

TDK DOLGOZAT

Készítette:

Müller Péter János



PÉCSI TUDOMÁNYEGYETEM
UNIVERSITY OF PÉCS

Pécsi Tudományegyetem
Pollack Mihály Műszaki és Informatikai Kar
Műszaki Informatika Tanszék

**Valós idejű intelligens szemmozgás követő rendszer
fejlesztése**

Készítette: Müller Péter
Mérnök Informatika MSc

Konzulens: Dr. Schiffer Ádám,
Egyetemi adjunktus

Pécs
2013

Tartalomjegyzék

1. Bevezető	- 4 -
2. Meglévő rendszerek bemutatása	- 4 -
2.1 NRC Kft.	- 5 -
2.2. Eyetracking –project	- 6 -
3. A megoldandó probléma megfogalmazása.....	- 7 -
4. A digitális képfeldolgozás	- 7 -
5. A mérés összeállítása	- 8 -
5.1. A mérést megelőző teendők	- 13 -
5.2. A mérési folyamat során lehetséges fellépő hibaforrások	- 15 -
5.3. A mérés menete	- 16 -
6. A képfeldolgozási folyamat	- 17 -
6.1. A képfeldolgozási folyamathoz felhasznált programok	- 17 -
6.1.1. LabView 2013 rövid bemutatása	- 17 -
6.1.2 A Vision Builder bemutatása és használata.....	- 17 -
6.1.2.1 A Vision Builder használata közben felmerült problémák	- 19 -
6.1.3. Vision Assistante 2013 rövid bemutatása.....	- 20 -
6.1.3.1. A Vision Assistant fő funkciói.....	- 21 -
6.1.3.2. A Vision Assistant alkalmazása.....	- 21 -
7. A programhoz felhasznált egyéni tervezési scriptek.....	- 23 -
7.1. A mintakeresés	- 23 -
7.2. A pupilla megtalálása, avagy a kör alakzat keresése	- 26 -
8. A LabVIEW 2013 alkalmazásban elvégzett feladatok	- 29 -
8.1. A mintakeresés LAbVIEW implementációja.....	- 29 -
8.2. A körkeresési algoritmus LabVIEW implementációja	- 29 -
8.3. A koordináta transzformációs műveletet végző vi bemutatása	- 32 -
9. A főprogram bemutatása	- 34 -
9.1. A háromszög alakzat pont egyenes távolságának számítása.....	- 37 -
10. Kiértékelés	- 38 -
11. Konklúzió	- 42 -
11.1. Eredmények.....	- 42 -
11.2. Jövőbeni tervek.....	- 42 -
11. Ábrajegyzék	- 43 -
12.Hivatkozások:	- 44 -
13. Források.....	- 44 -

1. Bevezető

A mai világban sok figyelmet fordítunk az emberek tevékenységeinek, reakcióinak megismerésére. Ezt főként orvosi vizsgálatok segítségével, megfigyelések alapján végezzük. Jó példa az emberi devianciákat vizsgáló Szondi-teszt, ahol az alany szimpatikus és antipatikus arcok közül választ. Az egészségügyi vizsgálatok mellett fontos szerepet játszik az emberek, gazdaságban való részvétele, ebben a szférában végzett kísérletek is segítik a forgalom növelését („mysteryshopping”)^[1].

Az érzékszerveink közül fontos szerepet játszik a szem, mellyel felfogjuk és érzékeljük a minket körülvevő világot. A tekintetet, illetve a szemmozgást az egyik legnehezebb feladat követni és monitorozni, mert annak vizsgálatához szükségünk van az alany beleegyezésére és jelenlétére. A manuális módszerek nem elég megbízhatóak, hiszen a megfigyelt személytől ebben az esetben csak a vizsgált alany elmondására támaszkodhatunk, aki megpróbálhatja félrevezetni a mérést.

Ennek a kiküszöbölésére modernebb automata, félautomata rendszereket használnak, amiknek a fő feladata, az emberi szem mozgásának megbízható, hiteles megfigyelése és annak követése, illetve a mérést követően az eredmények kiértékelése. Az általam elkészített program, ennek a feladatnak az elvégzésére szolgál.

A dolgozat többek között tárgyalja a már meglévő rendszereket és azok rövid bemutatását, a mérés előkészítését és annak körülményeit, valamint a használt eszközök és programok bemutatását. Ezen felül külön fejezetben mutatja be a felmerült probléma megoldását, valamint az azokhoz használt algoritmusok és komplex matematikai műveletek leírását.

A feladatot ellátó program a National Instruments LabVIEW 2013-as verziójával és annak kiegészítő moduljaival, a National Instruments által fejlesztett Vision Assistance 2013-as verziójával, illetve a Vision Builder 2013-mal készült. Szó esik majd továbbá a technológia továbbfejlesztési lehetőségeiről is, illetve az aktuális megoldás hibáiról és hiányosságairól.^{[2] [3]}

2. Meglévő rendszerek bemutatása

Az elvégzendő feladat megtervezését megelőzte a már aktívan működő, rendszerek felkutatása és megismerése. Világszerte több nagyvállalat és magántársaság fordít egyre nagyobb hangsúlyt az általa forgalmazott termék vagy szolgáltatás jobb marketingjére és az eladási ráta növelésére. Kutatásokkal bizonyították, hogy egy jól megalkotott reklámkampány, akár több mint 30%-kal növeli meg az adott termék iránti érdeklődést. Annak vizsgálatára, hogy mi is vonzza az emberek figyelmét a verbális, non-verbális viselkedést vizsgálták. A verbális megnyilvánulás során a kérdésekre válaszoló alany félrevezetheti a felmérő személyt, de a non-verbális gesztusok és

érzékelések reflexszerűek, így ezek vizsgálata reprezentatív mérési eredményekkel szolgálnak. A szem mozgása is egy ilyen nem, illetve nehezen befolyásolható érzékelés.^[4]

A mai napig is több kutatócsoportot foglalkoztató téma, a szem mozgásának követése.

A kutatási munkám során a Pécsi Tudományegyetem által rendelkezésre bocsátott több folyóirat és könyvadatbázist, ahol az éppen folyó kísérletekről informálódtam.

Megfigyelhető hogy több helyen is alkalmaztak fejkamerás felvételezést, kijelzőbe beágyazott kamerát valamint szem elé tehető keretbe foglalt „szemüveget” is.^{[5][6]}

A módszerek között található kalibrációval ellátott, illetve képkorrektációs eljárást is alkalmazó megoldások is. A szem megtalálása valamint a pupillaközéppont megadása, javarészt mintakereséssel történik, de itt is találhatók különböző megoldások.^{[7][8]}

Az eredmények felhasználása széleskörű, de mégis a legnagyobb számban az orvostudomány számára jelent fontos és szinte kimeríthetetlen információforrást. Segítségével időben kiszűrhetőek az olvasási és figyelem zavaros gyermek, devianciák és betegségek valamint a tanulási képesség is fejleszthető.^{[9][10]}

Nem elhanyagolható a szem mozgás vizsgálatához kapcsolódó alkalmazások száma. A fejlesztők célja, hogy segítsék a mozgásukban korlátozott személyek mindennapjainak megkönnyítése.^[11]

2.1 NRC Kft.

Napjainkban az elérhető rendszerek professzionális megoldást kínálnak a szemmozgás követésére. Hazánkban az egyik vezető piackutatást végző cég, az 1999-ben alakult NRC Internet Marketing Tanácsadó és Piackutató Kft. felismerte, hogy a növekvő internet használat és az írásos média sikerességéhez szükség van a célközönség megismerésére, és annak feltétlen kiszolgálására.

Ennek elősegítésére több piac- és médiakutatási eljárás mellett nagy figyelmet fordítottak az igények pontos feltárására. E célok elérésében a cég segítségére volt a szemmozgást követő rendszer megvalósítása. Segítségével reprezentatív és értékes információt kaphatunk arról, hogy az általuk választott virtuális, illetve valós termék mennyire nyerte el a vásárló tetszését és mennyire egyértelmű az az információ, amit közölni kíván. Megfelelő példaként szolgálnak erre a tényre az eladott termékek csomagolásai, nyomtatott anyagok és weboldalak.

A boltokba kerülő termékek nagy részére – ha nincs nagy múltú hírneve – nehéz felhívni a vásárló figyelmét, hacsak nem ismeri a gyártó cég azokat a csomagolási technológiákat és módszereket, amivel megnyerhető a célközönség a tetszése. A weblapoknál a cég lehetőségeket nyújt a tervezési fázistól az eladásig, továbbá a folyamatos tesztelésre a vizsgálatba bevont személyek segítségével. A vizsgálati módszerek igény szerint változnak az egyszerű tekintet

követéstől az összetett feladat elvégzések is (próbavásárlás, regisztráció). Próbavásárlási művelet alatt, webshopon keresztül végzett vásárlást értünk.

A tesztek során feltárt hiányosságok ismeretében a termék megjelenése formálható és sikerebbé tehető.

Az NRC Kft. a mérést két nagyfelbontású infra kamerával végzi, ezzel rögzítve a vizsgált alany szemmozgását. Egy rövid kalibrációs eljárást követően a rendszer lehetőséget nyújt bármilyen képanyag által keltett ingeranyag vizsgálatára, mivel a szemmozgás a legtöbbször önkéntelenül reagál az új ismeretanyagra, és annak működése nehezen hamisítható.

A módszernek köszönhetően a kiértékelés után olyan információkat tudhatunk meg, hogy a vizsgált személy melyik pontra tekintett először, milyen sorrendben és milyen hosszú ideig figyelt egy adott pontra, régióra. ^{[12][13][14]}

2.2. Eyetracking –project

Az internet használatának növekedésével, az internetes reklámok, hirdetések elterjedésével megnőtt az érdeklődés az iránt, hogy megtudjuk, mi az, ami igazán megfogja a felhasználót a töménytelen mennyiségű adat közül. Egy weboldal sikerességét nagyban meghatározza annak tematikus tervezése és megjelenése.

Erre figyelt fel a C3 Alapítvány az OmnicomMediaGroup-pal együtt, melynek eredményeképpen egy közös projektben készítettek alkalmazást az internetezési szokások megfigyelésére. A projektjük célközönsége a 15-49 év közötti korosztály, iskolázottság és korcsoport szerint rendezve. A 120 fővel elvégzett mérés során bizonyították, hogy az emberek közötti különbségek (nemi hovatartozás, kor, érdeklődés) meghatározzák, hogy mit és mennyi ideig néznek egy oldalon. Ismertté válik, hogy mi az, ami több másodperc erejéig is megragadja a figyelmet, és mi az, amin átsiklik a felhasználó.

Érdekes eredménynek tartom, hogy a magyar kultúrát jellemző balról jobbra olvasásnak köszönhetően a weboldalak megtekintése közben is így működik a szemünk, de a kutatási eredmények mégis azt mutatják, hogy a lap tetején illetve alján elhelyezkedő reklámok több figyelmet kapnak.

Maga a kutatáshoz használt eszköz, egy szemkamera, amely megjelenésében egy kereskedelmi forgalomban kapható webkamerára hasonlít, de ezen felül rendelkezik infravörös fény kibocsátására alkalmas lámpákkal (ledekkel). Ezek a lámpák az emberi szemre ártalmatlan mennyiségben bocsátanak ki fényt, de ez elősegíti a mérés sikerességét. Hiszen a retináról visszaverődő infravörös fény homogén, könnyen vizsgálható és elemezhető anyaggal szolgál az automatizált eszköz számára, így csökkenthető, illetve kizárható több kritikusnak számító hibaforrás.

A cég távlati terve, hogy a szemkamerával ne csak a magyar internet használók habitusát térképezze fel, hanem több médiatípust is megismerhessenek. Ezek között szerepel a nyomtatott sajtó anyag és a televíziózási szokásokat monitorozó rendszerek használata. Továbbá hangsúlyozzák, hogy nem versenytársként akarnak jelen lenni a piacon, hanem párhuzamosan más kutatási eredményekkel együtt egy új perspektívából kívánják alkalmazni a fejlesztésüket.^[15]

3. A megoldandó probléma megfogalmazása

A TDK pályamunkám célja egy megbízható, emberi szemmozgást követő rendszer fejlesztése, és az ehhez szükséges modern képfeldolgozási eljárások alkalmazása.

Céлом az volt, hogy az általam elérhető eszközök segítségével, egy a szem követésére alkalmas környezetet és alkalmazást készítsék. Ezen felül a hardveres és szoftveres környezet biztosítása, és egy a feladatra alkalmas fejlesztői környezet kiválasztása. A program tervezése során képfeldolgozási-, matematikai-, és geometriai műveletek felhasználására, és a feladat specifikussá tételére is szükség volt. Az elkészült algoritmusok tesztelése és hibátűrésének vizsgálata, majd a részegységek összeillesztése és kiértékelésére is a feladat része.

A mérési folyamat sajátossága egy második ember, a vizsgálni kívánt alany kiválasztása, személyes jelenlétét és hozzájárulását igényli.

4. A digitális képfeldolgozás

Az előző két példánál bemutatott eszközök és rendszerekhez használt mérések, túlnyomórészt digitális eszközzel rögzített videó-, és képanyagok. Ezek feldolgozásához és kiértékeléséhez szükségünk van alap képfeldolgozási és kép manipulációs műveletek egységére, technikai, esetekben matematikai műveletek alkalmazására.

A digitális képfeldolgozásnál arra, törekszünk, hogy az eredeti képet egy képfeldolgozási módszer, illetve algoritmus segítségével manipuláljuk, és ezzel plusz információhoz jussunk.

