhoz rendeltem a 0. azonosítójú eszközt. Ezt követhette a *szenzor->NuiInitialize* (*NUI_INITIALIZE_FLAG_USES_SKELETON*) függvény, mely az inicializációt végrehajtotta. Az argumentumként

megadott flag biztosítja, hogy a csontvázkövetést a későbbiekben használhassuk. Mindkét függvény hibakóddal történő visszatérése megszakítja a program működését. A program végén az erőforrást a *NuiShutdown* függvény szabadítja fel.

Az inicializálás után a program ellenőrzi a fej elevációs szögét a *NuiCameraElevationGetAngle* függvény segítségével. Amennyiben eltérés mutatkozik a kívánt értékkel, a *NuiCameraElevationSetAngle* függvény segítségével az új értéket állítja be az eszközön.

Ezután ciklikusan elemzi a Kinect által szolgáltatott adatokat. Először a *NuiSkeletonGetNextFrame(0,&csontvazallapot)* lekéri a csontvázak aktuális állapotát. Itt az első paraméter egy késleltetés, csak ennyi idő után fogunk állapotleírást kapni. Természetesen a paraméter 0 értéke esetén is meg kell várnunk az adatok rendelkezésére állásának idejét. A *NUI_SKELETON_FRAME* típusú csontvazallapot nevű változóban tárolódnak az adatok. Az objektum *SkeletonData* tömbjében az egyes csontvázak *NUI_SKELETON_DATA* típusú objektumban találhatók. Ezen objektum *eTrackingState* tagváltozója *NUI_SKELETON_TRACKED* értékű, amennyiben az adott csontváz aktívan követett, egy ciklusban a program kiválasztja a tömb első ilyen objektumát, és a későbbiekben ezzel dolgozik.

A csontváz koordinátáit a *Position.x*, *Position.y*, és *Position.z* tartalmazzák, a választott testrészét pedig a *SkeletonPositions[kovetendoTestreszID].x*, *.y* és *.z* változók. Szöveges módban ezen adatok a konzol képernyőjére kiírásra kerülnek. Grafikus módban a testrész *x* és *y* koordinátája alapján 24 sor magas és 80 oszlop széles területen az értékek megfelelő átskálázásával egy X-et rajzol ki a program, míg a 25. sorban > jelek mennyisége jelzi a *z* koordináta értékét. Egy adatgyűjtő mód is része a programnak, ekkor a mért értékek egy .m kiterjesztésű fájlba kerülnek, a Matlab program által értelmezhető formában, a későbbi kiértékelés megkönnyítése érdekében. A mért értékeken kívül egy referenciaérték is bekerül a fájlba, ezt minden egyes mérési pont előtt a program bekéri a felhasználótól, ez biztosítja azt is, hogy az egyes mérési pontok csak a felhasználó beavatkozására kerüljenek rögzítésre.

3. A Kinect pontossága

3.1. A mérés leírása

Mivel a Kinectet eredetileg szórakoztatóelektronikai eszköznek tervezték, jogosan merül fel a kérdés, pontossága megfelelő-e komolyabb alkalmazások számára. A Kinect legfőbb erénye az emberalakok automatikus detektálása, ráadásul a projekt megvalósítása során is ezt a funkciót használok ki, éppen ezért pontosságának ellenőrzése is emberi alakok felismerése közben a legcélravezetőbb. Könnyen belátható, hogy élő emberrel elvégezni a mérést egyrészt körülményes volna, másrészt lehetetlen volna biztosítani a referenciatávolságok pontos betartását az állandó mozgólódás miatt. Éppen ezért szükség van egy próbababára, melyet a Kinect emberalakként ismer fel, azonban a mérés szempontjából hátrányos emberi tulajdonságokat nélkülözi. Egy ruhaboltbeli baba vélhetően tökéletesen megfelelt volna erre a célra, azonban ez nem állt rendelkezésre, építeni kellett egy saját mérőbábut, mely a 22. ábrán látható.



22. ábra Mérőbábu

A mérési elrendezésben a számítógéphez kapcsolt Kinect előtti térben helyezkedett el a próbababa, a méréshez szükséges referenciatávolságokat egy mérőszalag segítségével lehetett beállítani. A mérőszalag a 23. ábrán a mélységi mérésekhez van beállítva, az x koordináta mérésénél természetesen merőleges irányban volt kifeszítve.



23. ábra Mérési elrendezés

A Kinect y koordinátájának pontossága nem képezte vizsgálat tárgyát, ugyanis túlságosan bonyolult feladat lett volna a baba függőleges irányba történő mozgatása – ráadásul úgy, hogy a referenciaértékek pontosan beállíthatók legyenek. Szerencsére a próbababát emberi alakként detektálta a műszer, azonban néha a felismerés sikertelen volt, ami megnehezítette a mérést. Az érzékelési tartományok határán a mérés megkezdése szinte lehetetlen volt, a babát egy alkalmasabb pozícióban fel kellett ismertetni a rendszerrel (olykor a karok mozgatása is szükséges volt ehhez), majd csak ezt követően lehetett a tartomány határához kitolva megkezdni az adatgyűjtést. További problémát jelentett, hogy a mérés során az eszköz mindvégig a próbababa helyzetét detektálja, az esetlegesen pozicionálás céljából feltűnő egyéb emberalakok ne zavarják össze a mérést. Ennek a kritériumnak a mérések jelentős részénél csak úgy lehetett eleget tenni, hogy a pozicionálást végző személy mindvégig a próbababa mögött, annak takarásában tartózkodott.

3.2. Mérés egyhelyben

A méréssorozat kezdéseként a bábút egy-egy rögzített pozícióban hagyva mértem a pozícióját. A mérési eredmények a Kinect koordinátarendszere szerint értelmezendők, azaz az x koordináta az oldalirányú-, az y koordináta a függőleges elmozdulást, míg a z koordináta a mélységi információt hordozza. Mindkét mérési pont a készülék előtt, a kamerák tengelyével egyvonalban helyezkedett el, az egyik 2 m, míg a másik 3,5 m távolságban. A csípő közepét jelző pont (azaz a fő referenciapont) adatait lekérve, a

mérési adatokat kiértékelve a 2. táblázatban látható eredmények adódtak eredményül az egyes koordináták szórására (az eredmények normális eloszlást mutattak).

	2 m-es távolságban	3,5 m-es távolságban
x koordináta	0,45 mm	2,0 mm
y koordináta	0,61 mm	2,4 mm
z koordináta	0,23 mm	0,53 mm

2. táblázat Szórások (csípő közepe)

Amennyiben a csípő közepe helyett a fej pozícióját jelző pont helyzetét vizsgáltuk, abban az esetben a szórások a 3. táblázat szerintiek.

	2 m-es távolságban	3,5 m-es távolságban
x koordináta	1,2 mm	7,2 mm
y koordináta	1,5 mm	3,6 mm
z koordináta	1,8 mm	10,6 mm

3. táblázat Szórások (fej)

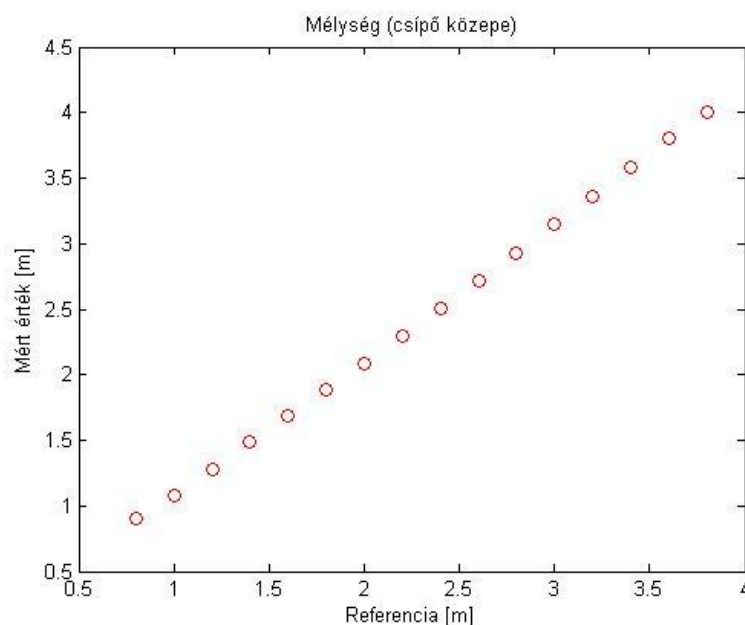
A szórások milliméteres, vagy annál kisebb nagyságrendbe estek. Egyedül a fej mérése esetén, 3,5 m-es távolságban a z koordináta mért eredményeinek szórása éri el a centiméteres nagyságrendet, azonban egy szórakoztatóelektronikai eszköztől lévén szó ez az eredmény is kiválónak mondható.

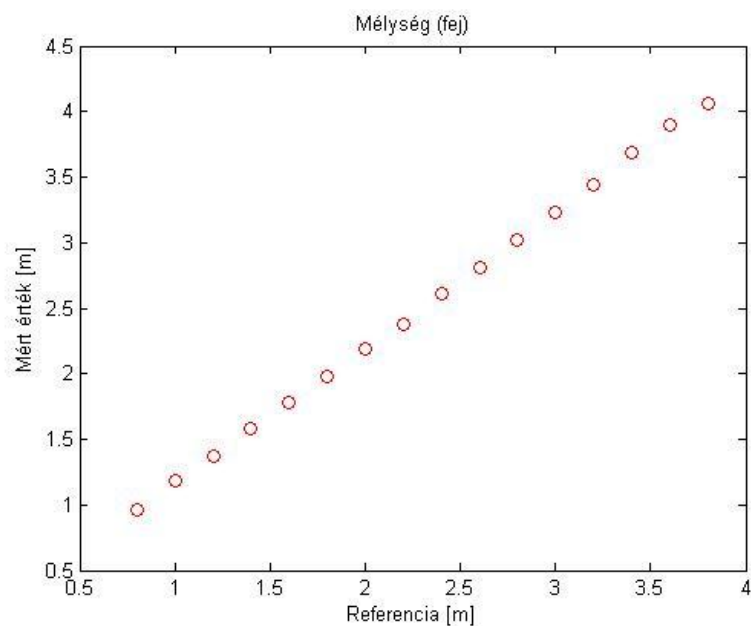
Érdemes megfigyelni, hogy a 3,5 m-es távolságban elvégzett mérések minden esetben rosszabb eredményt adtak a 2 m-nél elhelyezett mérőponthoz képest. Ennek az lehet a magyarázata, hogy a kamerákkal történő észlelés miatt a közelebbi objektumok felbontása nagyobb, hiszen a kamera egyes pixeleire eső fénynyalábok a tér egy adott szögtartományából érkeznek a kamerába, attól távolodva pedig a pixelhez tartozó szögtartomány alatt látott terület nagysága folyamatosan nő.

Az egyhelyben történt mérések – bár bizonyos szempontból ezek is tanulságosak – tulajdonképpen a további mérések pontossági követelményeinek megállapítására szolgáltak. Ugyanis ebben az esetben a mért értékek egymástól való eltérései egyedül a Kinectnek tudhatók be, a mért környezet a mért adatok felvétele között nem változott. Éppen ezért az ezt követő mérések során legalább ekkora pontosságot kell elérni, hogy a bábu különböző mérési pontokra történő pozicionálásának hibája ne zavarja meg a mérést. Ezt csak részben sikerült teljesíteni – a másfél méteres próbababát minden igyekezetem ellenére lehetetlen lett volna 10^{-4} m-es pontossággal pozicionálni, azonban a milliméteres pontossági kritériumot sikerült teljesíteni. Ezt figyelembe kell venni a további mérések értékelésénél.

3.3. Mélységi mérések

A Kinect képes az előtte lévő objektumok készüléktől mérhető távolságát rögzíteni, ez a funkció a csontvázkövetés esetében is megvalósul, így méréssorozatomat ennek vizsgálatával folytattam. A mérés során a mérőbaba az eszköz előtti vonalban, a Kinect dokumentációjában rögzített 80 cm és 4 m közötti optimális érzékelési tartományban mozgott, és 20 cm-es közönként mértem le a baba pozícióját a Kinect segítségével. Bár a dokumentáció 4 méteres távolságig határozza meg az optimális tartományt, ebben a mérési pontban már nem tudtam rögzíteni eredményt, minden esetben 3,8 m volt az utolsó, még felvett érték. A mérést a csípő közepe és fej referenciaponttal is elvégeztem, a 24. ábrán látható eredmények adódtak.