A kezdeti állapot egy digitalizált kép, amelyet kép-átalakítási módszerek segítségével alkalmassá teszünk további vizsgálatok elvégzésére.

A képfeldolgozásban főbb műveleteket az alábbi csoportokba sorolhatjuk: képkorrekció, elemzés, képfelismerés.

A képatalakítás műveleteknél az eredeti képben található zajforrások és kontraszthiányos állapotok megszüntetése a célunk. Erre megfelelő eszközök az alul-, felül áteresztő szűrők, mediánszűrők.

A képelemzés fázisában a vizsgálathoz szükséges információ kigyűjtésére törekszünk, illetve arra, hogy a vizsgálatra kijelölt részlet információ tartalma elegendő legyen. Ebben a fázisban történik a képen található alakzatok felismerése.

Az utolsó lépés a képfelismerés, amely során a kinyert jellemzők szerint csoportba sorolhatjuk a képet.

Fontos megemlíteni, hogy a képmanipulációs műveletek mennyiségének növelése a kép torzulásával együtt jár, ezzel pedig fontos képi információktól eshetünk el. Ezért a kép átalakítása közben figyelniünk kell arra, hogy a vizsgált minta a művelet után továbbra is tartalmazza a szükséges információ halmazt.^[16]

5. A mérés összeállítása

Az ideális mérési körülmények megteremtéséhez több főbb szempont figyelembe vétel is elengedhetetlen, mivel ezek szorosan kapcsolódnak egymáshoz, és a megfelelő környezeti paraméterek használata nagyban meghatározza a mérési eredmény sikeres kimenetelét.

Az ideális helyszín kiválasztásnál figyelembe kell venni azt, hogy a mérőkör felépítéséhez elegendő hellyel rendelkezünk, és a fény a vizsgált személy arcának irányából érkezzon. Fontos kihangsúlyozni, hogy a reflexió, a fény visszatükröződése elkerülése érdekében a vizsgálat közben az összes hátulról érkező fényforrást meg kell szüntetni.

A munkafolyamat során az alany és a vizsgálatot végző személy zavartalan kommunikációja elengedhetetlen, ezért lehetőség szerint nyugodt és a nagyobb társadalmi és fizikai behatásoktól mentes terület ajánlott.

A mérés pontosságát növeli a kiválasztott alany ideális mérési pozícióba való elhelyezése. A legfontosabb, hogy a vizsgált személy a teljes folyamat során fejét mozdulatlanul tartva kövesse az instrukciókat. Az ember számára ez nehezen fenntartható merev állapot. A fej pozíciójának megőrzésére egyéni tervek alapján készült fejtámasztó eszközt alkalmazunk, ami az 1. ábrán megtekinthető. Mielőtt a fej rögzített pozícióba kerül, lehetőségünk nyílik a magasasság beállítására a vizsgált személy igényeinek megfelelően.



1. ábra: A fejtámasz

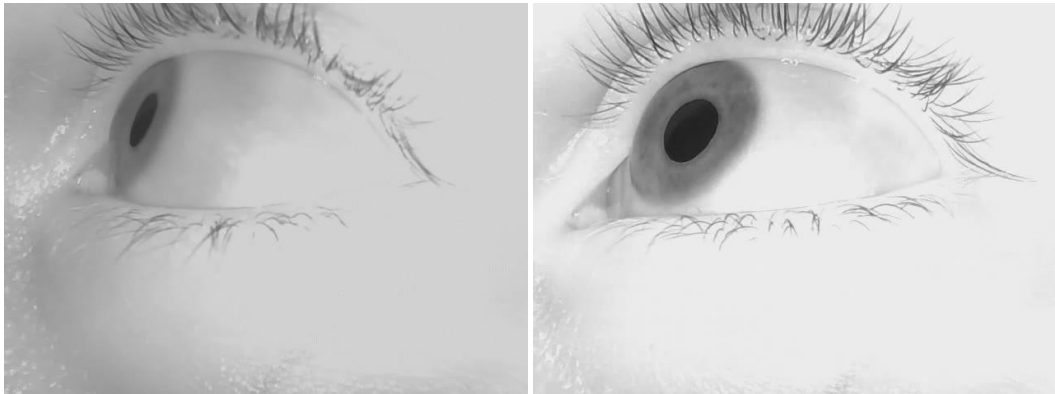
Korábban már említést tettem róla, hogy a mérőkör elemeinek kapcsolata szorosan összefügg, ezért a pozíciójukat meghatározza a többi egység elhelyezkedése. A fej fixálását biztosító fejtámla a monitortól vett távolsága 12 és 15 cm-között változott a mérések során, a kiértékelt mérésnél a 15cm-re helyezkedett el a kijelzőtől.

A felvétel rögzítéséhez egy pályázat során kapott NI 1774C Smart Camera-t használtam, képe a 2. ábrán szerepel. A mérés során a pupilla mozgását rögzítem, és az infraleddel megvilágított pupilla az utófeldolgozás során sokkal jobban kiértékelhetővé vált. A mérés megtervezése során az alapkoncepció az volt, hogy az alany szemmozgását a szükséges kalibrációs eljárás után rögzítem, majd valós időben kiértékelem, és megjelenítem azt. A mérést az előzőek alapján végeztem el. Az eszköz kiválasztásakor fontos szempont volt annak videó rögzítési képességének maximális felbontása. Az NI 1774C Smart Camera 1,3 Megapixeles kép folyamatos készítésére alkalmas optikával bír, maximális felbontása SXGA, azaz 1280*960 pixel



2. ábra: NI 1774C Smart Camera

A professzionális nagyvállalati rendszerekben alkalmazott fejkamerás, illetve közvetlen szem elé rögzíthető kamera hiányában, a kamera elhelyezését több előmérés előzte meg, ami arra szolgált, hogy az alany szeme minél kisebb szöget zárjon be, de a monitoron teljes egészében belátható maradjon. Ha ez a szög túl nagy, a pupilla képe deformálódik és sarló, illetve ellipszis alakúvá válik, mint az a 3. ábrán is látszik. Továbbá az sem kizárt, hogy bizonyos pontoknál a pupilla elmozdulás nagyobb lesz egy adott pixel távolságon, mint a másikon, s mindez a mérés pontatlanságához vezethet.



3. ábra: Deformált és ideális minta. A bal oldali ábrán a kamera-szem-monitor elhelyezése a feldolgozás szempontjából hibás értékekhez vezethet.

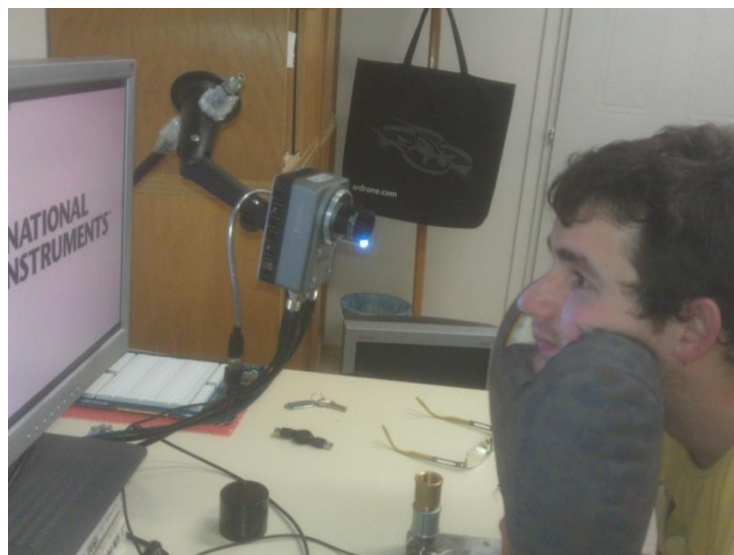
A kamera távolsága a szemtől 7-9 cm között változott. A regisztrált mérésnél 9 cm-re helyezkedett el.

A zavartalan méréshez a programot futtató PC-hez egy másodlagos kijelzőt csatlakoztattam, így az alany és a mérést végző személy is zavartalanul látja a számára szükséges információt.

A második kijelző lehetővé tette, hogy a vizsgálatot végző személy a főképernyőn dolgozhasson, és a mérés alatt figyelemmel követhesse a rögzítési folyamatot, a képek vetítését szabályozza, és indokolt esetben közbelépjen.

A mérés elvégzésére egy mobil eszközt, laptopot alkalmaztam, amelynek kijelzőjének dőlés szöge igény szerint változtatható, ami az alap felhasználásnál ideális, viszont precíziós méréseknél hátrányt jelent a mozgó alkatrész.

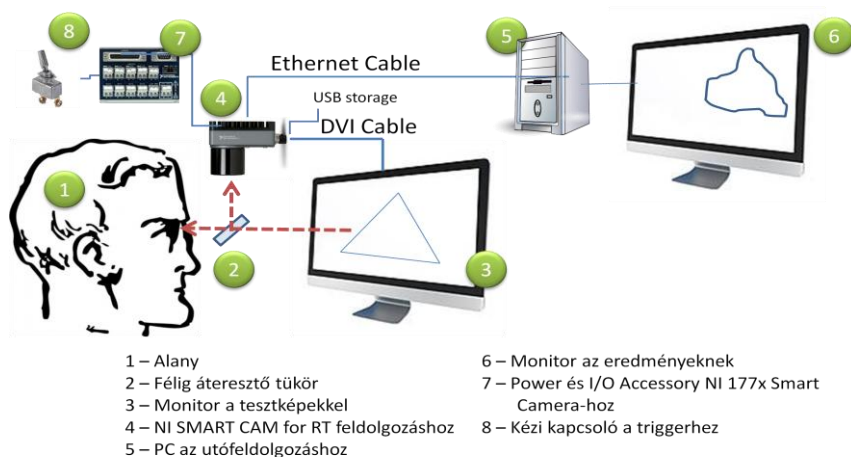
A másodlagos kijelző fix állású, és így nem áll fenn annak veszélye, hogy az mérés közben elmozdul.(4. ábra)



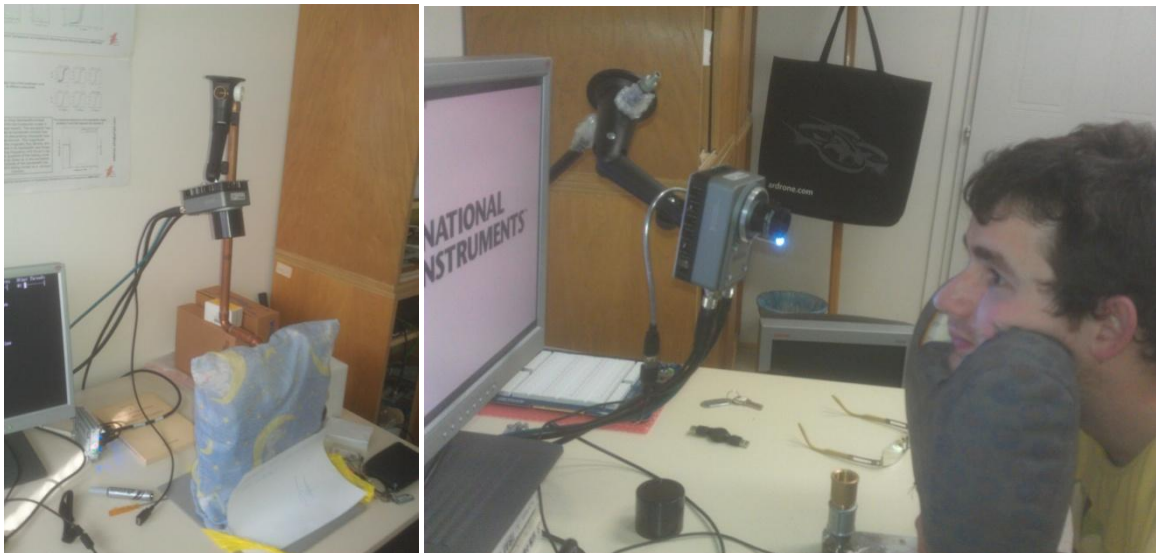
4. ábra: Az elkészült mérőkör alannyal

A mérőkör összeállításához mind szoftveres, mind hardveres eszközökre is szükség volt. Ezen felül az új eszközök megléte fontossá tette, hogy az általam készített eszközöket is fejlesszem.

A pályázaton igényelt eszköz paraméterei elérhetők a National Instruments oldalán, így annak, méreteinek ismeretében megalkottam a számomra szükséges szabadsági fokokkal rendelkező kameratartó elemet, amelyet összeépítettem a korábban alkalmazott fejtámasszal, valamint lehetőségem volt annak állványra szerelésére is. A mérési kör így stabilabb elemekkel bővült, és a körülmények megismétlése is egyszerűbbé vált. Az alábbi képen (5. ábra) látható az egyik új mérési kör tervezetének ábrája, továbbá az átalakított eszközök.



5. ábra: A mérési kör



6. ábra: A mérési kör két változata

A mérési körön látható (6. ábra) a teszt alany, aki keresztül néz egy félig áteresztő lencsén, és megtekinti az ábrákat. Ennek a lencsének köszönhetően a kamera képes rögzíteni a szemmozgást és elvégezni a programozott feldolgozást, valamint ezt követően menti az eredményeket. Lehetőség van arra is, hogy a géppel kapcsolatot létesítve a vizsgáló személy kiértékelje a kapott eredményt.

A pályázat során igényelhető eszközök és szoftverek lehetőséget nyújtottak, hogy a meglévő rendszerem tovább fejlesszem, és az ott megfogalmazott jövőbeli terveim megvalósítsam. Nagy előrelépést jelentett a National Instruments 2013 szoftvere, és annak kiegészítő moduljai, valamint az NI SMART CAM 1774 C nevű eszköz, és az ahhoz szorosan kapcsolódó VISION Builder AI. Ezek segítségével egy robosztusabb és sokkal megbízhatóbb rendszer kiépítését valósíthatom meg.

Az említett kamera saját architektúrával rendelkezik, és képes valós idejű feldolgozásra, valamint az elkészített eljárások tárolására és futtatására. Dupla akkora felbontással rendelkezik, mint a korábbi mérésben használt web kamera, ami még nagyobb pontosságot tesz lehetővé. Ráadásul, az NI által forgalmazott világítással bővíthető, továbbá a hozzá tartozó vezérlő panel segítségével trigger jelekkel is vezérelhető.

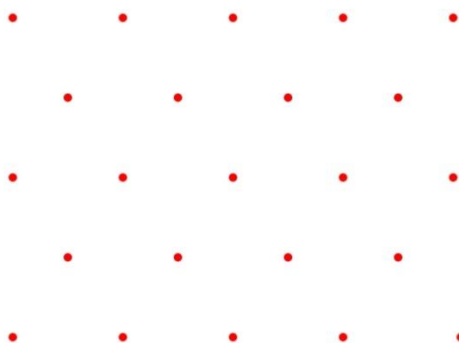
Az alkalmazás a National Instruments LabVIEW 2013-es verziójú programjával, illetve a National Instruments VISION Assistance 2013-es verziójú kiegészítő moduljával készült, ezek mellett még az NI Vision Builder használatával.

A választásnál döntésemet az indokolta, hogy az NI LabVIEW fejlesztőkörnyezet kiválóan alkalmas mérés-adatgyűjtési folyamatok elvégzésére vizuális környezetben, így a karakter forrású programozási nyelvekkel ellentétben nincs szükségünk a GUI és a grafikai modulok létrehozására, mert ahhoz kész elemként hozzájutunk. A fentiek mellett a PTE Pollack Mihály Műszaki és Informatikai Karán elvégzett Vizuális Programozás, Kép- és hangfeldolgozás kurzusok, illetve a karon lévő Labview Akadémián megszerzett CLAD vizsgám is sokat segített a fejlesztőkörnyezet választásában.

5.1. A mérést megelőző teendők

A mérőkör felállítását követően a méréshez szükséges kalibrációs pontok tervezése volt a következő lépés. A kalibrálás során, a képernyőn megjelenített pont koordinátája ismert, és ehhez társítjuk hozzá a felvételen rögzített szemmozgás állást. A kalibrációs pontok megválasztásakor arra törekszünk, hogy azok egyenletes eloszlásban helyezkedjenek el a teljes kép terjedelmében. A kalibrációs pontok növelésével a mérési folyamat pontossága is nő, de a túl sok ponttal történő számolás időigényes művelet lenne, és elérve egy határt a pontossági ráta növekedése megállna. A kalibrációs pontok tervezésének kezdeti szakaszában 18 pont elhelyezéséről döntöttem. A kalibrációs pontok az NI Labview 2013-as fejlesztőkörnyezetével paraméteresen kiszámolhatóak, illetve kép (*.png) formátumban elmenthetőek.(7. ábra) Az elvégzett próbamérést követően a kiértékelésnél derült ki, hogy a valós képen a két koordináta közötti érték változásra a szem túl kis változással reagál. Számszerűsítve ezt a leírást, a képen 270 pixel távolságra a szem 20 pixelnyi elmozdulással reagált, a teljes pupilla út pedig mindössze 94 pixelt tett ki. Ez több mint 10%-os hibát jelent, aminek értéke a kiértékelés során, és azzal, hogy az alany akaratlan más felé néz tovább nő.

Ahogy korábban azt már említettem, a kalibrációs pontok növelésével a szem mozgásának apróbb reakciói is megfigyelhetők és letárolhatók. A feladat megoldásához végül a 23 pontból álló kalibrációs pontrendszerrel választottam ezzel csökkentve a mérési hibát. A mérési hiba alatt a pupilla helyzetének a képen történő reprezentációját értjük. Ebben az esetben már 145 pixelváltozásra a képen a szem 34 pixellel reagál, az összes mozgás pedig 210 pixel távolság volt.



7. ábra: Kalibrációs tábla NI Labview fejlesztőkörnyezettel előállítva

A tekintet megtartásával tovább növelhetjük a pontosságot, ezért 3 stílusú kalibrációs pontot készítettem. Az első normál méretű piros pont volt, a második dupla méretű, a harmadik pedig a dupla méretű fekete pont közepén egy normál méretű pirossal, melyek képe a 8. ábrán látható.

De nem csak a mérési kör változik az új erőforrásokkal. Az igényelt kamera felbontása is nagyobb, a felbontás növelésével a kalibrációs folyamat során alkalmazott kalibrációs táblázatot is bővítettem 23-ról 28 pontra. Ezzel fenntartva a korábban maximalizált hibaküszöböt és pontosságot.



8. ábra: Kalibrációs pontok

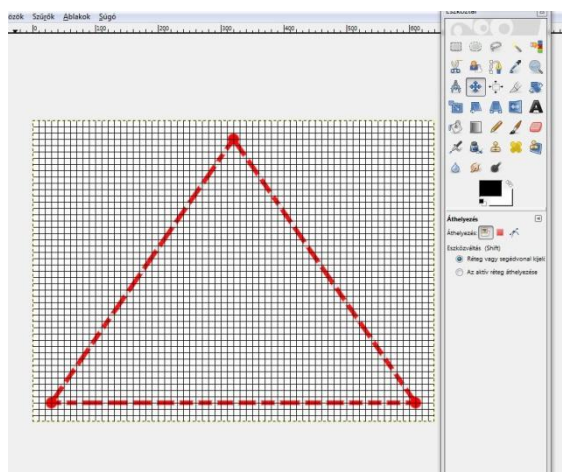
A kalibrációs tábla pontjainak koordinátája

X	Y
30	30
175	30
320	30
465	30
610	30
102,5	135
247,5	135
392,5	135
537,5	135
30	240
175	240
320	240

X	Y
465	240
610	240
120,5	345
247,5	345
392,5	345
537,5	345
30	450
175	450
320	450
465	450
610	450

A kalibrációs táblát követő lépés a mértani alakzatok tervezése és megvalósítása volt. Ezekre azért volt szükség, mert egy egyszerű alakzatot a tekintetünkkel irányítottan képesek vagyunk végigjárni. Ennek a kiértékelés fázisában van jelentősége, hiszen ezek pontjait ismerjük és ennek segítségével kiszámolható, hogy a program által kalkulált érték mennyiben tér el a valóságtól.

Hasonlóan a kalibrációs pontok tervezéséhez, itt is próbamérésekből szerzett tapasztalatok alapján, kézi tervezést követően és a GIMP képszerkesztő programot alkalmazva készült el a két ábra, egy egyenlő oldalú háromszög és egy téglalap. A 9. ábrán a háromszög képe látható.



9. ábra: Alakzat készítése GIMP-ben

Az alakzatok elkészítése során a tekintet megtartására és irányítására a formák csúcspontjait megnagyobbítottam, és az alakzatok pontjait összekötő egyenesre fehér pontokat helyeztem,

amellyel szaggatott hatást értem el. A több rövid szakasz megfigyelése és nyomon követése könnyebb feladatot jelent a vizsgált személynek, mint az egybefüggő egyenes.

Az alakzat koordinátáit a mérés kiértékelését követően az egyenes egyenletek felírásához használtam fel.

Az alakzatok koordinátái:

Háromszög	X	Y
1	30	450
2	610	450
3	320	30

Négyszög	X	Y
1	30	450
2	610	450
3	610	30
4	30	30

Harmadik lépésként három méretben az alakzatokkal megegyező kép letöltése volt, melyek pontjai nem ismertek és a felhasználó számára új ismeretanyaggal szolgálnak, amit ő tetszőleges körülménnyel tekinthet meg.

5.2. A mérési folyamat során lehetséges fellépő hibaforrások

Méréseim során több hibajelenséggel is találkoztam.

Ezek közé tartoztak technikai jellegű hibák (elmozdult kamera, kamera szoftverével kapcsolatos hibák), illetve az emberi mulasztásból elkövetett hibák. Az emberi mulasztásból származó hibák száma automatizálással (időzített diavetítés funkció) csökkenthető. Továbbá a korábbi alfejezetben említett hibák kiküszöbölése és minimalizálás, csökkenti a mérés közben elforduló hibák számát és mértékét.

Mindezek mellett még egy, a véleményem szerint fontos lélektani szempontra hívnám fel a figyelmet. Megfigyeltem, hogy az alany kezdetben feszült volt a számára szokatlan körülmények miatt. Emiatt ajánlott több mérés elvégzése, mivel a második, illetve harmadik mérés után feltűnően csökken a vizsgált személy diszkomfort érzése.

5.3. A mérés menete

A mérés megkezdése előtt az alany elfoglalja a helyét és fejét az erre a célra készített fejrögzítőbe helyezi. Ezt követően a vizsgálatot végző személy beállítja a megfelelő magasságot és a kamera, monitor pontos pozícióját.

A vizsgálatot végző személy megnyitja a képek levetítéséhez szükséges alkalmazást. Erre a feladatra az ingyenesen elérhető IrfanView nevű alkalmazást választottam, amelynek több funkcióját is felhasználtam a pályázat készítése során. A szem mozgását az NI Smart Cam segítségével rögzítettem.

A műveletet megelőzően ismertetni kell a vizsgált személlyel a mérés menetét.

A kalibrációs pontok rögzítésének megkezdése előtt szükséges egy utolsó ellenőrzés, hogy az alany megfelelően belátja-e a megjelenítő minden részét, illetve a vizsgálatot végző személy is készen áll-e a mérés megkezdésére. Ezt követően elkezdődhet a mérési folyamat, mely 3 ciklusra bontható: kalibrációs pontok rögzítése, alakzatok felvétele, végül pedig a választott képek megtekintése. Fontos ezek egyidejű végrehajtása.

A mérés menete dinamikus. Az alany zárt szemhéjjal jelzi, hogy készen áll a folyamatra. Ezt a tény nyugtázva a vizsgáló személy elindítja a felvételt és megjeleníti az első pontot a kijelzőn, majd vezényszóra a szem nyitására kéri a vizsgált személyt.

A pont megtalálása a másodperc töredéke alatt történik, ami a kontroll képernyőn figyelemmel kísérhető. Ezt követően egy 2-3 perces késleltetéssel a szemhéj lezárására szólítjuk fel.

Ezt a műveletet az összes kalibrációs pontra megismételjük. Ezzel lezárva a kalibrációs pontok rögzítését, amelyet a videón egy hosszabb idejű csukott állapotú szemmel jelzünk.

A második ciklusban az ábrák pontjainak felvétele történik. Előzetes egyeztetés során minden esetben az alakzat bal alsó sarkából indulva az óra mutató járásával ellenkező irányba halad a tekintet. Amint az alany készen áll, ismét vezényszóra kinyitja a szemét, és tekintete természetes tempóban végighalad a megadott irányba az alak vonalán, közben pedig rövid ideig várakozik a csúcsokon. A mérések lezárását itt is csukott szemmel indikáljuk. A harmadik fázisban az alanynak instrukció nélkül mutatunk képeket, így azokat a saját tempójában és egyéni döntés alapján tekintheti meg. A képek közötti váltást, hasonlóképpen a korábbi ciklusokhoz, a szemhéj lezárásával jelöljük.

Amint az utolsó képet is megmutattuk és a vizsgált személy lezárja a szemét, a mérést lezártnak tekintjük.

A felvételek rögzítésére illetve a pontok mentésére a mérőkör ábrájában szereplő kapcsolót triggert használjuk, az állapotok közti ugráshoz.

6. A képfeldolgozási folyamat

6.1. A képfeldolgozási folyamathoz felhasznált programok

6.1.1. LabView 2013 rövid bemutatása

A National Instruments által fejlesztett LabView (Laboratory Virtual Instrumentation Engineering Workbench) fejlesztési környezete lehetőséget nyújt mérnöki és főleg mérés-adatgyűjtési munkák vizuálisan történő megvalósítására. A hagyományos és nagyobb múlttal rendelkező OOP (C++, Java) szemléletű illetve procedurális programozási nyelvekkel (Pascal, C) ellentétben a vizuális programozási nyelv sok esetben nyújt egyszerűbb és áttekinthetőbb megoldási modellt. Ehhez azonban egy másik programozási szemléletre kell áttérnünk, amit Dataflow-programmingnak nevezünk.

Az adatfolyam programozásra jellemző a „balról jobbra” elv, melynek lényege, hogy a vezérlő elemeket balra, a megjelenítő elemeket jobbra helyezzük el, ezek között pedig huzalokkal létesítünk kapcsolatot. Egy LabVIEW fájl, melynek kiterjesztése *.vi (Virtual Instrument - virtuális műszer) három komponensből épül fel: a Front Panel, a Block Diagramm és a Connectorok.