24. ábra Mélységi adatok

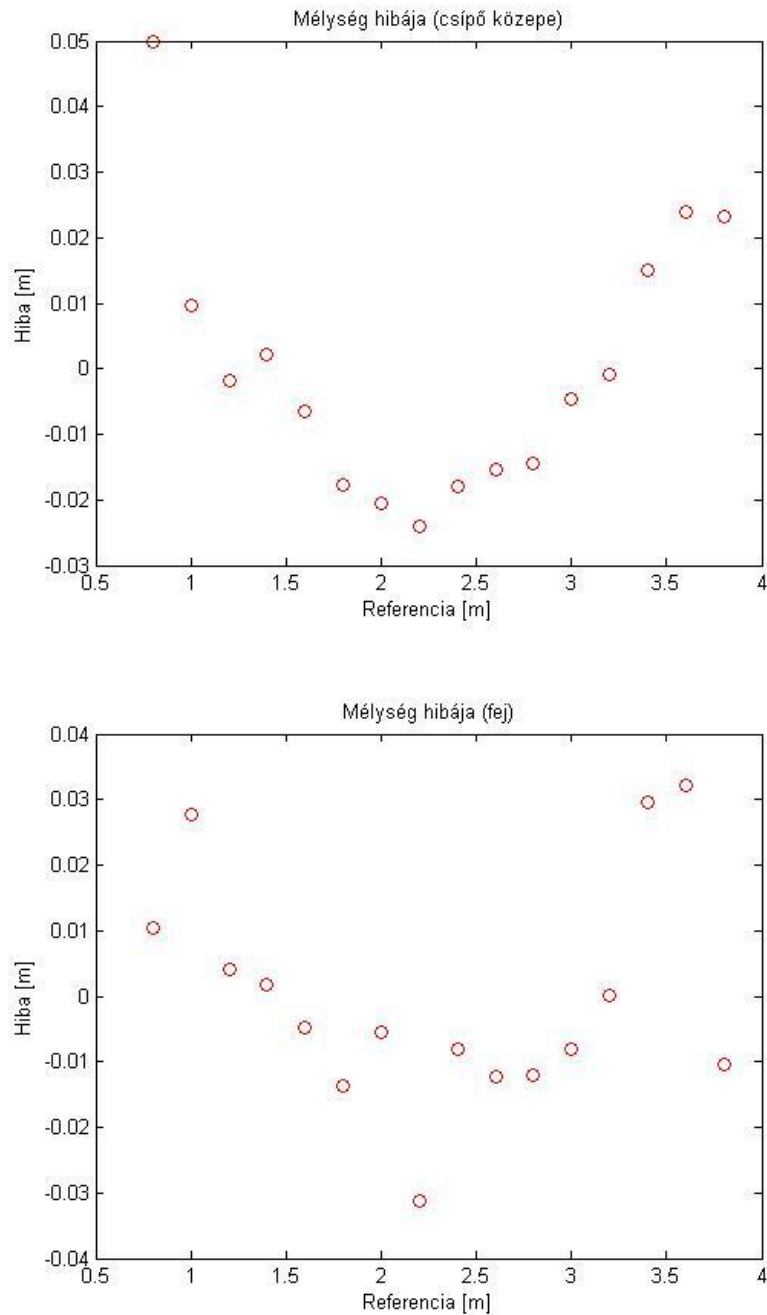
Rájuk nézve, mindkét grafikon a várakozásoknak megfelelő, lineáris jelleget mutatja. A legkisebb négyzetek módszerével illesztett egyenesek a 4. táblázatban összefoglalt paraméterekkel rendelkeznek.

	Csípő közepe	Fej
Ofszet	0,0255 m	0,1197 m
Meredekség	1,0419	1,0395

4. táblázat Mélységmérés illesztett egyenseinek paraméterei

Érdekes, hogy a fej mérése esetén az ofszet majdnem 10 centiméterrel nagyobb, azonban ez vélhetően annak köszönhető, hogy a papírból készített fejnek mélységi kiterjedése (az orrot leszámítva) tulajdonképpen nincs, a ruhakeretre történő rögzítésnek köszönhetően pedig a kabát elülső vonalához, így a detektált csípő közepe ponthoz képest ennyivel hátrébb helyezkedett el. A meredekség mindkét mért pont esetében közel 1, ami megnyugvásra adhat okot, hiszen ez azt jelenti, hogy a Kinect az általa mért paramétereket megfelelően skálázza méter egységekbe.

Vizsgáljuk meg a mért értékek illesztett egyenestől való eltérését, melyet a 25. ábra mutat be.

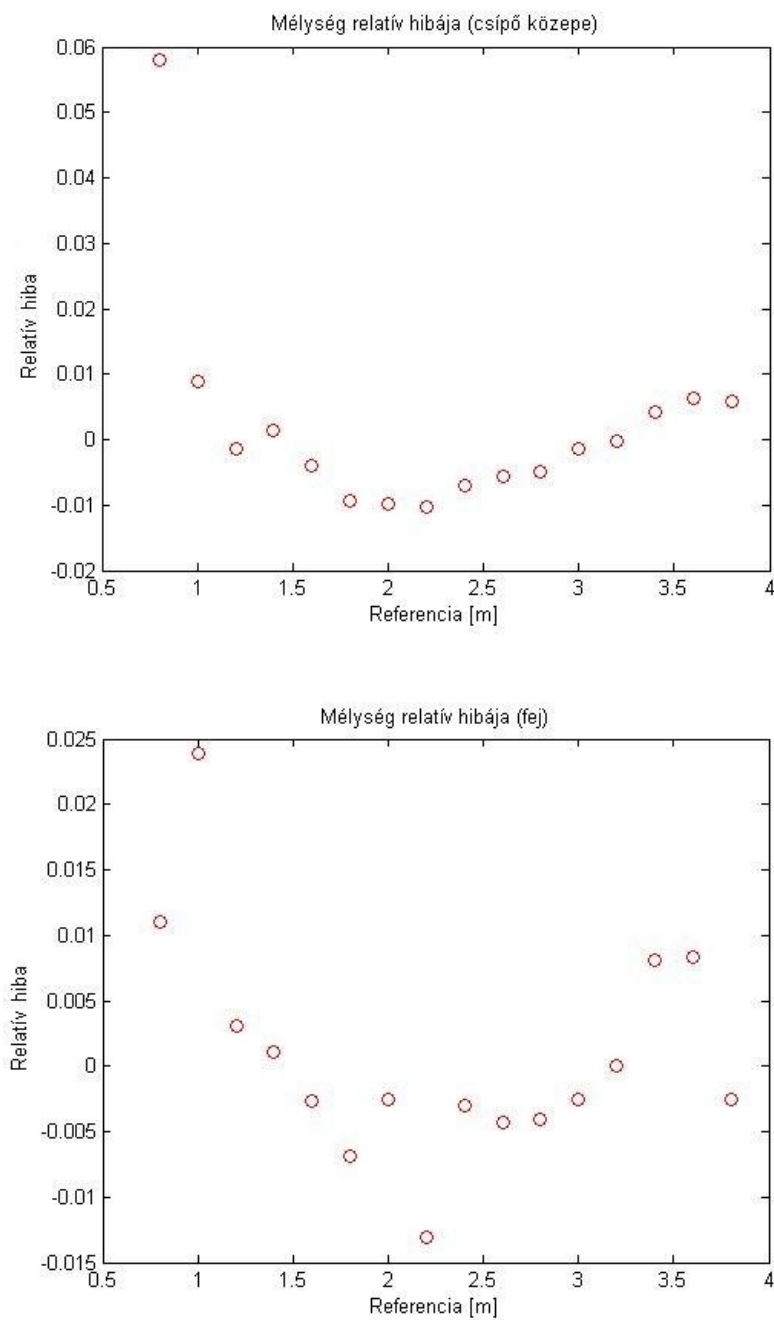


25. ábra Mélységmérés hibái

A csípő közepe esetén a legpontatlanabb a 0,8 m-es referenciapontban felvett érték, azonban ez is mindösszesen 5 centiméterrel tér el az illesztett egyenestől. Nem szabad megfeledkeznünk arról sem, hogy ez a mérési pont az optimális tartomány széle, így némileg pontatlanabb eredmények itt megengedhetők.

Bár a Kinect legtöbb alkalmazása esetében az abszolút hiba fontosabb paraméter, hiszen a készüléktől mérhető távolságunktól függetlenül várunk el egy bizonyos

pontosságot, vizsgáljuk meg a mélységérzékelés relatív hibáját is, mely a 26. ábrán látható.



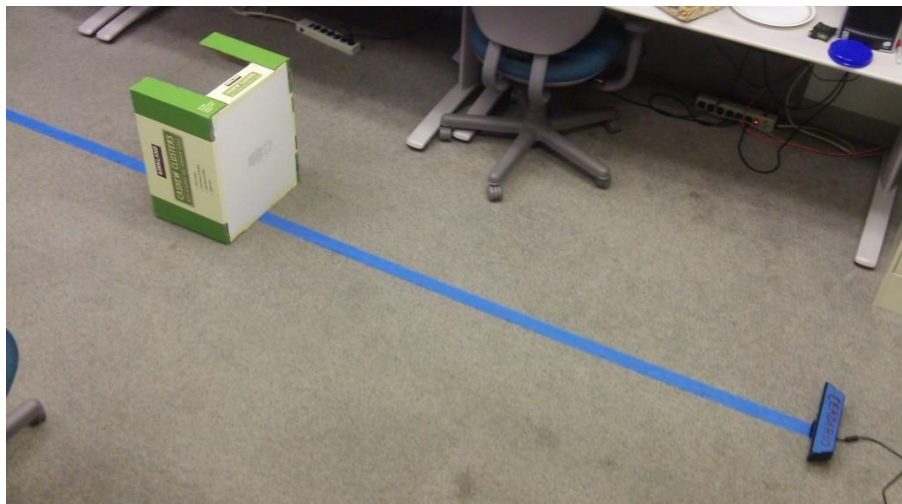
26. ábra Mélységmérés relatív hibái

Megfigyelhető, hogy a korábban már említett 80 centiméternél felvett értéket, vagy a fej mérésénél az 1 m távolságban mért értéket leszámítva minden eredmény 1%-os pontosságon belül van, vagy éppen csak túllépi azt.

Továbbá az is érdekes, hogy mindkét esetben a mérési tartomány szélein a mért értékek az illesztett egyenes megfelelő pontjaihoz képest nagyobbak, míg a mérési tartomány közepén, 2 méter környékén kisebbek.

3.4. Nyers mélységi adatfolyam

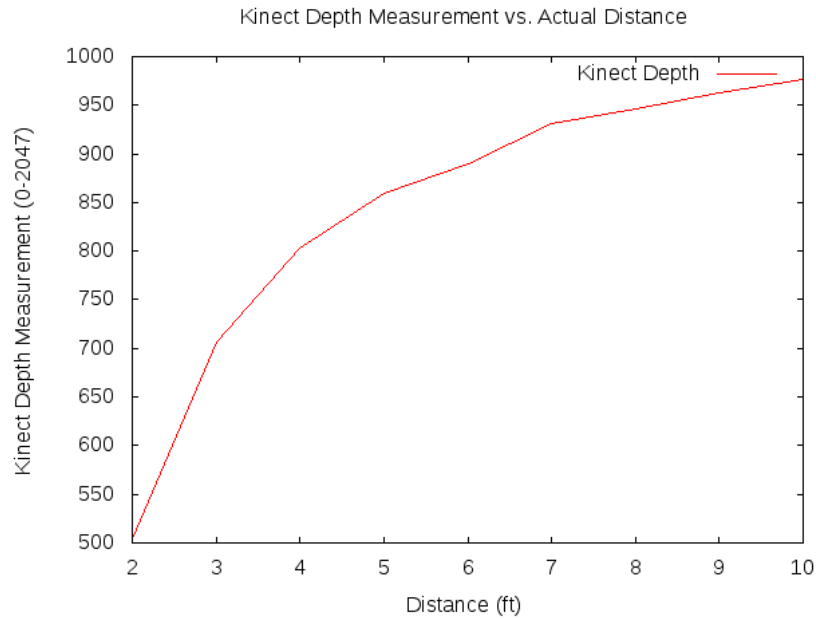
Nathan Crock emberi alakok vizsgálatát helyett a Kinect nyers mélységi adatfolyamát vette górcső alá. A 27. ábrán látható mérési elrendezés nem sokban különbözött az általam felállítottól, a legmarkánsabb különbség az, hogy a mérőbábut egy kartondoboz váltja fel, melynek sima hátlapja kiválóan alkalmas a mélységi mérések lefolytatásához. [27]



27. ábra Nathan Crock mérési elrendezése [27]

A nyers adatfolyam 11 bites értékeket tartalmaz, amivel elvileg 2048 db különböző távolságérték lenne reprezentálható. Crock 2 és 10 láb között lábanként (körülbelül 30 cm) vett fel mérési pontokat. [27]

A 28. ábrán jól látható, hogy a nyers adatfolyam egyáltalán nem lineáris, inkább logaritmikus jelleget mutat. Ez illeszkedik ahhoz a korábban már bemutatott jelenséghez, hogy a távolabbi objektumok felbontása kisebb, ezért pozíciójuk csak pontatlanabban mérhető ebben a tartományban, így ideálisabbnak tűnik a logaritmikus skála, amely a közelebb lévő és pontosabban detektált objektumokhoz több, míg a távolabbi és ezért kevésbé pontosan detektált objektumokhoz kevesebb különböző távolságértéket rendel.