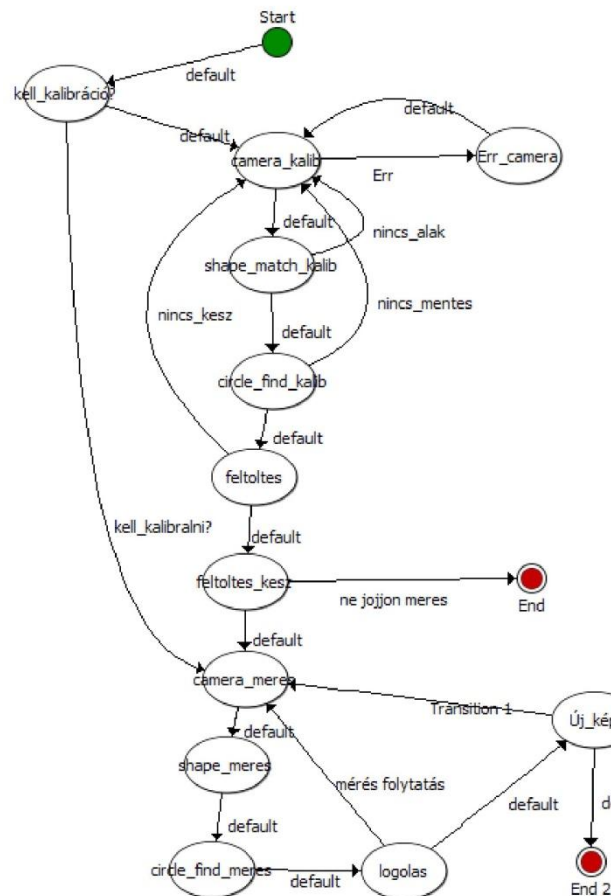
A felhasználó a Front Panelen keresztül készítheti el a program vázát, helyezhet fel vezérlő és megjelenítő elemeket a palettáról, továbbá itt definiálhatók a kezdeti beállítások is. A program futtatása közben ezen a felületen kaphatunk információt az aktuális folyamatról.

A Block Diagrammon nyílik lehetőség a műveletek, funkciók, programozási struktúrák implementálására, amiket a Tools palettán elérhető Wiring Tool segítségével kapcsolhatunk össze.

A Connectorok az elkészített vi-ok újrafelhasználásnál kapnak nagy szerepet. Itt a vi-on használt ki- és bementekhez rendelhetünk connectort, így egy kompakt eszközt – más néven (User Defined Vi-t) – kapunk, amit egy másik programban felhasználhatunk, és ezzel egy komplex program megvalósítása modulokra bontható.^{[17][18][19][20]}

6.1.2 A Vision Builder bemutatása és használata

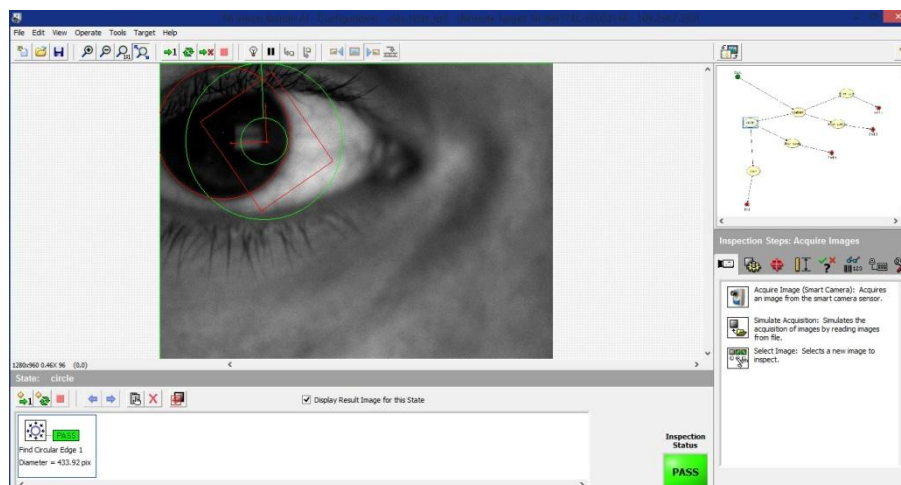
A kamerához alkalmazhatjuk a Vision Builder AI-t, amivel a vizsgálati eljárást megtervezhetjük, és elkészíthetjük. Az itt elkészített eljáráshoz felhasználhatjuk a Builder AI, a Vision Module funkcióit és nem utolsósorban a LabVIEW fejlesztői környezet teljes eszközkészletét, felhasználói Vi formájában. Ez a modul lehetőséget nyújt, hogy a programot állapotgráfos szemléletben (10. ábra) készítsük el. Az alábbi ábrán megfigyelhetők az állapotok.



10. ábra: Az alkalmazás állapotdiagrammja

Az első lépésben inicializálom a kamerát, majd a Vision Assistant segítségével előfeldolgozást végzek az adott képeken. Ide sorolható a képek szürkeárnyalatossá tétele, valamint a fényerő és kontraszt beállítások. Ezt követően egy minta keresési eljárás segítségével megkeresem az adott képen a szemet, és a megtalált pontra koordináta rendszert illesztetek. Az ábrán jól látható, hogy az állapotgráfos megoldás segítségével a hibakezelés is könnyen kivitelezhető. Ebben a lépésben két értéket is vizsgállok. Első alkalommal a kamera állapotát figyelve alkotom meg a feltételt, és annak hibája esetén jelzem azt egy Overlay segítségével, a második hibafigyelő állapot pedig azt figyeli, hogy sikerült-e lokalizálni a mintát. Hiba esetén mindkét alkalommal újra indul a vizsgálat. A második állapotban a megtalált mintára illesztem a körkeresési algoritmust. Hiba esetén itt is újra indítom a vizsgálati ciklust. A következő állapotban a kapott középponti koordinátákat egy felhasználói vi segítségével transzformálom át a képernyőn látott koordináta értékekre, és USB-n csatlakoztatott háttértárra logolom a kapott mérési értékeket és a rögzített képeket is. A letárolt koordináta pontokat *.csv formátumba mentem, és egy konverziós műveletet követően a vizsgáló gépen kiértékelem a korábban látott módon. Ez a megoldás több szempontból is kimagasló megoldást nyújt. A mérési állapotot a vizsgált személy adataival, a mérési körülmények paramétereivel, és a mérés során kapott értékekkel egyidejűleg tárolhatom. A mentés szöveges

formátumba történik, így helytakarékos a megoldás. Minden elvégzett mérés visszaállítható újbóli kiértékelés céljából. A következő ábrán megfigyelhető a mérési folyamat képe (11. ábra). A Vision Builder használatával verzió kezelési problémák miatt nem volt lehetőségem dolgozni. Ezeket a problémákat a későbbiekben még kifejtem.



11. ábra: A Vision Builder AI használat közben

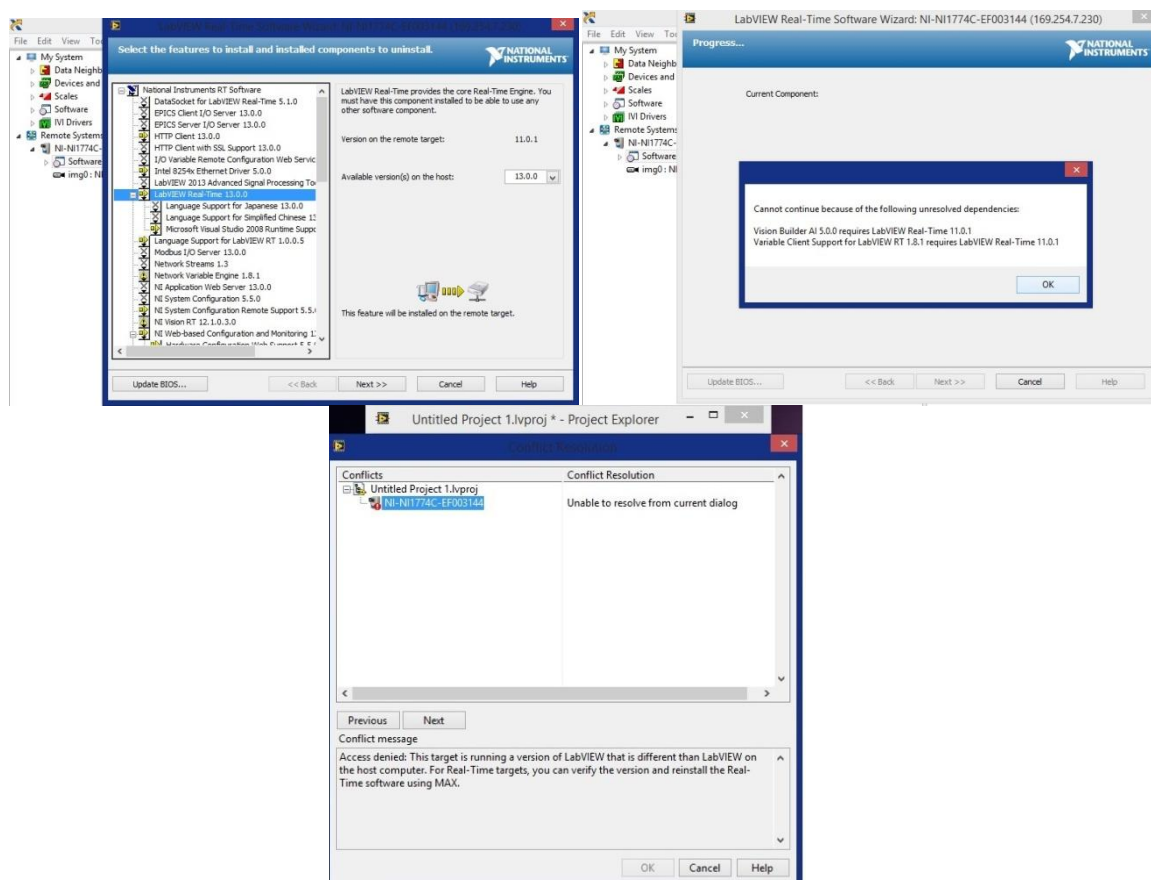
6.1.2.1 A Vision Builder használatában felmerült problémák

Az előző fejezet végén említettem, hogy számos szoftveres akadállyal is szembesültem a fejlesztés során (12. ábra).

Először a hardware-es elemek közül említeném meg a világítást, mely az igénylés ellenére sajnos nem volt kompatibilis a kiválasztott és megkapott eszközzel. A képeken azonban látható, hogy ennek pótlása sikeresen megtörtént.

Szoftveres oldalról több problémával is szembesülnöm kellett. A kamerához megkapott szoftver, az NI Vision 2012 SP1 verziójú csomagot tartalmazta, és ehhez kaptam a LabVIEW 2013 fejlesztő csomagját és kiegészítőit. A Vision 2012 SP1 tartalmazza a kamera felprogramozásához szükséges fájlokat, amelyeket a kamera első használatbavételekor kell telepíteni, és feltölteni. A későbbiekben lehetőség nyílik arra, hogy a PC-ről modulokat frissítsünk a MAX-on keresztül. Először itt szembesültem a verziók közti különbségekből adódó problémákkal. A rendelkezésemre álló RT modul a verziószáma 13.0.0 a kameráé pedig 11.0.1. Itt két választási lehetőségem lett volna. Elsőként, hogy a LabVIEW 2011-es verzióját használom, és megszüntetem a különbséget, vagy a kamera Real-Time modulját frissítem fel 2013-ra. Sajnos egyik út sem volt járható. Az elsőt a szoftver hiánya gátolta, a másodikat pedig a Vision Builder AI 5.0.0 függése. Ennek hiányában nem tudtam volna végre hajtani a számomra fontos műveleteket a kamera segítségével. Kísérletet tettem a probléma megoldására, azzal, hogy egy vi-on keresztül próbálom elvégezni a műveleteket, de a RT modulok verziószám különbsége itt sem engedte a végrehajtást. Az alábbi képeken láthatók a hibaüzenetek.

Az NI Pályázat eredményhirdetése során lehetőségem nyílt a fejlesztőkkel történő konzultációra, akik megerősítették, hogy a fejlesztőkörnyezet verziókezelési problémái ismertek és biztosítottak arról, hogy az alkalmazott megoldások megfelelőek és elfogadottak a fejlesztés során.

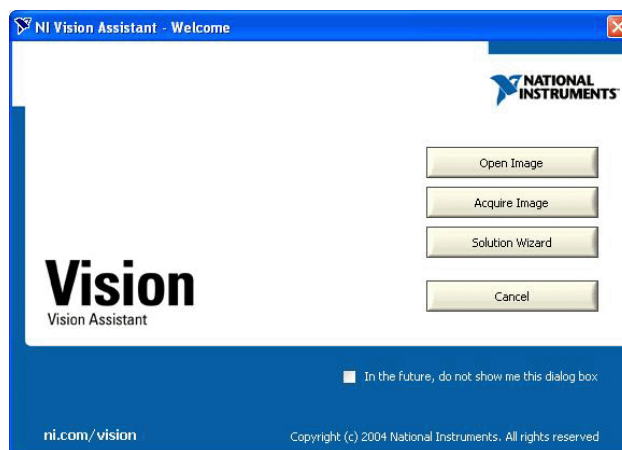


12. ábra: Az előforduló verzió kezelési problémák

6.1.3. Vision Assistant 2013 rövid bemutatása

A Vision Assistant modul (13. ábra), egy a LabVIEW alkalmazástól független, képmanipulációs műveletekre optimalizált szoftver. Lehetőséget biztosít egyéni algoritmusok létrehozására, és annak lépéseinek, paramétereinek script fájlban történő rögzítésére.

Az algoritmus elkészítése mellett lehetőségünk van tesztelésre, és a fázisok lépésinek beállítására, finomhangolására. A LabVIEW VI Creation Wizard segítségével módunkban áll az elkészített scriptből LabVIEW vi generálására.



13. ábra: NI Vision üdvözlő képernyője

6.1.3.1. A Vision Assistant fő funkciói

Acquire Image – Ennek segítségével választhatjuk ki a különböző adatforrásokat, ami lehet az NI által fejlesztett SmartCam, szabványos USB, illetve soros portos vagy hálózati kamera, szimulált beolvasás (előre felvett videó anyag vagy képek gyűjteménye).

Browse Image – Ezzel a funkcióval a beolvasott képek közül kiválaszthatjuk a számunkra szükségeseket, amelyekre a tervezett algoritmusnál szükség van.

Process Image – A script tervezésére szolgáló felület. A képernyő bal oldalán található a funkció paletta, ami tematikusan magába foglalja a szükséges műveleteket. Mellette foglal helyet a script ablak, ahol az algoritmushoz szükséges funkciókat helyezhetjük fel dataflow programozási mintára. Az elkészített scriptet az image browser által meghívott képekre tesztelhetjük, melynek eredménye az aktuális képen látható.

LabVIEW VI Creation – Az elkészült scriptet a Vision Assistant képes vi-já generálni egy varázsló segítségével, ahol kiválaszthatók a látni kívánt paraméterek és a vezérlőelemek.

6.1.3.2. A Vision Assistant alkalmazása

Miután kiválasztottuk a képrögzítő eszközt, a Processing Window-ra lépünk. A referencia ablaktól jobbra található a Processing Window (14. ábra). A script aktuálisan kiválasztott műveletének eredménye látható, és a legtöbb esetben itt adhatjuk meg azt is, hogy a kép melyik pontján akarunk dolgozni.

A bal alsó szekcióban található a Funkciók paletta, amiben tematikusan elhelyezett csoportokban a script elkészítéséhez szükséges műveleteket sorolja fel. Az Image csoportjába tartozó eszközök az alapvető képkorrekciós és képelemzési technikákat vonultatják fel, többek között ilyen a Histogram, Measure Image Correction, Brightness. Ez utóbbi a fényerőn kívül a kontraszt és gamma

értékváltoztatására is lehetőséget nyújt. A következő csoport a Color nevet kapta, ami segítségével a képet alkotó színeket elemezhetjük, vizsgálhatjuk több szempont szerint.

A következő két funkció a szürkeárnyaltos és a bináris képekhez alkalmazható elemeket tartalmazza, többek között szűrési és alakzat felismerési funkciókat foglal magába.

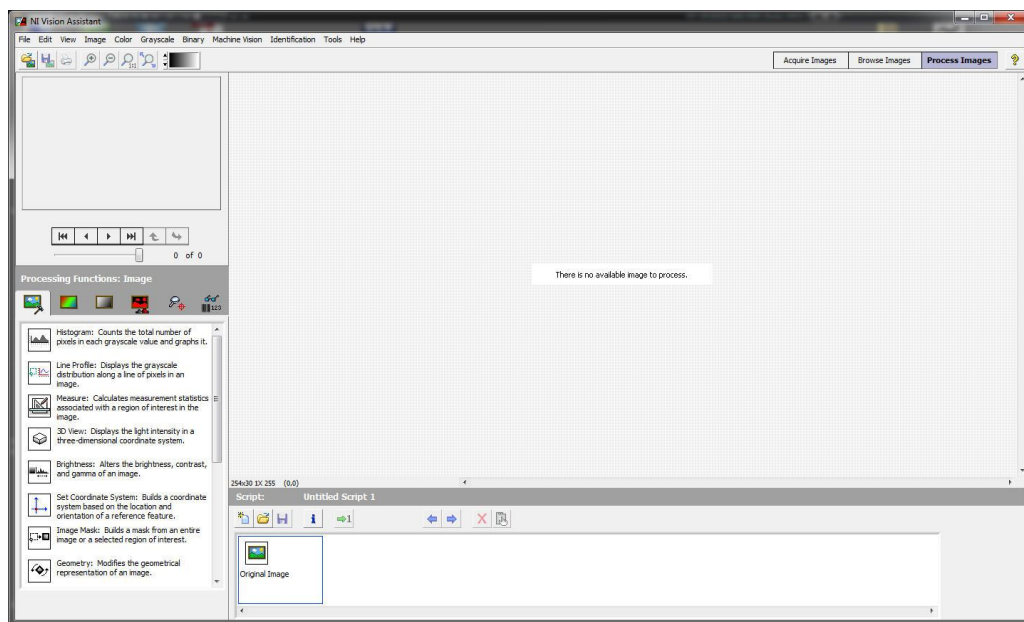
Az utolsó két csoport a Machine Vision és Identification. Az Identification a régi és újonnan megalkotott azonosításra szolgáló kódok kiértékelésére használható elemeket találhatunk, mint például a termékek azonosítására használt vonalkód, és az azt leváltani kívánt QR kód, ami az azonosításon kívül többlet információval szolgál. A Machine Vision a Grayscale-, és a Binary-ban található alak és mintakereséssel ellentétben több képtípusra is alkalmazható eljárásokat tartalmaz, és az azokban nem szereplő elemek itt megtalálhatók. A fontosabb funkciók közé sorolhatók az él-, alakzat- és mintakeresés. Az alakzat- és él-keresési műveletekhez szükséges, hogy a képen kijelöljük azt a területet, ahol mérési eredményt várunk. Ennek a műveletnek az elvégzése sok esetben hasonló, de a terület kiválasztása a feladat típusa szerint változhat, tehát egyenes élkeresésnél, egyenest kell húznunk. Kör keresésénél pedig egy illeszkedő kör paramétereit kell beállítani. Az említett funkciók közé tartozik a mintakeresési folyamat. Erre jellemző, hogy egy elkészített minta segítségével több képen is megtalálható legyen az azonos érdeklődési terület. Például ha egy nyomtatott áramkör képéről egy szekcióján kivett alkatrészt mintának veszünk, akkor azt egy másik ilyen típusú lapról készített képen megtaláljuk. Ezt az eljárást sorozatgyártásnál, a hibás összeszerelési gyártási folyamatok kiszűrésére alkalmazzák. Továbbá, szintén ezzel a technikával módunkban áll nagy mennyiségű képeknél az érdeklődési terület megtalálására is. Ezt a lehetőséget a pályázat során én is alkalmaztam.

A negyedik és egyben legfontosabb szekció a Script Window, amelynek felépítése a következő: Felül láthatjuk az aktuális script nevét és kiterjesztését (*.vascr). Ez alatt található a script létrehozásához és mentéséhez szükséges gombok, valamint a scripthez csatolható leírás kiválasztása, a funkciók törlése és szerkesztése, illetve a műveletek közti léptetés. Külön kiemelendő a „>1” gomb, amely az elkészült scriptet a kiválasztott képre elvégzi.

Alatta található maga a script, aminek kezdő eleme minden esetben a gyűjtőből meghívott, és a referencia ablakból kiválasztott eredeti kép. A funkciópalettáról kiválasztott művelet bekerül a script szerkesztő ablakába, egyúttal beállíthatjuk a szükséges paramétereket. Munkánkat meggyorsíthatjuk, ha a felhelyezett elemek szerkesztésére az elemen történő dupla kattintást használjuk a fent említett szerkesztés gomb helyett.

Az elkészült scriptet elmenthetjük, és a program segítségével bármikor meghívhatjuk. Fontos megemlíteni, hogy a mentési művelet csak az elvégzett műveleteket menti, így pedig a képeket mindig újra és újra meg kell hívnunk.

A Tools menüpont egyéb lehetőségeket is kínál. A Batch Processing segítségével az elkészült scriptet tesztelhetjük általunk megadott mappák képeire, sőt akár aktív eszközökre is (a SmartCam, az NI intelligens kamerája). A tesztelési fázis során megtudhatjuk, hogy a műveletek elvégzése mennyi időt kíván, és a View Measurement opciót választva a kiértékelést is láthatjuk. Talán a legfontosabb és leghasznosabb funkció a Create LabVIEW VI. Ennek köszönhetően egy varázslón keresztül alkalmunk nyílik az elkészített program LabVIEW-ba történő implementálására. A varázsló felkínálja a már mentett scriptek átalakítását, illetve az aktuális legenerálását is. Lehetőséget biztosít a megjeleníteni kívánt paraméterezhető kontrollerek és indikátorok kiválasztására.



14. ábra: A Process Images ablak

7. A programhoz felhasznált egyéni tervezési scriptek

7.1. A mintakeresés

A scripthez szükséges képeket a korábban leírtak alapján, a feldolgozáshoz szükséges módosításokat és leválogatást követően, a szimulált képbeolvasási művelettel a gyűjtőbe helyeztem. A következő feladatként az eredeti kép, fényerő és kontraszt módosításához szükséges Brightness, képe a 15. ábrán, műveletet hívtam meg, aminek szükségességét a következő script leírások alkalmával taglalom.

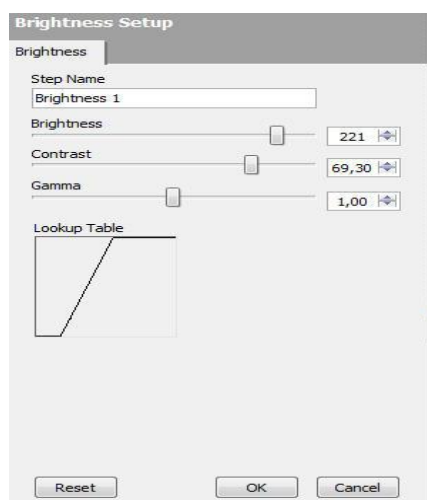
A szükséges változtatásokat elvégezve a „Pattern Matching” (16. ábra) műveleti elemet használtam. Ennek a sajátosságait már korábban bemutattam, most részletesen szeretném ismertetni a működését és beállításait. A meghívás után a funkció paletta helyén egy másik ablak jelenik meg, ahol a paraméterek beállíthatók. A Main fül alatt a funkció egyedi neve adható meg, ami azonosítja

azt a későbbiek folyamán. A Template fület választva definiálhatjuk a programban a mintának használt képet. Készíthetünk újat és szerkeszthetünk már meglévőt, sőt korábban más módszerrel létrehozott képet is meghívhatunk.

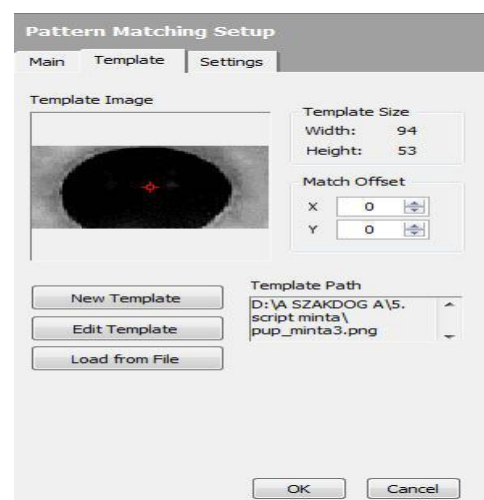
A minta elkészítésében ismét egy varázsló lesz a segítségünkre, ami először lehetőséget nyújt, hogy a képen kiválasszunk a nekünk szükséges részletet. Ezt követően a mintán lehetőségünk van ignorációs műveletre. Ez annyit tesz, hogy a képen kijelölt részleteket a program a minta illesztése során nem veszi figyelembe. A minta felvétele során én is alkalmaztam ezt a lépést, mert a kamera indikátor-, és infraledje a felvétel során visszatükröződött a szemről, ezzel nem volt mód tiszta mérésre. Az ignoráció a kijelölő és ecset funkciókkal végezhető a referencia képernyőn, és a kizárt területet piros maszkolással indikálja program.(17. ábra)

A minta létrehozását követően a program az illeszkedési pont koordinátáit adja vissza, ahol a képzeletbeli [0, 0] pont a minta bal alsó sarka. Ezt a pontot az offset érték megadásával módunkban áll X és Y koordináta pontokba eltolni. A minta beállítása után meg kell határoznunk azt, hogy hány – a kritériumnak megfelelő – egyezést szeretnénk megjeleníteni (Number of Matches to Find), és azt is, hogy minimum hány pixelnek (Minimum Score) kell megegyezni a mintával. Ha nem fix helyzetű a képen elhelyezkedő alakzat, akkor a (Search for Rotated Pattern) beállításaink módosításával a más szögben elhelyezkedő alakzat is megtalálható. A Script lezárása a Set Coordinate System-mel történik, ami a Pattern Matching által kapott koordináta pontokra egy derékszögű koordináta rendszert ültet.

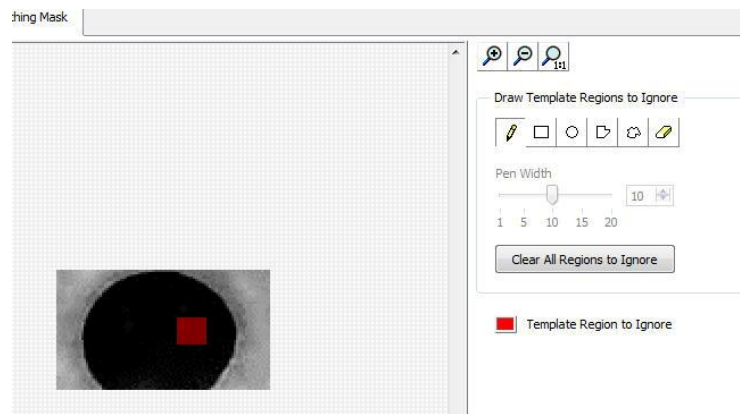
A 18. ábrán láthatók a Script Windowba helyezett dataflow nézetben. A 19. és 20. ábrán látható az első és a hetedik kalibrációs pontra tesztelés.