28. ábra Nathan Crock mérési eredménye [27]

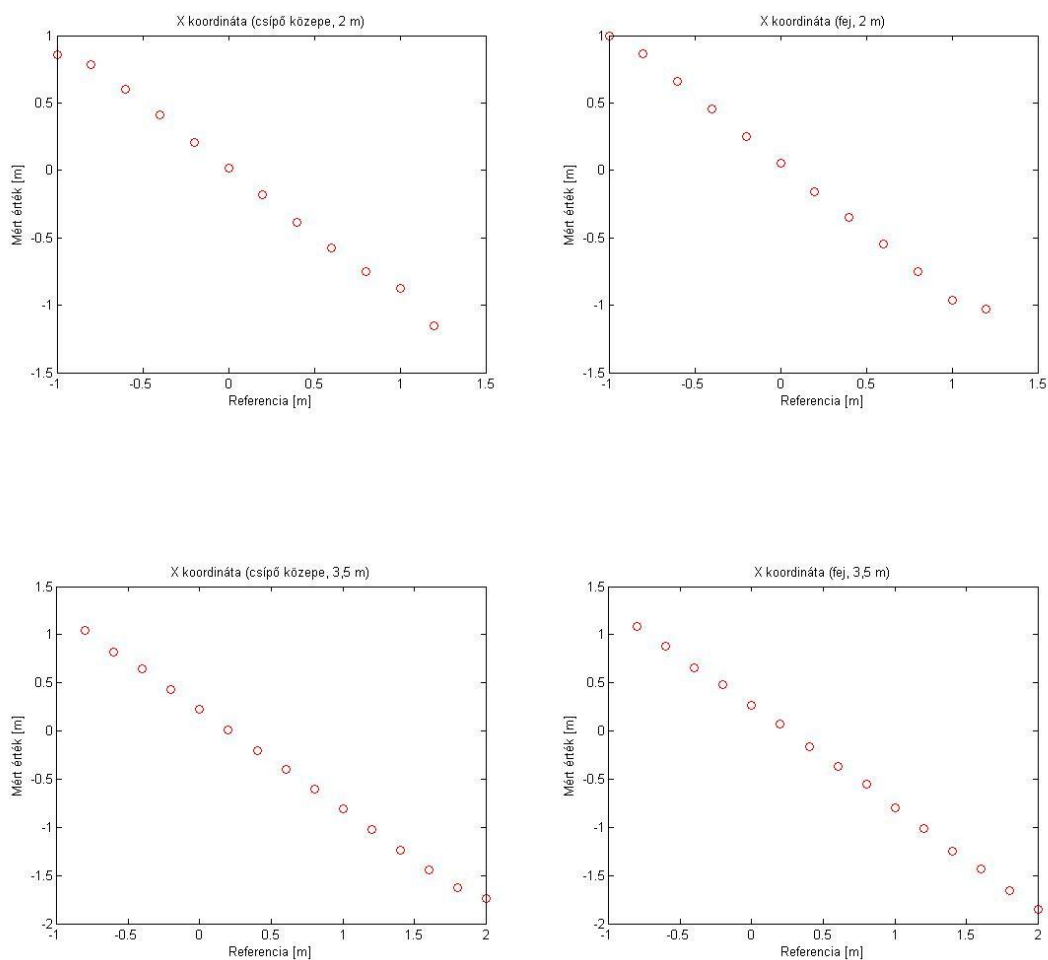
Meglepő, hogy 2 lábhoz (ami már az optimálisan érzékelhető tartomány határánál közelebb, de még az értelmezhető tartományon belül van) már az 504-es értéket rendeli a szenzor. 10 lábhoz pedig a 976-os érték társul – ha feltételezzük, hogy a logaritmikus skála lineárisává válik, még akkor sem érnék el az értelmezhető tartomány felső határában, azaz 8 méteres távolságban a 11 bitből számítható 2047-es utolsó távolságértéket.

3.5. Oldalirányú mérések

A mélység érzékelésének pontossága után vegyük szemügyre az oldalirányú elmozdulások detektálásának kérdését. Két méréssorozatot folytattam le: először a kamerától 2 méter, majd 3 és fél méter távolságban helyeztem el a kamera irányára merőlegesen a mérőszalagot, és ennek mentén mozgott a bábu. Mindkét esetben a csípő közepének, majd a fejnek a pozícióját is megmértem.

A közelebbi és a távolabbi mérésnél is a kamera tengelyétől egy méterre jobbról (a kamera felé fordulva) kezdtem a mérést, majd 20 centiméterenként haladtam a másik irányba, egészen addig, ameddig a csontvázkövetés eredményt adott. Ez a 2 méterre lévő mérések esetében 1,2 méterre balra a kamera tengelyétől, míg a 3,5 méterre lévő mérések esetében 2 méternél jelentette az utolsó mérési pontot, így a közelebbi

mérésnél 12, míg a távolabbinál 16 érték került rögzítésre. A mérési eredményekről készült grafikonokat a 29. ábra tartalmazza.



29. ábra x koordináta mérési eredményei

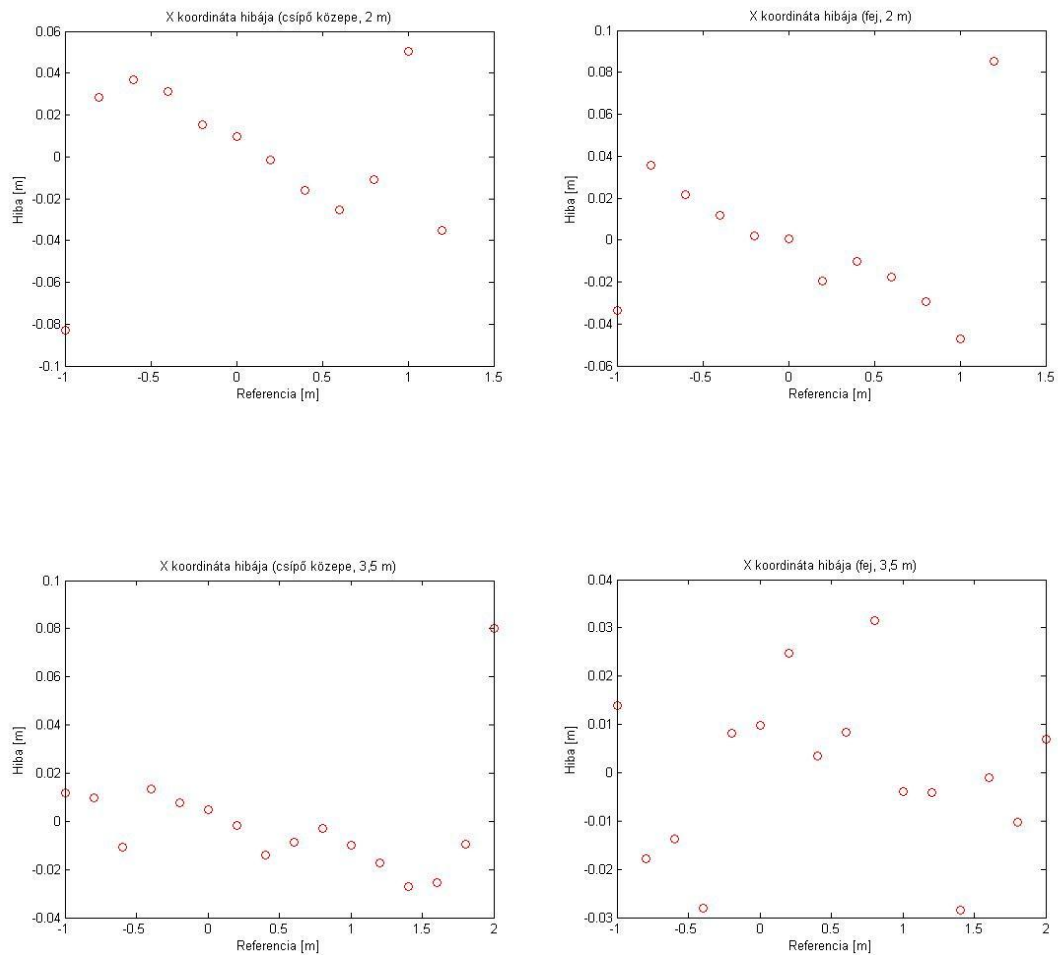
A későbbiekben az eredményekre illesztett egyenesek negatív meredeksége annak köszönhető, hogy a mérés közben alkalmazott referenciaértékek pozitív iránya pont fordítva lett meghatározva a Kinect koordinátarendszeréhez képest. A meghatározott meredekségek az 5. táblázatban olvashatók.

	Csípő közepe	Fej
2 méteres távolságban	-0,9347	-0,9722
3,5 méteres távolságban	-1,0206	-1,0566

5. táblázat x koordináta mérése során illesztett egyenesek meredeksége

2 méteres távolságban a valóságosnál némileg kisebb oldalirányú távolságokat becsül a Kinect, ezzel szemben a 3,5 méternél elvégzett mérés esetében a távolságokat felülbecsülte.

Az illesztett egyenestől való eltéréseket a 30. ábra grafikonjai szemléltetik:



30. ábra x koordináta hibái

Jól megfigyelhető, hogy a kamera által belátott tartomány határához érve a mérés bizonytalanná válik. Ennek az az oka, hogy a csak részletében látott emberalakokat is feldolgozza a rendszer, azonban a nem látott részletek kiegészítése nem mindig megfelelő, így részben fedett emberalakok esetén az egyes testrészek pozícióinak detektálása teljesen kiszámíthatatlan.

3.6. Összegzés, a Kinect színpadi alkalmazásának korlátjai

Bár a projekt megvalósítását jelentősen megkönnyítette a Kinect alkalmazása, komoly hátrányokkal is számolnunk kell. A berendezés egy szórakoztatóelektronikai eszköz, ipari alkalmazhatóságával kapcsolatban kérdések merülnek fel, robusztussága, megbízhatósága valószínűleg nem elégítenék ki egy valódi alkalmazás követelményeit.

A Kinect pozíciódetektáló rendszere meglehetősen hatékonyan működik. A különböző irányokba történő mérések esetén a jellemző hibaértékek 1-2 cm körül mozogtak. Ez rendkívül precíz helymeghatározást tesz lehetővé, ráadásul közvetlenül lekérdezhető a színész fejének pozíciója, mely a projekt szempontjából kulcsfontosságú. Azonban az eszköz által érzékelt terület meglehetősen kicsi, mindössze egy közepes méretű szoba lefedésére alkalmas – az érzékelési tartomány elfér egy 4 m oldalélű kockában, ami még az olyan kisméretű színpadok méreteit sem fedi le, mint például a Centrál Színház Kisszínpada, mely körülbelül 8*5 méteres. Ez a korlát a maximálisan együtt kezelhető 4 db érzékelő csatlakoztatása esetén sem enyhül jelentősen, egy nagyobb terület lefedésének kérdése nem megoldott, rendkívül bonyolult probléma volna akár egy ilyen kis színpad megfelelő belátása is, nem beszélve arról, hogy a szenzorok megfelelő elhelyezése nem is biztos, hogy lehetséges volna. Ráadásul a több eszköz koordináta-rendszereinek illesztése nem támogatott, arról szintén gondoskodni kellene. [28]

További komoly probléma, hogy az alakzat-felismerés csak álló és – korlátozott mértékben – ülő emberalakokra, és a kamera felé nézve működik, színházi előadások alkalmával viszont gyakorta vesznek fel egyéb testhelyzetet is a szereplők. A szereplők azonosítása is nehézkes – ugyan a Face Tracking SDK segítségével lehetséges arcfelismerés, az érzékelési tartomány szélén megjelenő színészek azonnali azonosítása lehetetlen, ez csak a kamerához közelebb érve válna lehetővé, viszont a színre lépő színészeket késlekedés nélkül a megfelelő világításképpel kellene megvilágítani. Összességében elmondható, hogy a rendszer egészének működése remekül demonstrálható a Kinect segítségével, ipari felhasználásra azonban nem ajánlott.

4. Színpadvilágítás

4.1. A színpadvilágítás rövid története

Már a római színház esti előadásainál is alkalmaztak fáklyákat, olajlámpákat, sőt, attól sem riadtak vissza, hogy egy drámai jelenetben a látvány kedvéért olykor-olykor a díszletet is felgyújtsák. Azonban a római színház esetében még nem beszélhetünk a klasszikus értelemben vett színpadi világítástechnikáról, hiszen a fénysugarak irányításának, szabályozásának lehetősége ekkor még nem állt fent. Erre a 16. században került először sor – a fényt gyertyákkal állították elő, színezésüket különböző folyadékszűrőkkel oldották meg. A 17. századtól csillárokkal világították meg a színházakat. A nézőtérben a világítás folyamatos volt, a színpadot azonban el lehetett sötétíteni, ehhez a zsinórpadlásról leengedhető különböző szerkezeteket konstruáltak. A 19. században a gázlámpák hódítottak. Bár felforrósították a levegőt, és a gáz kellemetlen szaga is érezhető volt a levegőben, mégis előszeretettel alkalmazták, ugyanis a gáz áramlási sebességének változtatásával könnyen lehetett a fényerőt változtatni. A lámpákhoz futó gázcsövek egy helyen futottak össze, ahol szelepekkel lehetett állítani a különböző lámpákhoz eljutó gáz mennyiségét – tulajdonképpen ez volt az első „fényvezérlőpult”. Az elektromossággal begyújtható gázlámpák újabb áttörést hoztak, hiszen mindezidáig a fényforrások bekapcsolása komoly nehézségekbe ütközött. [29]

A 19. sz. második felétől ívlámpát használnak különleges megvilágításokhoz, így például a párizsi operában a fejtékeket üzemeltették ennek segítségével. Az izzószálas elektromos színpadvilágítás elterjedése sokkal biztonságosabbá teszi a színpadot – nem véletlenül alakult ki a színpadot a nézőtértől elválasztó tűzzáró vasfüggöny, korábban sok tüzesetet okozott a veszélyes színpadvilágítás. Kizárólag elektromos világítást először 1881-ben a Savoy Színházban alkalmaznak, bár a színház átadásakor még gázlámpás világítás működött, ugyanis elővigyázatosságból ez a rendszer is beépítésre került. Az év végétől kezdve azonban nem használták tovább a gázlámpákat.