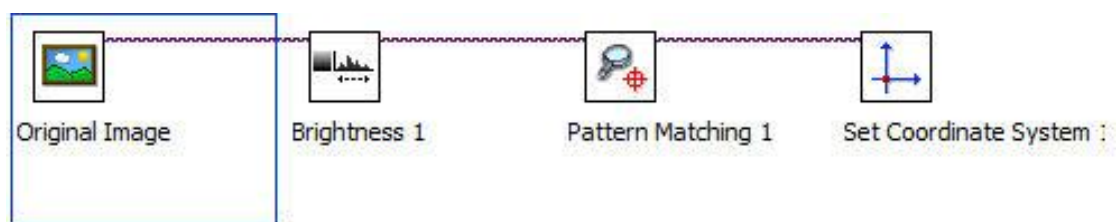
15. ábra:
A Brightness Setup felépítése



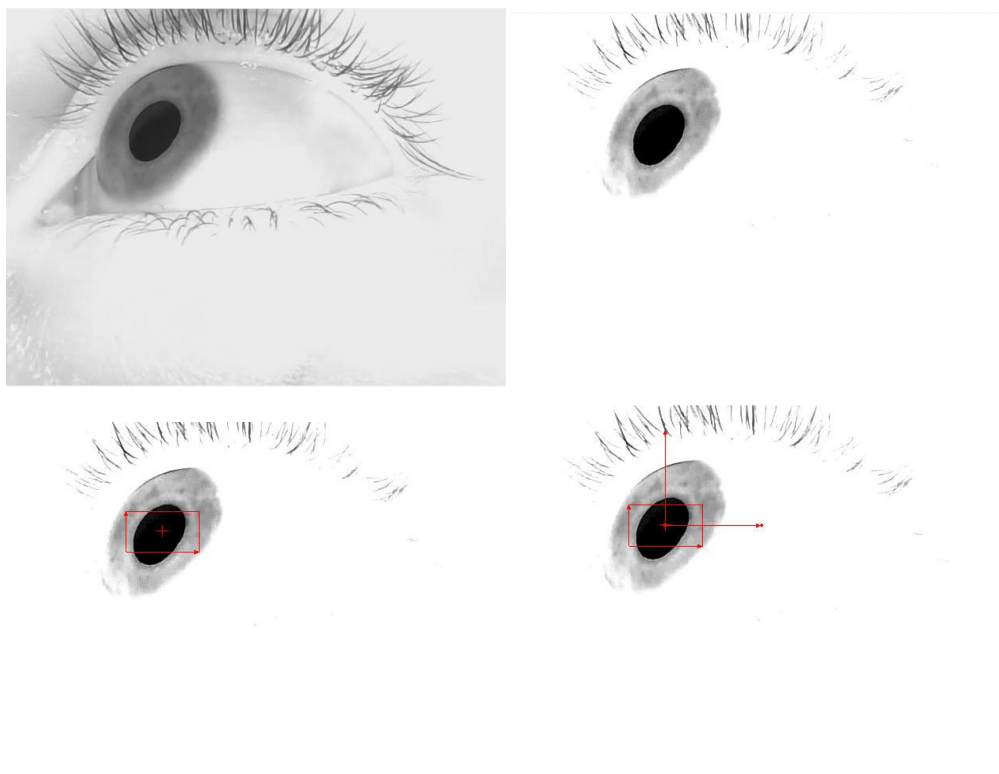
16. ábra:
A Pattern Matching Setup felépítése



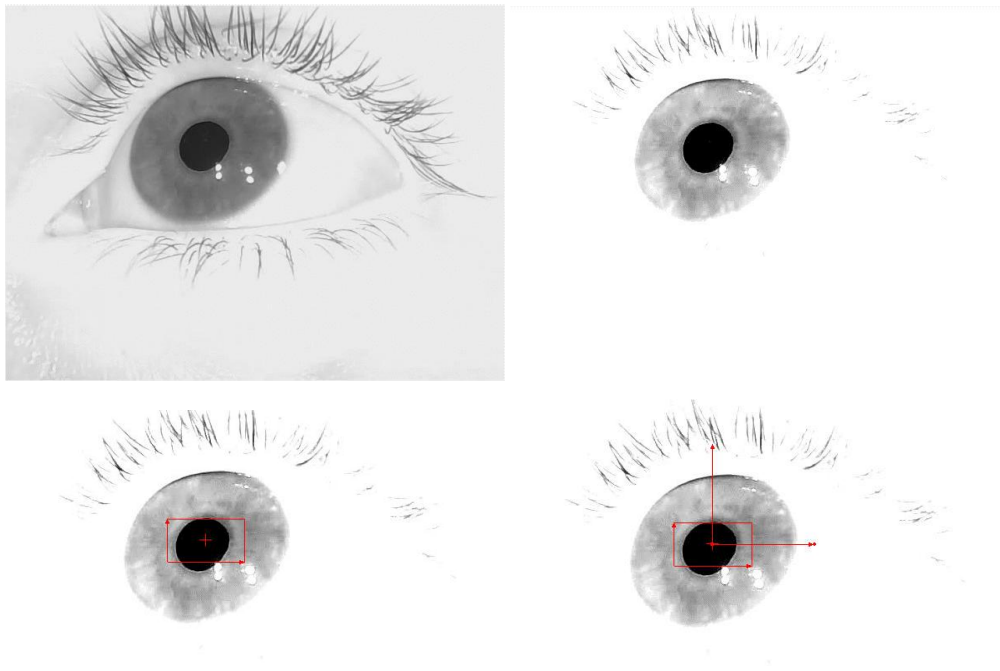
17. ábra:
A Pattern Matching Edit Template funkciója



18. ábra:
A minta keresés algoritmus Vision folyamatábrája



19. ábra:
A mintakeresési művelet bemutatása az első kalibrációs ponton



20. ábra:
A mintakeresési művelet bemutatása a hetedik kalibrációs ponton

7.2. A pupilla megtalálása, avagy a kör alakzat keresése

A script megírása során a már korábban megismert műveletek ismétlődnek, hiszen ahhoz, hogy a két műveletsorozat együtt működhessen, azonos feltételekre van szükség. Ezért a Brightness, mint második művelet és a Set Coordinate System, mint záró elem itt is megtalálható.

Azonban itt a mintaillesztési művelet helyett a pupilla és a pupilla középpont koordinátáinak megtalálása a cél. Annak érdekében, hogy a pupillát a legmegbízhatóbb módszerrel találjuk meg a Find Circular Edge funkciót alkalmazzuk. A művelet alapja, hogy egy körrel kijelölt területen szín intenzitás különbséget figyelembe véve a program a keresési beállításoknak megfelelő kör alakzatot és visszaadja annak középponti koordinátáit és sugarát. Hasonlóan a korábban megismert Pattern Matchinghoz jelen esetben is elnevezhetjük és paraméterezhetjük az eszközt. A Find Circular Edge Setup, képe a 21. ábrán, alatt a következő beállítások állnak rendelkezésre:

A Direction, magyarul irány segítségével megadható, hogy a körkeresés belülről-kifelé, vagy kintről-befelé mutasson. Számunkra itt a belülről-kifelé mutató opció a megfelelő.

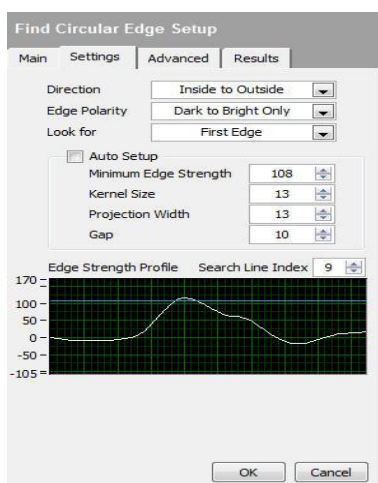
Az Edge Polarity paraméter segítségével megadható, hogy a sötét tónusból a világosba – Dark to Bright Only – vagy a világosból sötétbe –Bright to Dark Only – hajló alakzatokat mutassuk meg, de lehetőség nyílik e szűrési feltétel mellőzésére az All Edge választásával. A pupilla sajátosságait

figyelembe véve a Dark to Bright Only, azaz a sötétből-világosba mutató lehetőség választása szükséges.

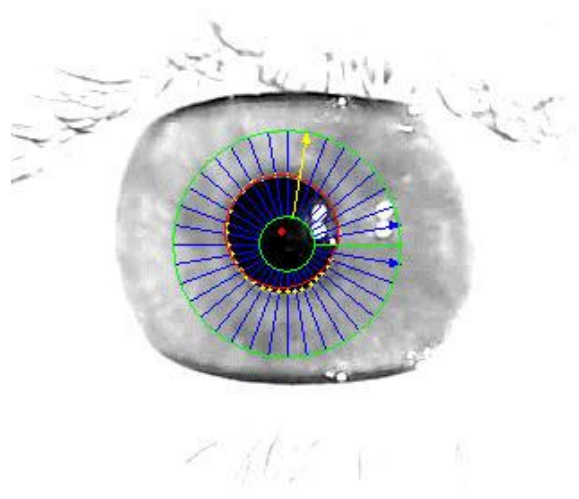
A Look for segítségével meghatározhatjuk, hogy az alakfelismerés során megtalált elemek közül melyiket választjuk. Lehetőség szerint választhatjuk az elsőt vagy a legjobbat.

Az alapvető keresési beállításokat követően, az illeszkedő kör változói találhatók, például az intenzitás különbség, kernel méret vagy mintavételező vonalak közti szög távolság, amiket alapértelmezetten a Vision automatikus állít be. Szükség esetén az értékek manuálisan is beállíthatók.

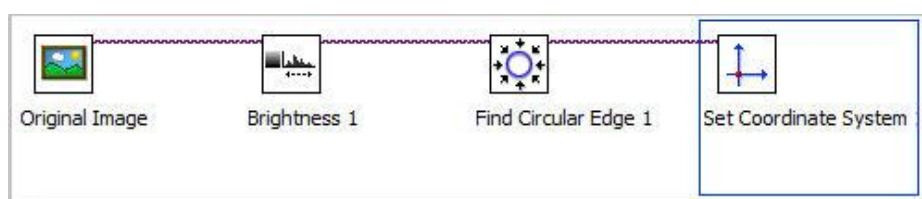
Az Advanced fül alatt található koordinátarendszerben figyelemmel kísérhetjük, hogy az egyes élkereső vonalak milyen erősségű pontot találtak a peremen.



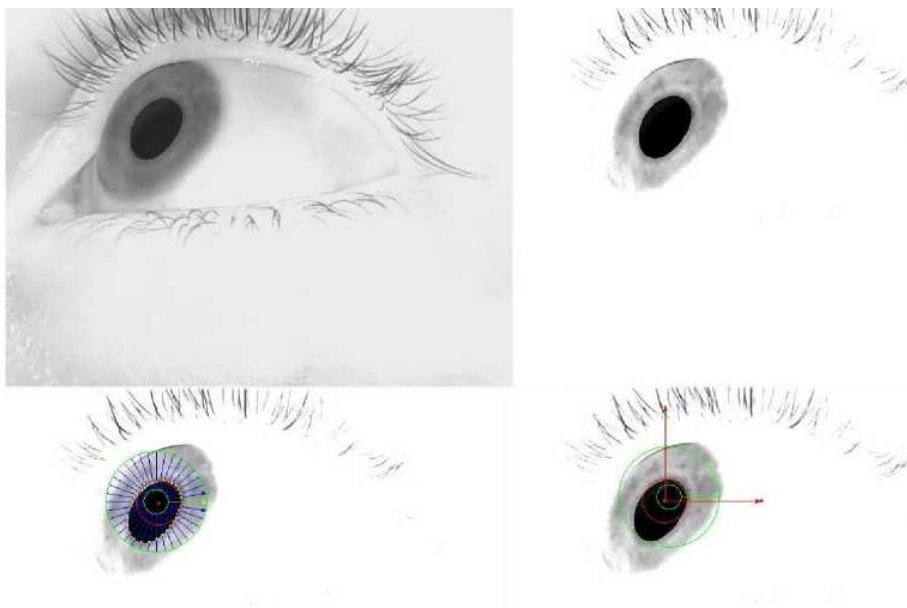
21. ábra:
A Find Circular Edge Setup felépítése



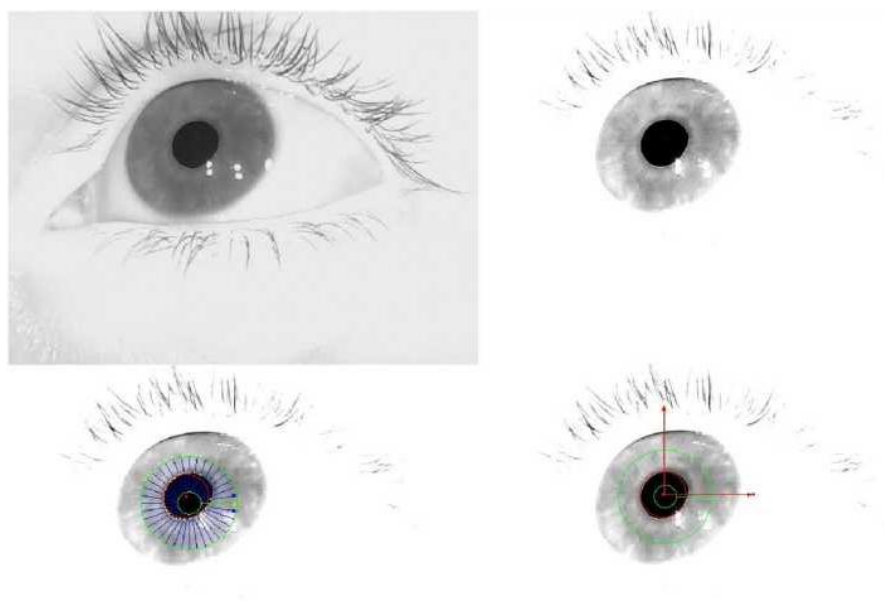
22. ábra:
A Find Circular Edge eredménye



23. ábra:
A kör keresési algoritmus Vision folyamatábrája



24. ábra:
A körkeresési művelet bemutatása az első kalibrációs ponton



25. ábra:
A körkeresési művelet bemutatása a hetedik kalibrációs ponton

8. A LabVIEW 2013 alkalmazásban elvégzett feladatok

8.1. A mintakeresés LabVIEW implementációja

Az előző fejezetben az NI VISION Assistant 2013 segítségével generált vi-okat a LabView 2013 szoftverével nyitottam meg. Ebben a fejezetben a vi-ok részletes bemutatását szeretném tárgyalni. Mindezt a program hierarchiájának alsó fokáról, illetve annak lefutási ciklusának megfelelő sorrendjében.