A fényforrások és az optika tovább fejlődött. Gondot okozott a nehezen hozzáférhető helyen lévő lámpák színének módosítása, így automatikus színváltókat kezdtek használni, majd megjelentek az intelligens fényvetők, melyek a megvilágítás számos paraméterét képesek automatikusan, távoli vezérlés hatására beállítani. Ehhez

különböző sötétítő egységeket, szintárcsákat, gobókat (ábrát tartalmazó maszkokat), íriszblendéket és nagyító lencsét tartalmaznak, melyeket léptető motorok állítanak a megfelelő pozícióba. Intelligens fényvetőkkel kialakított fényjáték látható a 31. ábrán.



31. ábra Fényjáték intelligens fényvetőkkel

Az intelligens fényvetőknek két főbb típusa létezik, melyek a fénycsóva iránybeállításának módjában térnek el, mindkettőre látható példa a 32. ábrán. Először a mozgótükrös fényvetők jelentek meg. Itt a lámpatestből kilépő fény egy tükorről verődik vissza a színpadra, ennek a tükörnek a forgatásával lehet a fénynyalábot irányítani. A konstrukció előnye, hogy a kis tükröt könnyű mozgatni, így gyors beállási idő érhető el, azonban hátránya a tükör tökéletlen reflexiója miatti fényintenzitás-csökkenés, valamint a bevilágítható térszögtartomány korlátossága. A lámpák általában 150° -ot képesek átfogni vízszintesen, 110° -ot függőlegesen.

Különösképpen ez utóbbi hátrány kiküszöbölésére jöttek létre a mozgófejes fényvetők. Ebben az esetben a lámpafej egészének orientációja változik meg a talphoz képest, így függőlegesen 250° a szokásos szögtartomány, míg vízszintesen 450° , így teljesen körbe tudnak fordulni. Vannak robotlámpák, melyeknél úgy alakították ki a mechanikát, hogy vízszintesen a fenti szögtartomány sem legyen korlát, és a tengelyük körül akárhányszor körbe tudnak fordulni. További előny a mozgótükrös fényvetőkhöz képest, hogy a tükör hiánya miatt, annak reflexiós tényezőjével sem kell számolni. Legnagyobb hátránya a mozgófejes kivitelnek, hogy az egész lámpafej mozgatása miatt jóval nagyobb motorokat kell alkalmazni, melyek egyrészt zajosabbak, másrészt még így is csak mintegy tized akkora sebesség érhető el a tükör mozgatásához képest. Éppen ezért mindkét konstrukció megtalálható a piacon, bár a mozgófejes változatok egyre csendesebbek és gyorsabbak, így kezdenek előtérbe kerülni. [29]



32. ábra Mozgótükrös- és mozgófejes fényvető

4.2. Színpadi vezérléstechnika

A vezérlés igénye a kezdetekkor a fényerőszabályzásban merült ki, a ma használatos rendszerek is magukon hordozzák az ebből eredő korlátokat. A lámpák teljesítményszabályzása úgynevezett dimmerek segítségével lehetséges. Ezek olyan eszközök, melyek a vezérlőjelek függvényében az egyes lámpákhoz a megfelelő villamos teljesítményt képesek juttatni. Leggyakrabban triakkal fázishasítást alkalmaznak, illetve újabban az IGBT segítségével megvalósított váltóáramú szaggatók is egyre elterjedtebbek.

A dimmerek vezérlése analóg illetve digitális úton is történhet, bár az analóg megoldás szinte teljesen eltűnt. Ennek oka, hogy analóg vezérlés esetén az egyes lámpacsoportokhoz tartozó információt egy feszültség szint közvetíti, így minden egyes lámpacsoport vezérléséhez külön ér szükséges, ami megnehezíti a kábelezést, ráadásul a környezeti zavarok is gondot okozhatnak. Digitális vezérlés esetén, egyetlen érpáron átvihető az összes eszköz üzemeltetéséhez szükséges információ. [29]

Mindmáig a DMX512 a legelterjedtebb szabvány a színpadi világítástechnika körében, és többnyire az új fejlesztések is ehhez a szabványhoz nyúlnak vissza. Az USITT (The United States Institute of Theatre Technology) 1986-ban fejlesztette ki a protokollt. Nevét a „Digital Multiplex” kifejezés alapján kapta, ezzel is utalva arra a korábbi megállapításra, hogy több lámpacsoport jele együtt, digitálisan kerül átvitelre. Az 512 arra utal, hogy ennyi csatornát, azaz alapesetben ennyi lámpacsoportot képes irányítani. A robotlámpák megjelenésekor meg kívánták tartani a már bevált protokollt, viszont az intelligens fényvetők esetében a fényerősségen kívül rengeteg egyéb paraméter állítható, melyekhez mind-mind külön csatornát kell rendelni. Ráadásul egy

csatornán csupán 8 bit információ kerül átvitelre, ami például a korszerű robotlámpák pozícióállításának pontosságigényét nem elégíti ki, ilyen esetekben egy funkcióhoz is felhasználnak két csatornát. A fentiek figyelembevételével egy-egy ilyen lámpa akár 30 csatornát is lefoglalhat. Ebben az esetben az 512 maximális csatorna korlátot jelenthet, ekkor a vezérlőpultból több, független DMX vonalat szoktak indítani. [30]

A DMX512 az EIA/TIA-485 szabványon (sokszor RS-485-ként emlegetik) alapszik. Ez egy aszinkron, szimmetrikus kommunikációs protokoll. Egy master egység lehet, az abból kiinduló vonalra kapcsolódnak a slave eszközök, a vonal végét a reflexiók elkerülése végett le kell zárni – a leggyakoribb megoldás az, hogy a két szimmetrikus jelvezeték közé 120 ohmos ellenállást kötnek. A DMX512 az EIA/TIA-485-nél szigorúbb előírásokat állít fel a vonal hosszával kapcsolatban, csupán 1000 lábat (~300 m) engedélyez, a vevők között 300 láb távolság lehet. Összesen 32 eszköz kapcsolódhat a vonalra. [30] [31]

Az előírt kábel az RS-485 szerinti, ezen kívül a DMX512 specifikálja a csatlakozók típusát. Az ötlábú XLR csatlakozóra esett a választás, melyből csak három láb használt, a maradék kettő fenntartott későbbi felhasználás céljára – például visszafele irányuló kommunikáció valósítható meg itt (üzemidő, hőmérsékleti adatok lekérése, stb.). Mivel két láb a csatlakozón nem használt, akadnak megoldások, melyek a háromlábú XLR csatlakozót alkalmazzák, ezek azonban nem kompatibilisek a szabványos eszközökkel. [32]

Lábszám	Jel
1	Föld
2	Adat -
3	Adat +
4	Fenntartott (Adat -)
5	Fenntartott (Adat +)

6. táblázat XLR csatlakozó lábkiosztása [32]

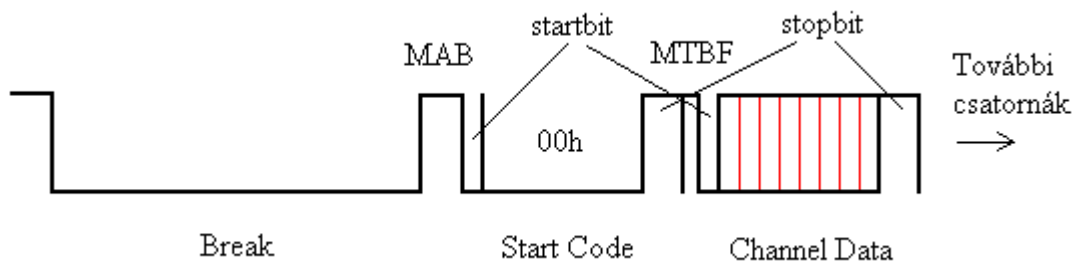
A szabvány által előírt jelzési sebesség 250 kbaud, azaz a szimbólumidő 4 μ s. Az adatátvitel során alkalmazott protokoll meglehetősen hasonlít az RS-485 szabvány által előírtéhoz, azonban a DMX esetében egy multiframe struktúra is megtalálható az egyes csatornák adatainak átviteléhez. A keretszervezést egy speciális kezdőbájt indítja, majd sorban következnek az egyes csatornák (legfeljebb 512). Nincs szükség csatornánkénti címzésre, ráadásul nem kötelező elküldeni az összes csatorna konfigurációját, bármelyik után befejezhető az átvitel, ezáltal jelentősen növekszik az adatsebesség. Az egyes bájtok átvitele egy startbittel (0) kezdődik, ezután nyolc adatbit következik, melyek közül az LSB kerül először küldésre, legutoljára az MSB. Paritásbit nincs, a keretet két darab stopbit (1) zárja. A paritásbit hiánya a felhasználást tekintve nem okoz különösebb problémát, hiszen az esetek döntő többségében annyira gyorsan küldjük újra a csatornainformációkat, hogy egy esetleges hibás érték hatását észre sem lehet venni. Azonban az átvitel megbízhatatlansága miatt a szabvány kifejezetten tiltja alkalmazását olyan esetekben, ahol a hibás átvitel sérüléseket okozhat (például színpadgépészet, pirotechnika). [30]

A csatorna inicializálására a Break jelzés szolgál. Ez legalább 88 μ s-ig tartó alacsony jelszintet jelent. Ezt követi a Mark After Break (MAB), mely jelzi a vevőkészülékek számára a soron következő multiframe indulását. Ez legalább 8 μ s magas jelszint. Ezután a fentebb leírt bájtviteli protokoll szerint küldi a DMX vezérlő a multiframe kezdőbájtját, mely dimmereknél szóló adatcsomag esetén 00h. Ezt a részt (a start- és stopbitekkel együtt) a szabvány Start Code (SC) névvel illeti. Jelen pillanatban minden DMX átvitel (a robotlámpáknak szóló is) a 00h kezdőbájtot alkalmazza, ugyanis nincs más definiálva. A DMX512-ből kialakult új szabványok (pl: RDM) viszont használnak más kezdőbájtokat, ugyanis így kompatibilitás érhető el a régi eszközökkel, hiszen ezen eszközök az új üzeneteket tartalmazó csomagokat az általuk hibásnak értelmezett kezdőbájt miatt eldobják. A Start Code után Channel Data (CD) blokkok következnek, melyben az egyes csatornák bájtojai kerülnek átvitelre. A Start Code után, illetve a Channel Data blokkok között Mark Time Between Frames (MTBF) szerepelhet, mely tulajdonképpen a stopbitek meghosszabbítása, legfeljebb egy másodpercig magas szinten maradhat a vonal. Az utolsó csatorna elküldése után, azaz a csomagok között is szerepel egy hasonló elem, a Mark Time Between Packets (MTBP). Az időzítések a 7. táblázatban foglalhatók össze. [32] [33]

	Tartalom	Méret (szimbólum)	Időtartam
Break	00..	22 – 250000	88 μ s – 1 s
Mark After Break	11(11..)	2 – 250000	8 μ s – 1 s
Start Code	0 00000000 11	11	44 μ s
Mark Time Between Frames	(11..)	0 – 250000	0 – 1 s
Channel Data	0 xxxxxxxx 11	11	44 μ s
Mark Time Between Packets	(11..)	0 – 250000	0 – 1 s

7. táblázat DMX512 keretszerkezése

A 33. ábra az eddigiek szemléltetésére szolgál.



33. ábra DMX512 jelalak [30]

4.3. Lámpavezérlés

A projekt során egy DTS Vinciscan intelligens fényvetőt bírtam működésre. A lámpa mozgótükros kivitelű, tartalmaz egy szín- és egy gobótárcsát. Ennek megfelelően a manapság használatos lámpákhoz képest nagyon kevés csatorna felhasználásával programozható, két csatornát foglal le a tükör irányítása, és egyet-egyet a két tárcsa, így csupán 4 csatorna szükségeltetik a működéséhez. Bár a fényvetőhöz nem állt rendelkezésre gépkönyv, és kora, illetve gyártásának megszűnése miatt az interneten sem lehetett ráakadni, a felderített csatornakiosztás a 8. táblázatban olvasható.

Csatorna	Funkció
1	Vízszintes mozgás
2	Függőleges mozgás
3	Színváltás
4	Gobóváltás, stroboszkóp effekt

8. táblázat DTS Vinciscan fényvető csatornakiosztása

A lámpán beállítható az első csatorna DMX vonalbeli ofsztetje, így a fenti csatornaszámok relatív értendők. Ugyancsak állítható a vízszintes és függőleges csatorna felcserélése, illetve a pozicionáló csatornákon belüli invertálás – ekkor az irányok megfordulnak. A vízszintes csatorna (invertálás nélkül, a lámpára felülről ránézve) a matematikai pozitív irány szerinti, azaz 0-s vezérlés esetén éri el a fénycsóva a teljes jobb-, míg 255 esetén a teljes bal kitérést. A függőleges eltérítés egyre nagyobb értékei esetén a fénycsóva egyre lejjebb kúszik. A szín- és gobócsatorna kiosztása a függelékben található.

A számítógéppel történő vezérléshez az FTDI USB to RS485 Serial Converter Cable termékét használtam. Ennek oka a DMX512 és RS-485 szabványok korábban már leírt hasonlósága. A kábel egyik fele egy USB csatlakozóban végződik, melyben egy lapka található, amely az USB-n kapott adatok alapján meghajtja az RS-485 vonalat. A kábel másik oldalán csatlakozó nélkül, szabadon hozzáférhetők az erek. Mivel a DMX szabvány XLR csatlakozókat használ, a kábelnek erre az oldalára ráforrasztottam egy megfelelő 5 lábú csatlakozót. A kábel ereinek és a csatlakozó lábainak összerendelését a 9. táblázatban foglalom össze, az elkészült kábel a 34. ábrán látható. [34]

Név	Szín	XLR lábszám
GND	Fekete	1
Terminator 1	Barna	-
POWER	Vörös	-

Data +	Narancs	3
Data -	Sárga	2
Terminator 2	Zöld	-

9. táblázat Az FTDI kábel ereinek és az XLR csatlakozó lábainak összerendelése



34. ábra USB-DMX kábel

Készítettem egy csatlakozót a vonal lezárásához is, a szokásos módon, a 2. és 3. láb közé beforrasztottam egy 120 ohmos ellenállást. Azonban ezt a lezárást használva, a lámpával történő kommunikáció sikertelen volt. Egy véletlen folytán rájöttem, hogy a saját konfigurációm akkor működik megfelelően, ha a lezáró ellenállás az 1. és 3. láb között található, ahogyan az a 35. ábrán látható.



35. ábra Csatlakozók

4.4. Lámpavezérlő tesztprogram

Az FTDI chippel a D2XX driver segítségével lehet kommunikálni, melyet az FTD2XX.DLL implementál. Ennek eléréséhez a linker számára hozzá kell adni az ftd2xx.lib könyvtárat, forrásainkhoz pedig csatolni kell az ftd2xx.h header fájlt. A driver függvények használnak a window.h-ban elérhető függvényeket is, éppen ezért ezt is hivatkozni kell. [35]

A lámpavezérlés tesztelésére megírt programom a DMX vonal inicializálásával indul. Először az *FT_Open(0,&ftHandle)* függvény meghívásával az *ftHandle* mutatóhoz rendeli az eszközt. A függvény visszatérési értéke egy státuszkód, melynek 0 értéke esetén volt sikeres a művelet. Ezt követően az *FT_SetBaudRate(ftHandle,250000)* függvénnyel a szabványban előírt 250 kbaud-ra állítja az átvitel sebességét. Végül a bájttáviteli paramétereket állítja be az *FT_SetDataCharacteristics(ftHandle,FT_BITS_8,FT_STOP_BITS_2,FT_PARITY_NONE)* függvény segítségével: 8 adatbit alkot egy keretet, két stopbit zárja a kereteket, nincs paritásbit. A kód végén az *FT_Close(ftHandle)* függvény szabadítja fel az erőforrást.

Két tömböt alkalmaz a program az adatok tárolására. A *TxBuffer* tömb mindig az aktuálisan lámpára kerülő adatsort tartalmazza, míg a *TxAlap* tömb referenciaértékeket tárol. Ennek a csatornák végigpásztázásánál van jelentősége. A tömb elemeinek típusa char, hiszen egy bájt hosszú adatokat kell a DMX vonalon kiadni. Mivel a referenciaértékek az egyszerű kezelhetőség érdekében szövegfájlba mentődnek, és a program indulásakor onnan is töltődnek vissza, több esetben is szükséges a karakter és szám típusok közötti oda-vissza kasszolás, mely bizonyos esetekben a típusok szélességének differenciája miatt konverziós problémákat vet fel, melynek leküzdésére bitmaszkolások szolgálnak. Bár a lámpa vezérlése csak négy csatornán történik, és a program funkcionalitása is ebben merül ki, a tömbök tartalmazzák a nem használt csatornákat is, ezeket a program alapesetben 0-val tölti fel, de a dmxertekek.txt fájlban felülírhatók, ugyanis induláskor a fájl tartalma alapján töltődnek fel a tömbök.

A tömbök inicializálása után egy egyszerű kezelőfelület generálódik, melyen egy karakter hosszú parancskódokkal lehet navigálni. A 'k' leütésével kilép a program. A 'd' elindít egy bemutatót, mely végigpásztázza csatornákat, így a lámpa összes paraméterére fény derül. Az 'e', 'a', 's', és 'g' gombok hatására az eleváció, azimut,

szín és gobó csatornákat lehet beállítani, majd a lámpa állapotát egy DMX üzenet azonnal frissíti, a dmxertekek.txt fájl pedig felülíródik a *mentKiir* függvény meghívása által. Előbbi két paraméter értékei relatív állíthatók az aktuális értékhez képest, a szín és gobó beállításoknál ellenben abszolút csatornaértéket kell megadni. Az 'r' parancs az aktuális referenciaértékeket menti fájlba és juttatja a DMX vonalra.

5. Követés

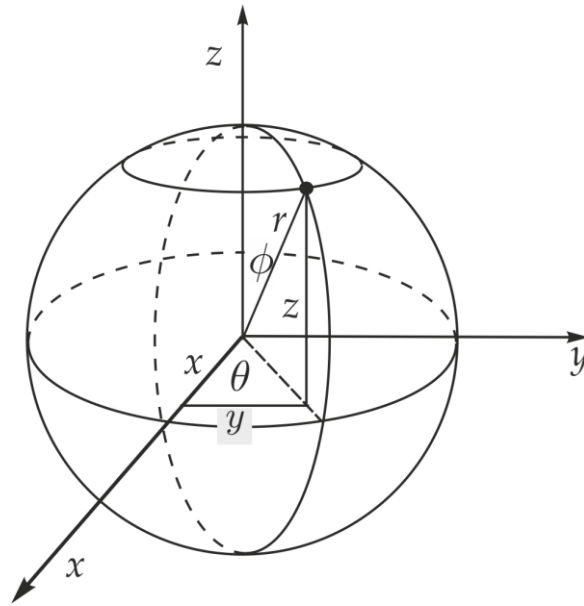
Az eddigi fejezetekben láthattuk a rendszer két szélső komponensét: a Kinect segítségével megtörténik az emberalakok helyzetének detektálása, illetve a DMX vonalon lehetséges a lámpák vezérlése. A követéshez azonban szükség van még a két komponens összekötésére.

5.1. Koordinátarendszerek

A Kinect Descartes-féle koordinátarendszerben ábrázolja az objektumokat. Ennek a koordinátarendszernek az alapja három ortogonális tengely, melyek – általában jobbsodrású rendszert alkotva – rendre x , y és z . Egy pont helyzete három koordinátával írható le: x a pont előjeles távolsága az y és z tengelyek által kifeszített síktól, ugyanígy y és z koordináták is a másik két tengely által kifeszített síkoktól mérhető távolságokat jelölik.

A lámpák vezérléséhez szükséges információk leírására a gömbi koordinátarendszer a legalkalmasabb, melyet a 36. ábra szemléltet. A gömbi koordinátarendszerben egy pontot három paraméter határoz meg: r , θ , φ . Az r megadja az origótól mért távolságot, azaz r a gömb sugara. θ a pont x - y síkú vetülete és az origót összekötő egyenes hajlásszöge az x tengelytől, azimut szögnek nevezik. Végül a φ a pontot az origóval összekötő egyenes hajlásszöge a z tengelytől, ez az elevációs szög. [36]

A földrajzi szélességi és hosszúsági fokok is ebben a rendszerben értendők, amennyiben z koordináta az Északi-sark felé mutat – annyi különbséggel, hogy a szélességi fok az elevációs értékből a $90^\circ - \varphi$ összefüggésből számítható, így értéke éppen az Egyenlítőnél 0° , a sarkoknál pedig $\pm 90^\circ$. A hosszúsági fokok referenciája a Greenwichen átfutó főkör.



36. ábra Gömbi koordináta-rendszer [36]

Amennyiben egy gömbi és egy Descartes-féle koordináta-rendszer a fenti ábrán látható módon kapcsolódik, az alábbi módon térhetünk át gömbi koordinátából közösleges koordinátákra:

$$\begin{aligned}x &= r \sin \phi \cos \theta \\y &= r \sin \phi \sin \theta \\z &= r \cos \phi\end{aligned}$$

A megvalósítandó feladathoz azonban éppen a Descartes-féle koordinátákból kell gömbi koordinátákra áttérni, ekkor az alábbi összefüggések alkalmazhatók:

$$\begin{aligned}r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \arctan\left(\frac{y}{x}\right) \\ \phi &= \arccos\left(\frac{z}{r}\right)\end{aligned}$$

5.2. Transzformációs modell

A lámpavezérlés során, a detektált objektum Kinect által szolgáltatott három Descartes-féle koordinátájából meg kell határozni a lámpa függőleges és vízszintes elfordulásának vezérlőinformációit. A Kinect koordináta-rendszerében y koordináta tartozik a függőleges irányhoz – a szokásos jelölésmód használhatósága, és az átláthatóság érdekében átneveztem a koordinátákat, hogy a z tengely jelölje a

függőleges irányt. Mivel a fényvető nem a Kinect által használt koordinátarendszer origójában van, első lépésként x , y és z koordinátákhoz rendre Δx , Δy és Δz eltolásokat adva el kell érni, hogy a kapott x' , y' és z' már egy olyan Descartes-féle koordinátarendszerben határozzák meg az objektum pozícióját, melynek origójában a lámpa található. A fentiek alapján $(-\Delta x; -\Delta y; -\Delta z)$ a lámpa pozíciója a Kinecthez rögzített koordinátarendszerben.

Ezt követően történik az előző fejezetben már látott gömbi koordinátákra áttérés. A kapott r érték jelen projektben nem kerül felhasználásra, ugyanis a teszteléshez használt lámpa nem képes a kibocsájtott fénynyaláb kúpszögét állítani. Amennyiben ez lehetséges volna, lehetővé válna a színészt körülvevő fénykör állandó nagyságon tartása, a kúpszög r függvényében történő állítása segítségével. Ha a lámpa ténylegesen azimut és elevációs szögeket dolgozna fel, a gömbi koordinátákra történő áttérés során kapott θ és ϕ közvetlenül képezhetné a lámpavezérlő információt. Azonban a helyzet ennél bonyolultabb: mind a vízszintes, mind a függőleges elmozdulás vezérlésére (a, e) a lámpa egy 8 bites értéket vár, azaz egy 0-tól 255-ig terjedő számot. Mivel a két tengely irányába az elfordulás maximális nagysága nem egyezik meg, ráadásul a nullapontok helyzete is önkényes, mindkét irányba bevezetendő egy skálafaktor és egy ofszet (f, g) , mely a kivezérelt szögtartományt leképzi a $[0; 255]$ tartományra.

A fentiek alapján a transzformációs modell az alábbi összefüggésekben foglalható össze:

$$\begin{aligned}x' &= x + \Delta x & y' &= y + \Delta y & z' &= z + \Delta z \\r &= \sqrt{x'^2 + y'^2 + z'^2} & \theta &= \arctan\left(\frac{y'}{x'}\right) & \Phi &= \arccos\left(\frac{z'}{r}\right) \\a &= f_1 \cdot \theta + g_1 & e &= f_2 \cdot \Phi + g_2\end{aligned}$$

5.3. Követő tesztprogram

A követés demonstrálására írt programom a már korábban bemutatott függvényeket, megoldásokat alkalmazza az adatok Kinect felől történő eléréséhez, és a lámpavezérléshez. Inicializálja a Kinectet és a DMX512 vonalmeghajtást, majd ciklikusan gyűjteni kezdi az adatokat a szenzor felől, az ezekből feldolgozott

vezérlőjeleket pedig minden ciklusban a kimenetre küldi. Amennyiben érvénytelen adat érkezik a bemenetről, azzal a feltételezéssel élünk, hogy éppen egyáltalán nem tartózkodik követendő objektum a megfigyelt területen, így ebben az esetben a kimenet gobócsatornájának 0 értékűre állítása elsötétíti a lámpát. Ellenkező esetben ennek a csatornának 10-es értékre állítása teljes megvilágítást eredményez, az azimut és elevációs csatornák pedig az első megtalált emberalak fejének pozíciója alapján értékelődnek ki, a transzformációs modell szerint. A hét modellparamétert (Δx , Δy , Δz , f_1 , g_1 , f_2 , g_2) fájlból olvassa be a program indulásakor.