A leírtak alapján az első a minta keresésére szolgáló vi, a Front Panelen helyezkednek a kontrollerek és az indikátorok. A VISION által beállított értékek a konverzió során megmaradnak, ezért a Front Panel nem igényelt további munkát.

Ezzel ellentétben a Block Diagramon egy kis változást kellett alkalmazni. A generált kódban képbetöltési műveletet törölni kellett, helyére egy File Path, kontroller került. Ennek oka, hogy ezt a programot subvi-ként alkalmazom, és a bementi képet a külső vi-tól kapja meg az alkalmazás. Ahhoz, hogy ez kivitelezhető legyen a Front Panelen a jobb felső sarokban elhelyezkedő VI-t jelképező ikonra az egér jobb gombjával kattintunk, és a Show Connector-t választjuk, majd hozzárendeljük a File Path-t. Itt találjuk a vi-hoz tartozó összes be- , és kimentési értéket. A színek az adattípusokat mutatják, a fehér pontok pedig a szabad pontokat jelölik. Kiválasztunk egy szabad csatlakozót és az egér bal gombjával rákattintva (a kurzor „cérnára” változik, ami a Wiring tool eszközt indikálja) értéket adhatunk neki, ha az egérrel a kívánt elemre kattintunk. Ha megfelelően végeztük el a műveletet, akkor a konnektoron (26. ábra) megjelenik az új bemenet. Abban az esetben, ha több kapcsolódási pontot szeretnénk megadni, de az adott minta nem tartalmaz eleget, a konnektorra jobb egér gombbal kattintva, a Patterns-re pozícionálhatunk és a számunkra megfelelőt kiválaszthatjuk.



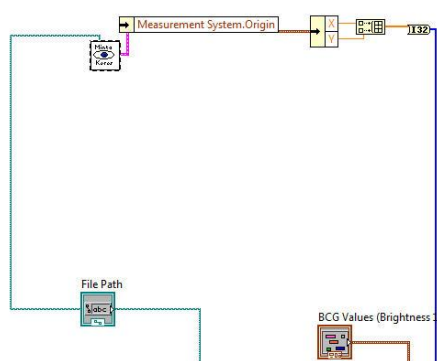
26. ábra: A LabVIEW Connector 7

8.2. A körkeresési algoritmus LabVIEW implementációja

A Vi bemutatása itt is a Front Panellel kezdődik. A korábban említett programhoz hasonlóan itt is a VISION végezte el a feladatok túlnyomó részét. A jobb átláthatóság érdekében az elemeket

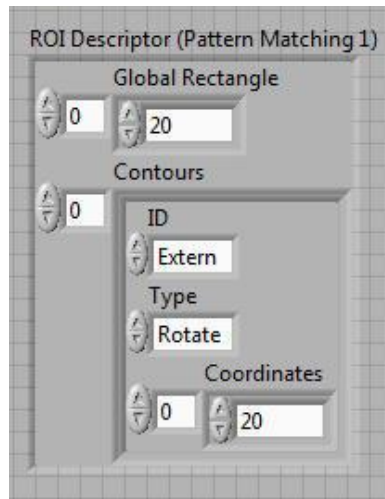
csoportokba rendeztem, de ez a művelet a LabVIEW-nál nem okoz gondot a lefutás szempontjából (nem okoz strukturális-, és adatfolyam változást.)

A program Block Diagramja ennél a vi- nál több figyelmet kíván, mert itt történik a két vi egyesítése. Jelen esetben a képbeolvasás egységét meghagyjuk, itt a beérkező kép elérési útját kötjük a mintakeresés vi-unk egyénileg létrehozott fájl elérési bementére. Ennek kimeneti ágán megjelenik felépített koordináta rendszer, ami 2 cluster típusú elemből épül fel. Ahhoz, hogy a körkeresésnél mindig a megfelelő helyre illesszük a kör mintát ebből a 2 elemű clusterből ki kell nyernünk az x és y koordinátákat. A LabView funkció palettájára a Cluster csoportjából kiválasztjuk az Unbundle By Name elemet, ami a clustert elemeit név szerint azonosítva bontja szét. Ezt követően rendelkezünk az összetett cluster Measurement System. Origin cluster részegységével. Azonban ez nem elegendő, mert így még mindig clusterben állnak rendelkezésre a szükséges koordináták. Az Unbundle By Name művelet megismétlésével a cluster tovább boncolható, és előállítható a két konstans koordináta, amikből a Build Array művelettel, egy egydimenziós tömböt készítünk, ennek a folyamatnak a képe a 27. ábrán szerepel. Ennek fontosságára, a későbbiekben visszatérek.



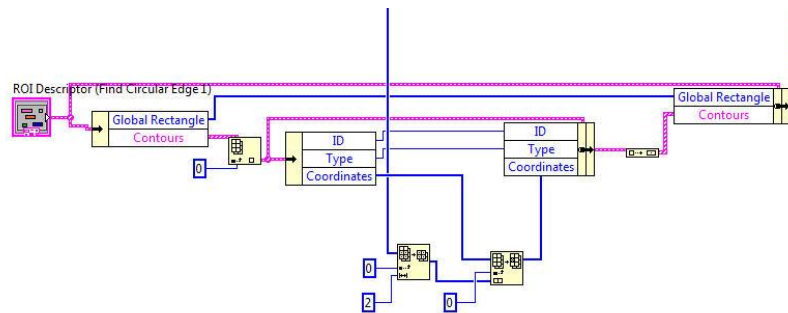
27. ábra: Elérési út átadása a Mintakereső VI számára

A vi-ban kulcsfontosságú szerepet játszik a ROI Descriptor nevű komplex cluster típusú vezérlő elem. A cluster Front Panel kontrollját a 28. ábra mutatja.



28. ábra: A ROI Descriptor képe a Front Panelen

Ez a kontroll határozza meg, hogy hol legyen az illesztett kör középpontja. Az erről gondoskodó koordináták a cluster mélyén találhatóak, ennek kibontásához a már említett Unbundle művelet többszöri alkalmazása szükséges. Az elsővel a ROI Descriptorból egy Global Rectangle egy dimenziós tömböt és egy Contours tömböt kapunk, ami egy clustert tartalmaz. Ennek a nulladik indexű elemét szeretnénk megkapni (itt fontos megemlíteni azt, hogy a tömb indexelése mindig nulláról indul), mert ezen a pozíción tárolódik az aktuális koordináta. A második Unbundle művelettel eljutunk a kívánt mélységre a clusterben. Itt lép be az előbb kinyert koordináta tömb fontossága. A következő programrészlet feladata, hogy kicseréljük az illesztett kör koordinátáit, a mintakeresés által kapott értékekre. Ha jobban megfigyeljük, a szétbontott clusterből kapott Coordinates változó szintén egy egydimenziós tömb, de ennek reprezentációja 32 bites long integer. Ellentétben a fentebb létrehozott tömbbel, ami 32 bites valós elemekből épül fel. Erre a problémára a funkciók palettáról a Numeric funkciók közül a Conversion almenüjéből a To Long Integer eszközzel a megfelelő típusra kényszerítjük az Array-t. Most, hogy rendelkezünk a szükséges koordinátákkal a megfelelő formában és adattípussal, már csak a csere műveletét kell megoldanunk. Ezt a feladatot két részre kell bontsuk, mert a fentről érkező koordináták tömb elemszáma több, mint amire szükségünk van. Ezért az Array Subset művelettel az első két elemét leválogatjuk, és ez a résztömb fog a Replace Array Subset funkció subarray bemeneteként szolgálni, valamint a Coordinates tömb értékeit lecserélni. A művelet befejezéseként a Bundle By Name alkalmazásával, újra felépítjük a clustereket, annyi különbséggel, hogy a Contours Coordinates tömb immáron az aktuális minta koordinátáit tartalmazza. Ezzel az összetett lépéssorozattal elértük, Block diagram képe a 29 ábrán szerepel, hogy az illesztett kör középpontja mindig a pupilla közepén helyezkedjen el, és a ROI Descriptor szerkezeti egysége is megmaradt.



29. ábra: A ROI Descriptor frissítése a kapott koordinátákra

Korábban már említettem, hogy a program egymásra épülő vi-okból épül fel, és ehhez a korábban már bemutatott Connectorok hozzárendelése szükséges. A megfelelő Pattern kiválasztása után a kép elérési útját és a kimeneti koordinátákat a megfelelő pontokra kell kötni.

8.3. A koordináta transzformációs műveletet végző vi bemutatása

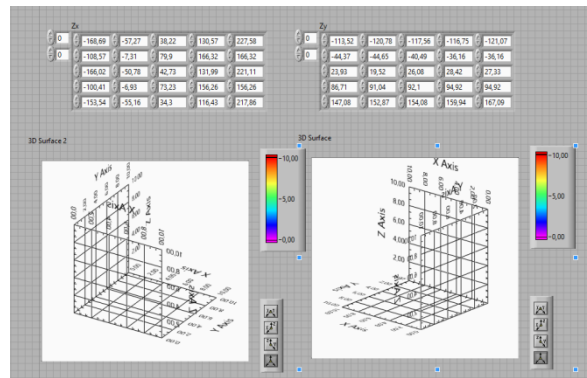
Most, hogy rendelkezünk a kalibrációs pontokhoz tartozó szem koordináta értékekkel egy matematikai eljárás segítségével a valós koordinátákat áttranszformáljuk a képzeletbeli (a monitoron látható) koordináta pontokba. Ezt az eljárást nevezzük interpolációnak. Front Panel képe a 30. ábrán, a Block séma pedig a 31. ábrán tekinthető meg.

Az interpoláció egy közelítő módszer, ahol az ismert koordináták segítségével egy ismeretlen függvény koordinátáit próbáljuk meghatározni. Esetünkben a 23 kalibrációs pont valós és képzeletbeli X és Y koordinátája ismert ezekből. Továbbá ezek segítségével megadható az interpolációhoz szükséges elmozdulási vektor. A koordináta pontok és az elmozdulási vektorok számolása automatikusan történik az ismert mentett pontokból az ehhez szükséges 5x5-ös mátrixok feltöltése is automatizált eljárás volt. A 23 kalibrációs pontból nem lehetett felírni egy 25 elemű mátrixot, ezt kiküszöbölve a 9. és a 18. elem kétszer szerepel egymás után a mátrixban.

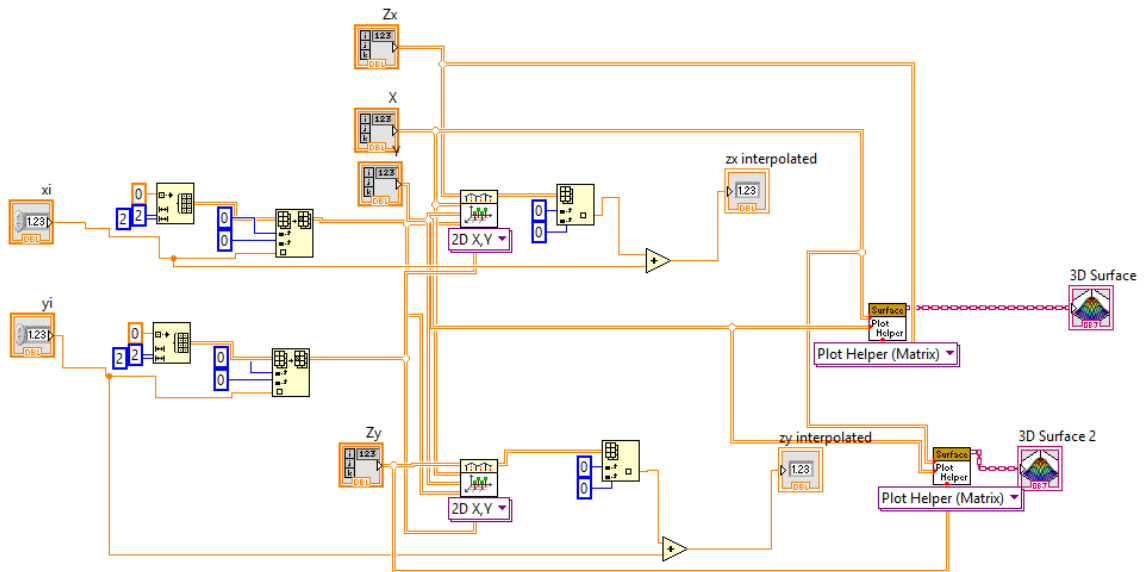
A Block Diagrammon látható elsőre bonyolultnak tűnő program működése egyszerű, a kapott X koordinátaértékeket egy 2 dimenzós tömbben tároljuk, és a fixen definiált pontok és elmozdulási vektorok megadásával a belső interpolációs függvény elvégzi az eljárást. Egy Index Array funkcióval a kimenetre érkező koordinátákat indexeljük, és egy indikátoron megjelenik a valós koordinátaérték.