6. Kalibráció

6.1. A kalibráció elve

Jogosan merül fel a kérdés: a szükséges modellparaméterek honnan állnak rendelkezésre? A korai tesztelések során a Kinectet a lámpára helyeztem, így az első három paraméterből kettőt nullának feltételeztem, a harmadikat pedig (a függőleges eltérést) lemértem, mely 0,1 m-nek adódott. Az elfordulások átskálázását szabályozó további négy paramétert empirikus módszerrel, döntően próbálgatással határoztam meg. Ez a módszer számomra is körülményes volt, valós körülmények között kivitelezhetetlen lenne. Biztosítani kell egy eljárást, mellyel a szükséges modellparaméterek egyszerűbben meghatározhatók – egyéb mérőeszközök igénybevétele, illetve próbálgatás nélkül.

A kalibráció a következőképpen történik: a színész egymás után néhány tetszőlegesen kiválasztott kalibrációs helyre áll, eközben a világosító technikus minden egyes esetben a lámpát manuálisan vezérelve a színész fejére állítja a fénykört, és az összes kalibrációs pont adatait rögzíti. A kalibrációs pontok rögzítése során olyan adathalmaz keletkezik, melyben a Descartes-féle koordináta-rendszerben meghatározott koordináták mellett ugyanezen pontokhoz tartozó lámpavezérlő információk is ismertek. A cél ezek alapján meghatározni a transzformációs modellparamétereket.

A transzformációs modell tárgyalásánál felsorolt 8 egyenlet mindegyike mátrixegyenletként fogható fel, hiszen egyszerre több kalibrációs pont transzformációs összefüggéseit vizsgáljuk. Így tulajdonképpen egy $8 \cdot k$ egyenletből álló nemlineáris egyenletrendszer adódik, ahol k a felvett kalibrációs pontok száma. Az

egyenletrendszerben $5 \cdot k$ ismert mennyiség (x, y, z, a, e) , a meghatározandó 7 modellparaméter $(\Delta x, \Delta y, \Delta z, f_1, g_1, f_2, g_2)$ és további $6 \cdot k$ ismeretlen $(x', y', z', r, \theta, \varphi)$ található. Így az egyenletek és ismeretlenek számára felírható a $8k > 7 + 6k$ egyenlőtlenség, ennek megoldása pedig azt jelzi, hogy 4 illetve több kalibráló pont esetén – ha a kapott egyenletek függetlenek – megoldható az egyenletrendszer.

6.2. Kalibráció a gyakorlatban

A probléma megoldására a Matlab fsolve függvényét alkalmaztam. A modellben használt utolsó két egyenletet nullára rendeztem, majd implementáltam a modell-egyenleteket egy függvényben, melynek bemenő paraméterei a modellparaméterek és a kalibrációs pontok adatai, visszatérési értékei pedig a nullára rendezett egyenletek nem nulla oldala által meghatározott érték. A nullával való egyenlőség ugyanis csak akkor állna fenn, ha a függvény paramétereként helyes modellparamétereket kapna. Az fsolve-val történő megoldáskeresésnek pontosan ez az alapja: a bemenő modellparamétereket addig iterálja a metódus, míg a visszatérési érték nulla (illetve egy bizonyos tűréshatáron belül) lesz.

A valóságos környezetben történő tesztelés előtt egy Matlabban elkészített modellt használva próbáltam ki az eljárást, amiben – a várakozásoknak megfelelően – 4 vagy több kalibrációs pont esetén adódott a helyes transzformációs modell. A kalibrációs adatokat zajjal terhelve is sikerült a kalibrálás.

A kalibrációs eljárás valóságban történő kipróbálásához szükség volt egy programra, mellyel felvehetők a kalibrációs adatok. A program a Kinect- és a lámpavezérlő tesztprogramban megismert módszerekkel lehetővé teszi a lámpa manuális ráirányítását a kalibrációs pontokra, majd rögzíti a lámpavezérlő információkat, illetve a Kinect által aktuálisan mért értékeket egy Matlab által könnyen értelmezhető fájlba. A kalibráció pontosságát elősegítendő a lámpa gobótárcsája teljes fénykör vetítése helyett az eljárás során csak egy kis fénynyalábot enged ki.

7. Integrált rendszer

Az egyes fejezeteknél bemutatott tesztprogramok segítségével megvalósítható a rendszertől elvárt működés, használatuk körülményes. Egyrészt minden funkciót külön tesztprogram valósít meg, másrészt a konzolos felület egyáltalán nem felhasználóbarát, nehezen kezelhető, ennek következtében például a kalibráló pontok felvétele meglehetősen hosszú időt vesz igénybe. Éppen ezért szükség van egy könnyen kezelhető felhasználói felületre, aminek segítségével az összes eddig megvalósított funkcionalitás (a lámpák manuális vezérlése, a követés, illetve a kalibrációs pontok rögzítése) elérhető.

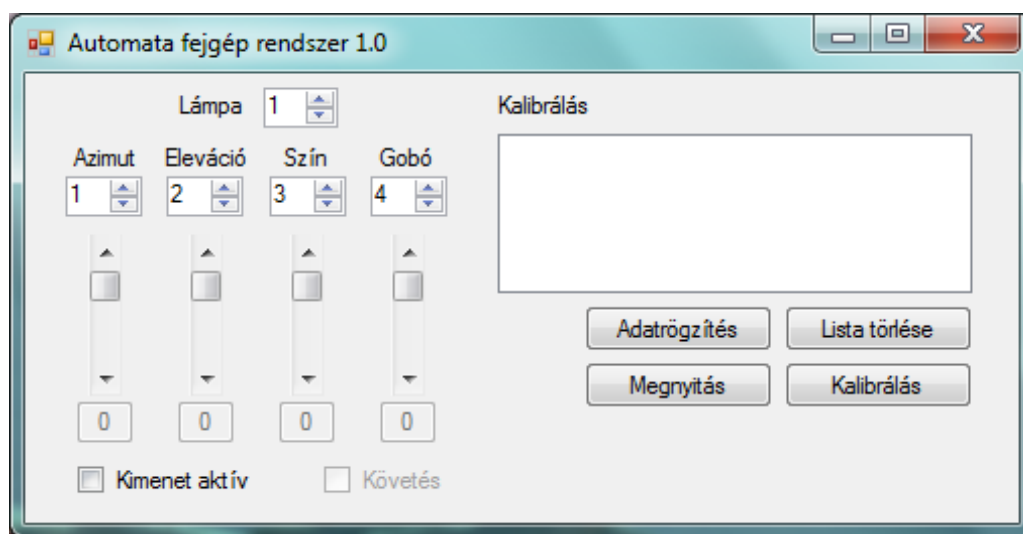
A program a korábbi tesztprogramokkal ellentétben már egyidejűleg több lámpát is képes irányítani. Moduláris felépítése pedig lehetővé teszi, hogy az egyes rendszerkomponensek további fejlesztése egy-egy osztály újrárásával megvalósítható legyen.

A programot C++ nyelvben írtam, a felhasználói felület kialakításához a Microsoft .NET keretrendszerének szolgáltatásait vettem igénybe, ami jelentősen megkönnyítette a kezelőfelület létrehozását. Továbbá a Microsoft Visual Studio fejlesztőkörnyezet segítségével a vezérlőelemek elhelyezkedése, mérete grafikus felületen szerkeszthető. Hasonlóképpen a vezérlőelemekre vonatkozó néhány alapbeállítás is (alapértelmezett érték, értékészlet, vezérlő letiltása, stb.) könnyen beállítható, az ezt megvalósító kódot a fejlesztőkörnyezet automatikusan generálja.

Mindazonáltal a C++ hagyományos megoldásai, és a keretrendszer programozási filozófiája sok helyen eltér, emiatt a két világ összekapcsolása néha problémákat vetett fel. Felmerül a kérdés, hogy ebben az esetben miért maradtam a C++ nyelvénél, miért nem választottam olyat, ami a keretrendszerhez közelebb áll (például a C#)? Ennek egyik oka, hogy jómagam C++ programozásban mélyedtem el, ennél azonban sokkal fontosabb, hogy a lámpák vezérlése során alkalmazott FTDI chip függvénykönyvtára ebben a nyelvben implementált. Így össze kellett egyeztetni a hagyományos és menedzselt memóriakezelést, illetve a típuskonverziók problematikája is többször felvetődött. A program (forráskódjával együtt) megtalálható a dolgozat elektronikus mellékletében.

7.1. Felhasználói felület

A program a 37. ábrán látható kezelőfelületet nyújtja a felhasználó számára.

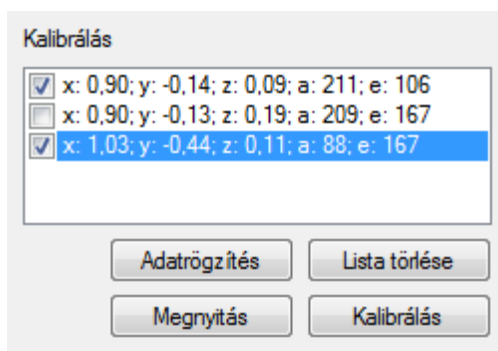


37. ábra A program kezelőfelülete

A „Lámpa” felirat melletti mező határozza meg az aktuálisan kiválasztott lámpa sorszámát – a többi kezelőelem mindig az éppen aktuális lámpát vezérli, lámpaváltáskor a lámpa eltárolt értékei töltődnek vissza. (Elindított követés azonban abban az esetben is folytatódik, ha közben más lámpa vezérlésére váltunk át.) A lámpafunkciók nevei alatti mezőkben lehet a DMX csatornákat beállítani. Ez alatt a csúszkák az adott csatorna értékének beállítására szolgálnak, a beállított értékeket a csúszkák alatti mezők is kijelzik (bár ezekben módosításra nincs lehetőség). A DMX512 vonal csak abban az esetben kerül meghajtásra, amennyiben a „Kimenet aktív” jelölőnégyzet be van jelölve. Az adott lámpánál a követő funkciót a „Követés” jelölőnégyzettel lehet be- és kikapcsolni. A fenti képen ez a jelölőnégyzet inaktív – ugyanis a program indulásakor még nincsenek a lámpákhoz társítva transzformációs paraméterek, ennek hiányában viszont nem kapcsolható be a követés. Aktív követés esetén a vezérelt funkciók csúszkái (azimut, eleváció és gobó), valamint a kalibrációval kapcsolatos vezérlőelemek tiltásra kerülnek.

Az adott lámpa transzformációs paramétereinek beállítása a jobboldalon található vezérlőkkel lehetséges. Az „Adatrögzítés” gomb egy kalibrációs pontot rögzít, mely a felette lévő listába kerül. Amennyiben a Kinect érvényes adatot szolgáltatott, az adott kalibrációs pont rögtön kijelölésre is kerül, ellenkező esetben jelöletlen marad, de természetesen a felhasználó is megváltoztathatja a kijelöléseket (elrontott kalibrációs

pontok érvénytelenítése céljából). A „Lista törlése” értelemszerűen törli a kalibrációs adatokat. A „Kalibrálás” gombra kattintva egy párbeszédablak nyílik meg, amiben kiválasztandó a kalibrált transzformációs paraméterek mentési helye. Ezt követően a listában kijelölt adatok alapján elindul a kalibráció, melynek során a program parancssoros módban meghívja a Matlabot, és elindítja a kalibrációt végrehajtó scriptet. Mivel a scriptet az *exit* utasítás zárja, a Matlab az eredmények kiszámítása után azonnal be is csukódik. Az eredmény a kívánt fájlba mentődik, és egyúttal a követés funkció is elérhetővé válik. (Valójában a program először összeállítja az adatokat a Matlab számára a kalibraloertekek.m nevű fájlba, a Matlab a temp.trp fájlba menti a paramétereket, amit aztán a program a megfelelő helyre és névvel másol, majd beolvassa – de erről a felhasználó nem értesül.) A „Megnyitás” gombbal korábban mentett transzformációs paramétereket lehet az aktuális lámpához társítani. Egy kalibrálás közben látható ablakrészletet mutat a 38. ábra.



38. ábra Kalibrációs adatok kezelése

7.2. A program felépítése

A programban használt objektumok a *lampairanyito* névtérben kerültek definiálásra. A *Kinect* nevű osztály kezeli az érzékelőt. Az osztály konstruktora inicializálja az eszközt, destruktora pedig lezárja. Az *adatRogzites* függvény az érzékelő által érzékelt első követett csontváz fejének pozícióját kérdezi le, a három koordináta eredményét *double* típusú tömbben adja vissza. A háromelemű tömböt a hívónak kell foglálnia, és a tömb mutatóját paraméterként kell átadni. Az *adatRogzites* függvény visszatérési értéke jelzi, hogy a csontvázkövetés éppen detektált-e emberalakot.

A *Dmx* osztály az FTDI chippel kapcsolatos teendőket látja el. Konstruktora az eszköz inicializálása után beállítja a DMX512 szabványnak megfelelő adatátviteli sémát, destruktora lezárja az eszközt. A *kuldes* függvény a buffer tartalmát a vonalra

küldi. A *set* és *get* függvény egy adott csatorna értékét írja a bufferbe illetve olvassa ki onnét.

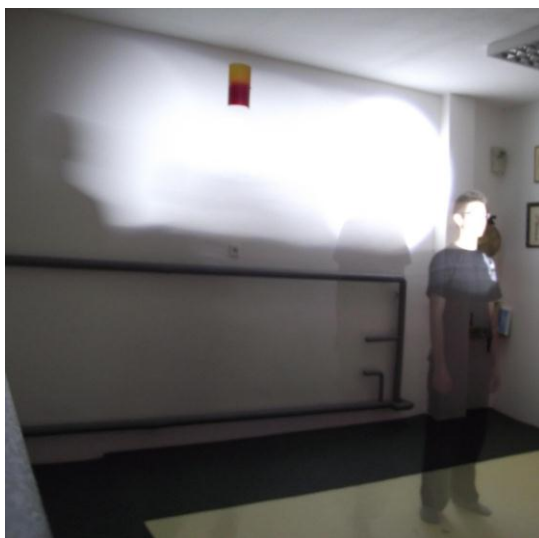
A *Lampa* osztály tárolja a lámpaváltáskor betöltendő összes adatot, és valósítja meg a transzformációs paraméterek segítségével a transzformációs modell működését. A *setTRP* és *getTRP* függvényekkel írhatók és olvashatók a transzformációs paraméterek. A *TRPOK* függvény meghívásával lehet jelezni, hogy a transzformációs paraméterek érvényesek, így a követés lehetséges. A *transzformacio* függvény a beérkező koordináták alapján, a transzformációs paraméterek segítségével beállítja a lámpa azimut és eleváció értékét a megfelelőre – azaz tulajdonképpen a tényleges követő funkciót implementálja. A *set* és *get* páros a *Dmx* osztályban látotthoz hasonlóan működik, annyi különbséggel, hogy csatornaszám helyett a lámpa funkcióját kell megadni. A *setCsatorna* és *getCsatorna* segítségével lehet karbantartani az adott lámpa csatornakiosztását, azaz hogy melyik funkciójához melyik DMX csatorna tartozik. A *setKovetes* segítségével állítható be a követés állapotváltozója, az *isKovetes* lekérdezi azt.

A *Kalibralas* osztály a kalibrálóadatok tárolására szolgál. A *pontFelvetel* függvény segítségével lehet új kalibrációs adatokat felvenni, a *get* függvénnyel adott indexű kalibráló adatsort lehet lekérdezni, míg a *torol* törli az összes adatot. A *hiba* osztály az eszközök inicializálása során fellépő végzetes hibák kezelését valósítja meg.

A *main* függvény a *Form1* osztály által definiált ablakot hozza létre és futtatja. A *Form1* tartalmazza a vezérlőkkel kapcsolatos összes adatot, amit a fejlesztőkörnyezet létrehozott. Ezen kívül tárol egy *Kinect*, egy *Dmx*, egy *Kalibralas*, és három *Lampa* példányosítást. Ez az osztály tartalmazza a vezérlők eseménykezelő függvényeit is. A kimenet vezérlése, és a lámpák követés funkciójának megvalósítása egy időzítő segítségével ütemezettek.

8. A rendszer tesztelése

A rendszer működését demonstráló videók megtalálhatók a dolgozat elektronikus mellékletében. Állóképen nehéz megjeleníteni a mozgást – a 39. ábrán látható hosszú exponálási idővel készített kép csak illusztrálni kívánja a követést.



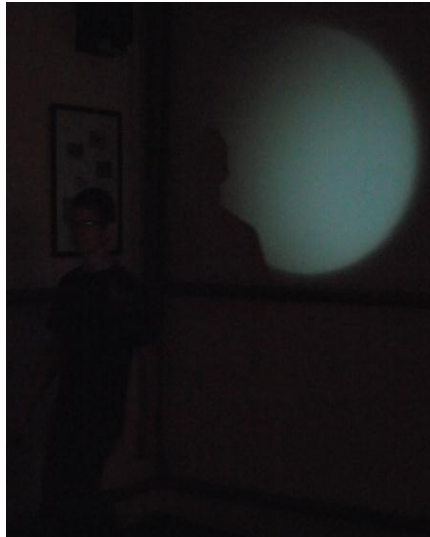
39. ábra Követés

A tesztelés során a program képes volt a lámpa manuális irányítására, a kalibrációs pontok felvételére, a kalibráció végrehajtására, illetve korábban elmentett transzformációs paraméterek betöltésére. Továbbá sikeresen követte az előtte mozgó alakot több érzékelő-lámpa geometriai helyzetet is kipróbálva, azok kalibrálása után. Több lámpa kiszolgálása is zökkenőmentesen működött – bár ennek kipróbálására csak a rendelkezésre álló egy lámpa áthelyezésével és átkonfigurálásával, szakaszosan volt mód.

8.1. Késleltetés

A követőrendszer működésének egyetlen komoly problémája a sebessége. A tesztelés során nyilvánvalóvá vált, hogy gyors mozgások esetén a fénycsóva valamelyest lemarad a mozgó testhez képest. Elég nagy sebesség esetén elérhető, hogy a fénycsóva elmaradjon a követendő alak mögött, amint ez a 40. ábrán látható – a színházakban nagyobb fényköröket alkalmaznak, mint amit a teszteléshez használt lámpa ilyen távolságból vetíteni képes, így valós körülmények között a fénycsóva elhagyásának veszélye kisebb, de a lemaradás minden bizonnyal ott is érzékelhető

lenne. Ennek oka valószínűleg a Kinectben keresendő, ugyanis a többi rendszerkomponens (transzformációs modell számításai, DMX vezérlés) sebessége lényegesen nagyobb.

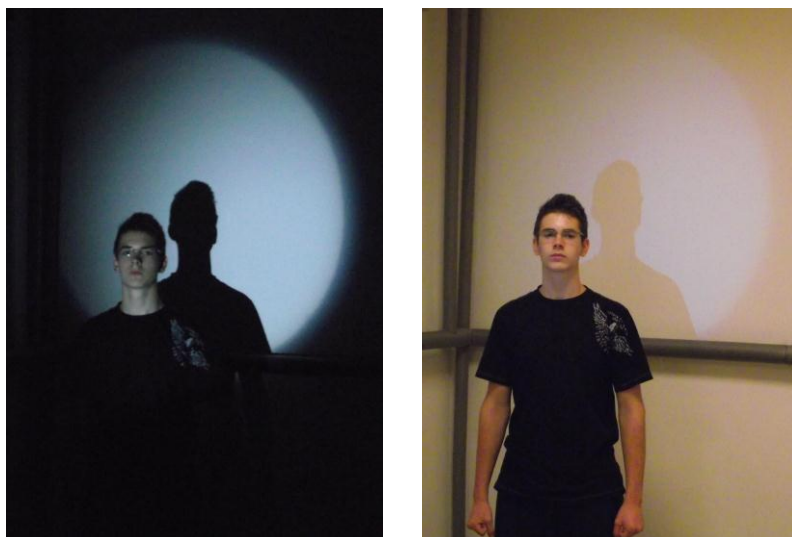


40. ábra Fénycsóva elhagyása

8.2. Érzékenység különböző fényviszonyokra

Mivel a Kinect képfeldolgozás útján határozza meg a szereplők pozícióit, erre hatással lehetnek a színpadon bekövetkező fényviszony változások. Ráadásul, amikor a színész a Kinect érzékelési területére belép, az emberalakként történő detektálás után a rendszer az addig elsötétített lámpát bekapcsolja, emiatt éppen a követett alak megvilágítása változik meg leginkább.

A színészre vetülő fénycsóva nem zavarta az érzékelést, a lámpa bekapcsolásakor jelentkező hirtelen változás sem. Továbbá kipróbáltam a rendszert különböző fényviszonyok esetén is: nappali fénynél, este felkapcsolt lámpák mellett, illetve teljes sötétségben. A követés mindhárom esetben megvalósult, nem érzékeltem eltérést a különböző esetek között. A 41. ábrán látható, hogy a követés különböző fényviszonyok mellett is zavartalanul működik.



41. ábra Követés különböző fényviszonyok esetén

Ennek az lehet a magyarázata, hogy a Kinect csontvázfelismerés funkciója a mélységi adatfolyam alapján határozza meg az előtte állók pozícióit, ez az adatfolyam pedig az infravörös kamera és az IR projektor segítségével készül, azaz aktív fényforrást alkalmazva, így a környezeti viszonyok nem zavarják a folyamatot.

8.3. Kalibráció

A matematikai modellel ellentétben valóságos kalibrációs eljárás során 4 pont felvétele a legtöbb esetben nem bizonyult elegendőnek (az így kapott transzformációs paraméterek hibás működést okoztak), ennek a jelenségnek egyrészt a felvett pontok által meghatározott egyenletek nem kellő függetlensége, másrészt a kalibráció során fellépő hiba (Kinect pontatlansága, kalibráló technikus hibája, lámpavezérlő információk 8 bites ábrázolásából fakadó kis pontossága) lehet az oka. 6 pont felvétele esetén azonban minden esetben sikerült helyes működésre bírni a rendszert, mindazonáltal a Matlab fsolve függvénye ebben az esetben is „No solution found” üzenettel tért vissza, ugyanis az iteráció során a függvény visszatérési értéke nem közelítette meg kellően a nullát. Valószínűleg ezt a jelenséget is az eljárás során fellépő különböző hibák okozzák. Mindenesetre az iteráció során kapott eredmény minden esetben ellenőrizendő a követő funkció kipróbálásával, hogy a megfelelő működést produkálja-e.

9. Összegzés

A projekt megvalósítása során először a pozíció meghatározó rendszerkomponenst vettem szemügyre, áttekintettem a pozicionáláshoz használható eszközöket. A színészek követésére a Microsoft Kinect érzékelőjét használtam, így bemutattam ennek képességeit. Majd egy tesztprogramot írtam a Kinect lehetőségeit feltérképezendő – természetesen a program elsősorban a projekthez kapcsolódó csontvázkövetés képességeit vizsgálja. Végül ennek segítségével egy méréssorozatot folytattam le, mely segítségével meggyőződtem az eszköz megfelelő pontosságáról.

A pozíció meghatározó rendszerkomponens taglalása után a rendszer másik oldala, a lámpavezérlés következett. Röviden áttekintettem a színpadi világítástechnikai megoldásokat, illetve a lámpavezérléshez használt DMX512 szabványt. Az FTDI USB to RS485 Serial Converter Cable megfelelő preparálásának segítségével létrehoztam egy csatolót, mellyel lehetségessé vált számítógép segítségével irányítani a lámpákat, ennek tesztelésére egy újabb programot írtam, mely képes a teszteléshez rendelkezésre álló DTS Vinciscan típusú intelligens fényvető minden funkcióját vezérelni.

Végül a két oldal közötti kapcsolatot kellett megteremteni, azaz a Kinect által detektált Descartes-féle koordináta-rendszerbeli pontokhoz lámpavezérlő információt rendelni. Létrehoztam egy transzformációs modellt, mellyel az áttérés megvalósítható. A transzformációs modell alkalmazásához 7 modellparaméter szükséges, melyek leírják egyrészt a rendszer geometriáját (a Kinect és a lámpa kapcsolatát), másrészt a lámpa pozicionáló rendszerének tulajdonságait. A paraméterek meghatározásához megalkottam egy kalibrációs rendszert, mely néhány pontra történő rávilágítás során rögzített pozíció- és lámpavezérlő információ segítségével kiszámolja a transzformációs paramétereket.

Megalkottam egy integrált rendszert, mellyel a lámpa manuális vezérlése, a kalibrálópontok felvétele, a kalibrálás, majd ezt követően a követés is ugyanazzal a programmal megvalósítható – a könnyebb kezelhetőség érdekében a korábbi tesztprogramok konzolos kezelőfelülete helyett ez a program már grafikus felülettel rendelkezik.

Végezetül a tesztelés következett. Ennek során a rendszer megfelelően biztosította a követést több, különböző elrendezésben elvégzett kalibrációt követően is (bár az ideális

modellből számítható 4 kalibráló pontnál gyakran több pont felvétele szükségeltetett a sikeres kalibrációhoz), a különböző fényviszonyok nem befolyásolták a működést. Amennyiben a követendő alak nagy sebességgel mozog az érzékelő előtt, a fénycsóva némileg lemarad, szélsőséges esetben a nagyon gyorsan mozgó alak képes elhagyni a fénycsót.

10. Továbbfejlesztési lehetőségek

A megvalósított felhasználói felület a kalibráció során elrejt a számításokhoz felhasznált Matlab programot, ám annak jelenléte szükséges a rendszer működéséhez. Az iterációs algoritmus közvetlen implementálásával olyan gépen is futtathatóvá válna a kalibrációs eljárás, amire nincs Matlab telepítve.

Felkészíthető a rendszer több szereplő követésére is, a Kinect által aktívan követett mindkét emberalakot meg lehet világítani. Az 5.2. fejezetben, a transzformációs modell tárgyalásánál már kitértem arra, hogy a szereplő lámpától mérhető távolsága is rendelkezésre áll, így ennek segítségével állandó nagyságú fénynyaláb lenne kialakítható a kúpszög változtatásával. Újabb lámpákon az irányok beállításához két csatornát is lefoglálnak, azaz egy bájt helyett két bájton ábrázolható a vezérlőinformáció. A rendszer kompatibilis ezekkel a lámpákkal, amennyiben a két bájt közül a nagyobbik helyértékűt vezérelheti a program. A kompatibilitáson túl továbbfejlesztési lehetőségként felkészíthető a program a kétbájtos információk kezelésére, és ezáltal pontosabb működésre ezeknél a típusoknál. Kiküszöbölendő, vagy legalábbis csökkentendő a rendszer késleltetése. Elképzelhető, hogy ennek egyetlen módja a Kinect lecserélése, de hosszú távú fejlesztési irányként ez amúgy is ildomos, a 3.6. fejezetben felsorolt hátrányai miatt.

A színészek követése nem csak a fejgépek automatizálása során használható ki. Hangosított előadások során – például musicalek esetében – a hangok panorámázása (a hangforrás irányának érzékeltetése) nem megoldott. A legtöbb esetben a hangosítás során mindig azonos helyről érzékeljük az adott szereplőt, azon ritka kivételek esetében pedig, amikor dramaturgiai jelentősége van a hang irányának, a hangkeverő pulton manuálisan végzik el a beavatkozást. A rendszer – ismervén a színészek pozícióit – a lámpákhoz hasonlóan a hangkeverő pultnak is küldhetne vezérlőjeleket a panorámázással kapcsolatban.

Forrásjegyzék

- [1] Chandra, A.M.: Higher Surveying, 2005. ISBN 978-81-224-1628-2, 2nd edition
- [2] Berle, F. J.: Mixed triangulation/trilateration technique for emitter location, 1986. IEE Proceedings-F, Vol. 133 <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04646979> [Online: 2012. nov. 30.]
- [3] Dr. Sárközy Ferenc: Térinformatika – Földi eljárások a pontok helyzetének meghatározására http://www.agt.bme.hu/tutor_h/terinfor/t31.htm [Online: 2012. nov. 30.]
- [4] Shinnosuke Hirata, Minoru Kuribayashi Kurosawa, Takashi Katagiri: Accurate measurement of distance and velocity using ultrasonic waves, 2008. Acoustics '08 Paris, Palais des Congrès <http://www.acoustics.org/press/155th/hirata.htm> [Online: 2012. nov. 30.]
- [5] IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) 2011. <http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf> [Online: 2012. nov. 30.]
- [6] Eiman Elnahrawy, Xiaoyan Li, Richard P. Martin: The Limits of Localization Using Signal Strength: A Comparative Study, 2004. IEEE Communications Society Conference <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1381942> [Online: 2012. nov. 30.]
- [7] Mohammad Said Hefny: Exploring GoogleGears Wi-Fi Geo Locator Secrets, 2010. <http://www.codeproject.com/Articles/63747/Exploring-GoogleGears-Wi-Fi-Geo-Locator-Secrets> [Online: 2012. nov. 30.]
- [8] Dr. Sárközy Ferenc: Térinformatika – Inerciális navigációs rendszerek http://www.agt.bme.hu/tutor_h/terinfor/t36.htm#inr [Online: 2012. nov. 30.]
- [9] Stephanie Crawford: How Microsoft Kinect Works <http://electronics.howstuffworks.com/microsoft-kinect.htm> [Online: 2012. nov. 30.]
- [10] Microsoft. Kinect for Windows <http://www.microsoft.com/en-us/kinectforwindows/> [Online: 2012. nov. 30.]
- [11] Nintendo. Wii <http://www.nintendo.com/wii> [Online: 2012. nov. 30.]
- [12] Wii Accidents Montage, 2009. <http://www.youtube.com/watch?v=HAhIqX6lSCs> [Online: 2012. nov. 30.]
- [13] <http://www.gameshop.hu/img/wii-remote-controller.jpg> [Online: 2012. nov. 30.]
- [14] Kinect megateszt, 2010. http://www.gamekapocs.hu/cikk/1495/kinect_megateszt [Online: 2012. nov. 30.]
- [15] Barna József: A Kinecttel újra izgalmassá válik a PC, 2012. http://itcafe.hu/hir/microsoft_kinect_for_windows_pc_sdk.html [Online: 2012. nov. 30.]
- [16] Microsoft. Kinect érzékelő kézikönyv
- [17] Darabjaira szedték a Kinect szenzort, 2010. http://www.gamekapocs.hu/hir/20889/darabjaira_szedtek_a_kinect_szenzort [Online: 2012. nov. 30.]
- [18] Microsoft Corporation: Kinect for Windows Software Development Kit Documentation, 2012. <http://www.microsoft.com/en-us/kinectforwindows/develop/resources.aspx> [Online: 2012. nov. 30.]

- [19] Seller Rudolf: Antennarendszerek alapjai <http://hvt.bme.hu/~seller/Radar/Antennarendszer.DOC> [Online: 2012. nov. 30.]
- [20] Seller Rudolf: Antenna topológiák <http://hvt.bme.hu/~seller/Adaptív/Jegyzet/5fejezet.doc> [Online: 2012. nov. 30.]
- [21] Encyclopedia Britannica: Human Eye – Binocular Vision <http://www.britannica.com/EBchecked/topic/1688997/human-eye/64969/Binocular-vision?anchor=toc64969> [Online: 2012. nov. 30.]
- [22] Csodák Palotája. Ames-szoba <http://www.csodakpalotaja.hu/index.php/hu/ames-szoba> [Online: 2012. nov. 30.]
- [23] Alan Johnston: Depth Perception, 2009. <http://www.psychol.ucl.ac.uk/alan.johnston/Depth.html> [Online: 2012. nov. 30.]
- [24] Hányingerkeltők és fárasztóak a 3D-s filmek, 2010. http://szamitastechnika.network.hu/blog/szamitastechnika_klub_hirei/hanyingerkeltok-es-farasztoak-a-3-d-s-filmek [Online: 2012. nov. 30.]
- [25] How Kinect depth sensor works – stereo triangulation?, 2010. <http://mirror2image.wordpress.com/2010/11/30/how-kinect-works-stereo-triangulation/> [Online: 2012. nov. 30.]
- [26] Candide – a parameterized face <http://www.icg.isy.liu.se/candide/> [Online: 2012. nov. 30.]
- [27] Nathan Crock: Kinect Depth vs. Actual Distance, 2011. <http://mathnathan.com/2011/02/03/depthvsdistance/> [Online: 2012. nov. 30.]
- [28] Centrál Színház. Műszaki adatok <http://www.centralszinhaz.hu/muszaki-adatok> [Online: 2012. nov. 30.]
- [29] Pelyhe János: Világítástechnikai jegyzet, 2010. http://www.pelyhe.hu/index.php?option=com_phocadownload&view=category&download=86:08-2-szinpadvilagitas-es-vezerlesteknika [Online: 2012. nov. 30.]
- [30] Ujjal Kar: DMX512 <http://www.dmx512-online.com/> [Online: 2012. nov. 30.]
- [31] Maxim Integrated. Guidelines for Proper Wiring of an RS-485 (TIA/EIA-485-A) Network, 2001. <http://www.maxim-ic.com/app-notes/index.mvp/id/763> [Online: 2012. nov. 30.]
- [32] Elation Professional. Dmx 512 Handbook, 2008. <http://www.elationlighting.com/pdffiles/dmx-101-handbook.pdf> [Online: 2012. nov. 30.]
- [33] RDM Protocol <http://www.rdmprotocol.org> [Online: 2012. nov. 30.]
- [34] FTDI. USB to RS485 Serial Converter Cable – Datasheet http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_USB_RS485_CABLES.pdf [Online: 2012. nov. 30.]
- [35] FTDI. Software Application Development – D2XX Programmer's Guide [http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer's_Guide\(FT_000071\).pdf](http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer's_Guide(FT_000071).pdf) [Online: 2012. nov. 30.]
- [36] Dr. Kovács Emőd: *Komputergrafika – Matematikai alapok*, 2011. <http://aries.ektf.hu/~hz/pdf-tamop/pdf-01/html/ch03.html#id672217> [Online: 2012. nov. 30.]

Ábrajegyzék

1. ábra Rendszerkomponensek.....	7
2. ábra Háromszögelés	8
3. ábra Trilateráció	9
4. ábra Kinect [10].....	11
5. ábra Wii Remote [13].....	11
6. ábra A Kinect belseje [17].....	13
7. ábra Érzékelők és a motor elhelyezkedése [18]	14
8. ábra Működési tartományok [18]	15
9. ábra θ szögből érkező síkhullám [19]	16
10. ábra Antennarendszer [19]	17
11. ábra A Csodák Palotájában található Ames-szoba [22]	19
12. ábra Retinális diszparitás [21].....	20
13. ábra A binokuláris látás segítségével térhatású kocka	20
14. ábra SDK architektúra [18]	22
15. ábra A csontvázkövetés által támogatott testrészek [18]	23
16. ábra Aktív és passzív követés [18].....	24
17. ábra Álló és ülő mód [18]	24
18. ábra A Kinect koordinátarendszere.....	25
19. ábra A testrészek hierarchikus szervezése [18].....	25
20. ábra Arckövetés által meghatározott pontok [18]	26
21. ábra A fej különböző mozgásai [18]	27
22. ábra Mérőbábu	30
23. ábra Mérési elrendezés.....	31
24. ábra Mélységi adatok	34
25. ábra Mélységmérés hibái.....	35
26. ábra Mélységmérés relatív hibái	36
27. ábra Nathan Crock mérési elrendezése [27].....	37
28. ábra Nathan Crock mérési eredménye [27].....	38
29. ábra x koordináta mérési eredményei.....	39
30. ábra x koordináta hibái	40
31. ábra Fényjáték intelligens fényvetőkkel.....	43

32. ábra Mozgótükörös- és mozgófejes fényvető	44
33. ábra DMX512 jelalak [30]	47
34. ábra USB-DMX kábel.....	49
35. ábra Csatlakozók	49
36. ábra Gömbi koordináta-rendszer [36]	52
37. ábra A program kezelőfelülete	57
38. ábra Kalibrációs adatok kezelése	58
39. ábra Követés.....	60
40. ábra Fénycsóva elhagyása	61
41. ábra Követés különböző fényviszonyok esetén	62

Táblázatok jegyzéke

1. táblázat A fej különböző mozgásai	27
2. táblázat Szórások (csípő közepe)	32
3. táblázat Szórások (fej).....	32
4. táblázat Mélységmérés illesztett egyeneseinek paraméterei	34
5. táblázat X koordináta mérése során illesztett egyenesek meredeksége	39
6. táblázat XLR csatlakozó lábkiosztása [32]	45
7. táblázat DMX512 keretszervezése	47
8. táblázat DTS Vinciscan fényvető csatornakiosztása.....	48
9. táblázat Az FTDI kábel ereinek és az XLR csatlakozó lábainak összerendelése ..	49

Rövidítések jegyzéke

3D	Three-dimensional
API	Application Programming Interface
CD	Channel Data
DDR	Double Data Rate
DMX	Digital Multiplex
EIA	Electronic Industries Alliance
FTDI	Future Technology Devices International Ltd.
IGBT	Insulated Gate Bipolar Transistor
IR	Infrared
LSB	Least Significant Bit
MAB	Mark After Brake
MSB	Most Significant Bit
MTBF	Mark Time Between Frames
MTBP	Mark Time Between Packets
NUI	Natural User Interface
RDM	Remote Device Management
RGB	Red, Green, Blue
RS	Recommended Standard
SC	Start Code
SDK	Software Development Kit
TIA	Telecommunications Industry Association
USB	Universal Serial Bus
USITT	The United States Institute of Theatre Technology
UWB	Ultra Wideband
VGA	Video Graphics Array
Wi-Fi	Wireless Fidelity

Függelék

DTS Vinciscan fényvető színcsatorna kiosztása

DMX érték	Szín
0-17	Fehér
18-35	Világoskék
36-53	Sötétkék
54-71	Vörös
72-89	Zöld
90-107	Bíbor
108-125	Sárga
126-143	Enciánkék
144-161	Rózsaszín
162-179	Narancs
180-197	Világoszöld
198-223	Sötétkék
224-231	Forgó tárcsa (1)
232-239	Forgó tárcsa (2)
240-247	Forgó tárcsa (3)
248-255	Forgó tárcsa (4)

DTS Vinciscan fényvető gobócsatorna kiosztása

DMX érték	Gobó
0-9	Elsötétítés
10-19	Nincs gobó
20-29	Negyedelt négyzet
30-39	Csillag körvonal
40-49	Csillagok
50-59	Csillag
60-69	Hajlított vonalak
70-79	Négy függőleges vonal
80-89	Nyolc pötty
90-99	Vízszintes sáv
100-109	S alak
110-119	Háromszög körvonal
120-129	Körgyűrű
130-139	Virág
140-149	Kör alakban kis körök
150-159	Három kör háromszög alakban
160-169	Három kör vízszintesen
170-179	Kisméretű kör

180-189	Közepes méretű kör
190-199	Elsötétítés
200-209	Nincs gobó
210-217	Stroboszkóp (1)
218-225	Stroboszkóp (2)
226-233	Stroboszkóp (3)
234-241	Stroboszkóp (4)
242-255	Nincs gobó