A jobb szemléltetés érdekében a pontok eltolási vektorainak, valamint az eltolási vektormező indikálására a 32 ábra szolgál. Az ábra megerősít azon állítást, hogy a kalibrációs pontok száma elegendő a terület lefedésére.

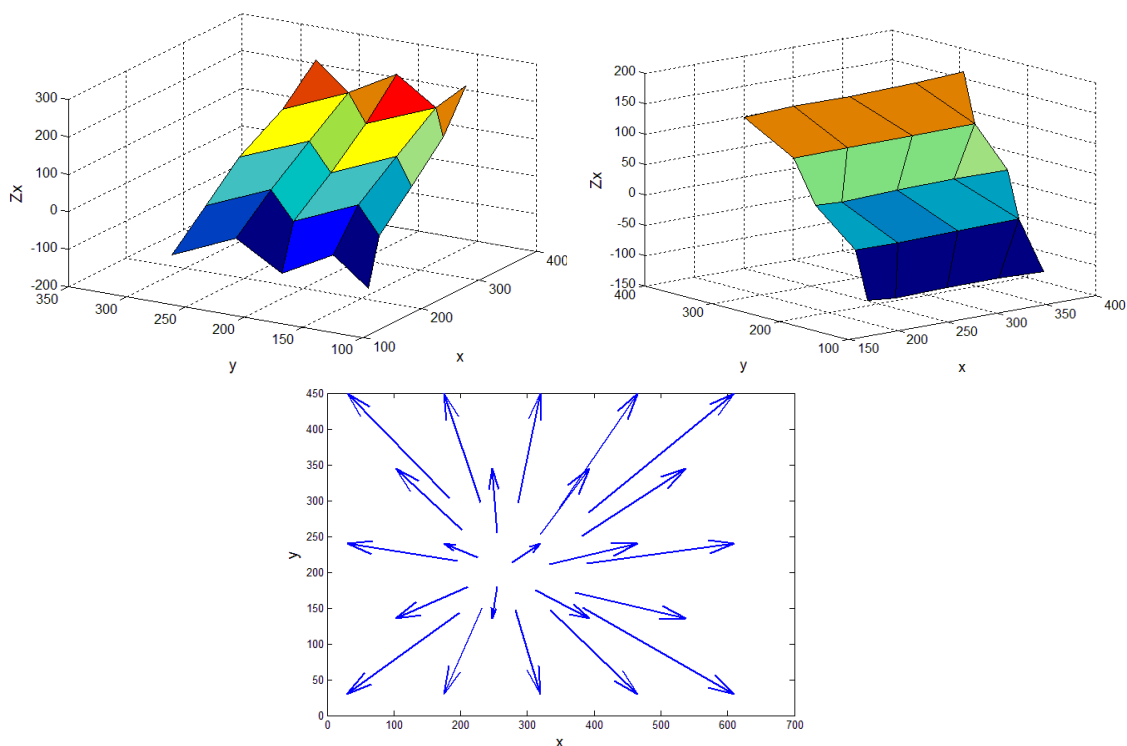
A fenti műveletet az Y koordináták tömbjére is elvégezzük.



30. ábra: A koordináta transzformációs VI Front Panel



31. ábra: A koordináta transzformációs VI Block diagrammjának részlete

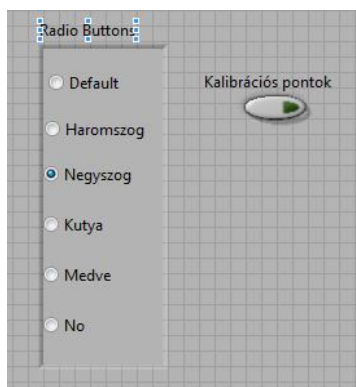


32. ábra: Az x és y irányú eltolások valamint az eltolási vektor mező

9. A főprogram bemutatása

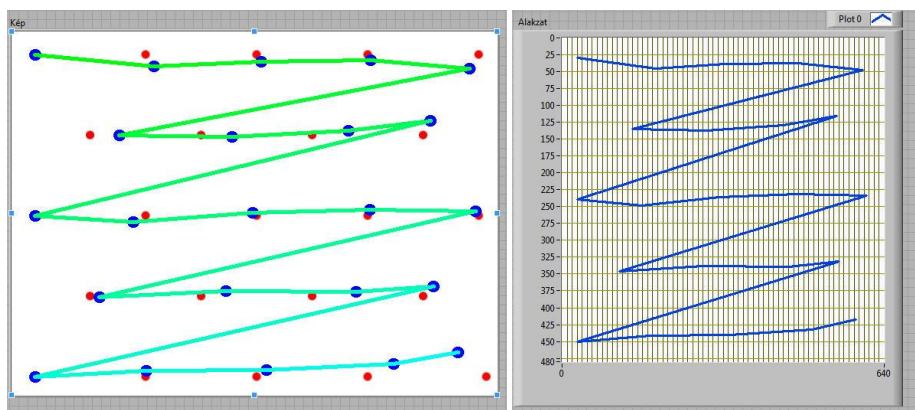
Elérkeztünk a programhierarchia legfelső szintjéhez. A feladat végrehajtását ezen a Formon végezzük és a kiértékelés végső szakasza is itt zajlik. A program Front Panelén a méréshez szükséges képek elérési útjait találhatjuk és a hozzájuk tartozó referencia képeket.

A lehetőségek közül egy Radio Button, képe a 35. ábrán, kontrollér segítségével választhatjuk ki melyik kiértékelését szeretnénk megtekinteni. A fixen megadott elérési utak mellett található az aktuális mintához tartozó képek listája, feljebb pedig az éppen vizsgált kép neve látható.



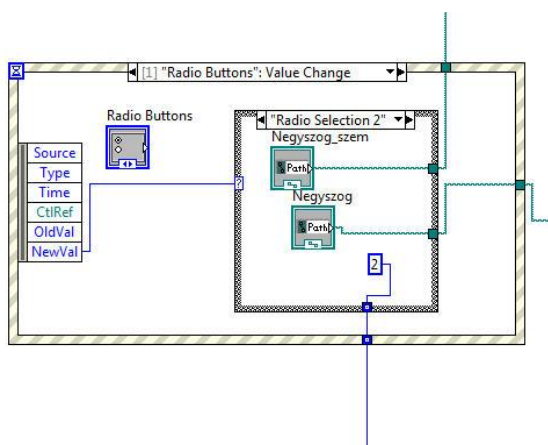
33. ábra: A Főprogram Radio gombos választófelülete

Továbbá helyet kapott még a program felületén 4 tömb kontroll, ami jelen esetben csak szemléltetési cél szolgál. A panelen helyet foglal még egy Image indicator, ahol a kirajzolást végezzük, egy Waveform Chart, ami a hiba indikálására hivatott és egy XY Graph.(34. ábra)



34. ábra: A kalibrációs pontok ábrázolása

A főprogram megértéséhez érdemes a Block sémát részenként tárgyalni, és a megvalósított megoldásokat részletezni. A bemutatni kívánt kép kiválasztásához a korábban már említett Radio gombos megoldást alkalmaztam. A Radio gomb egyedi tulajdonsága, hogy az elemek közül egyszerre egy választható ki, és a kiválasztott elem egy értéket képvisel. A Radio gomb kiválasztása esetén egy értékváltozás esemény generálódik, amit egy event handler fogad, és a Radio gomb kiválasztott elemének értéket egy case struktúra fogadja. Ennek belsejében az alakzathoz szükséges referencia kép elérési útja és a hozzá tartozó rögzített képek mappájának helye van megadva, ez jól megfigyelhető a 35. ábrán

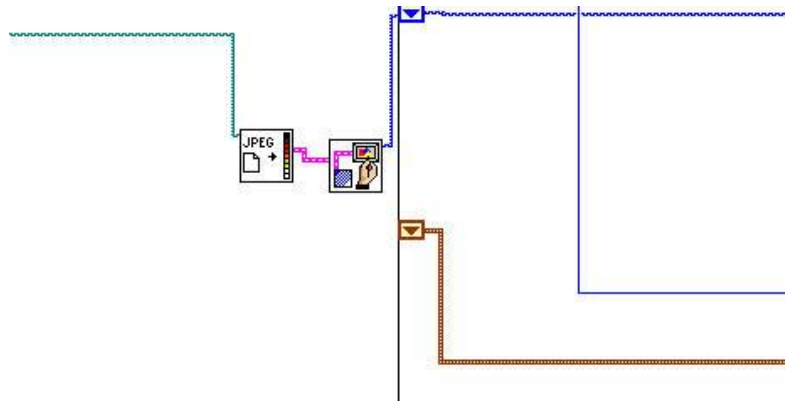


35. ábra: A rádió gombos választás Block Diagrammon

A File Path kontrollerből érkező érték a mappa elérési útvonalát tartalmazza, ezért annak tartalmát egy List Folder funkció segítségével kiolvassuk, és egy Bulid Path-t használva a bázis mappából és a relatív útból valós elérési utat állítunk elő.

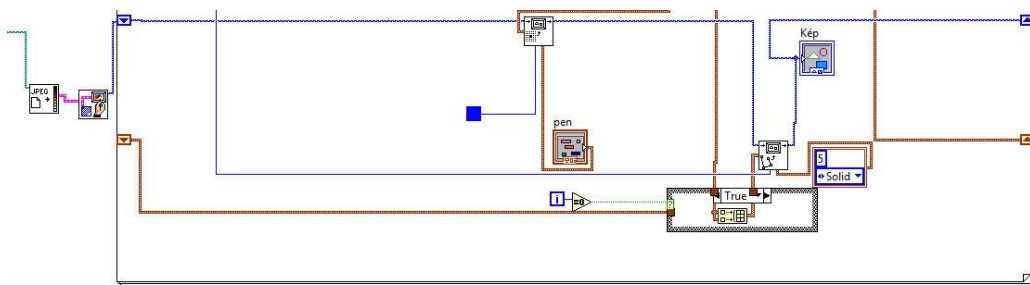
A képbeolvasás folyamatosságát a for ciklus garantálja, és a korábban bemutatott és előkészített subvi.-okat felhasználva megtörténik a kép kiértékelése. A subviok kimenetére, az értékek megmutatására indikátorokat alkalmazunk.

A következő feladatrészt a kép pontjainak kirajzolásáért felelős. A Radio gombok által kínált lehetőségekből válogatva a kiválasztott opció referencia képének elérési útját az azt értelmezni tudó READ JPEG File funkció bementére kötjük. A kimeneten immáron a kirajzolható kép adatai jelennek meg, amit a Draw Flattened Pixmap vi segítségével plottolásra használhatunk.



36. ábra: A kiválasztott kép beolvasása

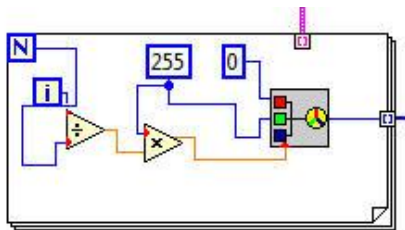
A kiválasztott alakzatra a kiszámított pontokat a Draw Point metódus segítségével rajzoljuk ki. Ennek bemenetei a kép és azon pontok koordinátái, amiket rá akarunk rajzolni, továbbá a pont méretét és színét meghatározó Pen cluster. Ahhoz, hogy a pontok megjelenítése közben ne csak az aktuális pont jelen meg, Shift registert alkalmazunk, s így a már felkerült pontok sem tűnnek el a képről. Annak érdekében, hogy a megjelenítés során még tisztábban látszódjon, merre nézett az alany a pontokat egyenesekkel kötjük össze. A Draw Multiple Line műveletet alkalmazva, aminek bemeneteire a pontrajzoláshoz hasonló paramétereket kötünk. Azzal a különbséggel, hogy az első vonal plottolásánál a rajzolás az abszolút 0,0 koordináta helyett (ez az alapértelmezett beállítás) a kirajzolást csak a második műveletnél indítjuk.



37. ábra: A mért pontok kirajzolására alkalmas Draw Point és Multiple Line

A pontokat összekötő vonal színe az idő tengelyen történő előre haladással árnyalatot vált, s ezzel a program lefutása után is figyelemmel követhető a tekintet által a bejárt útvonal.

A szín változtatásához egy for ciklust alkalmaztunk, amiben az RGB to Color vi bemenetén a képek számát és az iterációs lépések hányadosát felhasználva növeljük a kék szín értékét. Így a sötét tónus jelöli a kezdeti értéket, a világos pedig a lekésőbb megtekintett pontokat.



38. ábra: A szín változtatására szolgáló ciklus

Végezetül a mért pontokra két geometriai alakzatra hibaszámítást végeztünk. Ehhez a művelethez a három- és négyszöget alkotó egyenesek egyenleteire volt szükség. Ezt a feladatot két subvi látja el, aminek bemenetei a főprogramból az X és Y koordináták és a kimenetén a pont egyenes távolság értéke.

Mivel a két alprogram egy analógiára épült, ezért elegendő csak az egyik bemutatása.

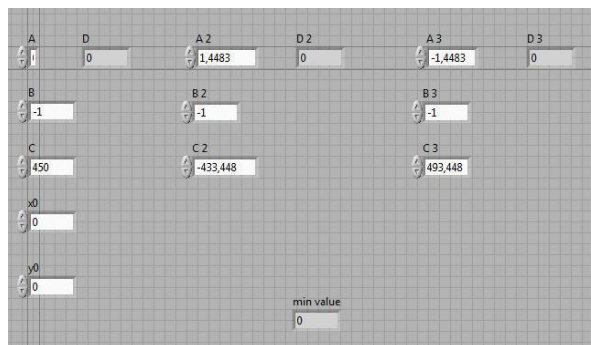
A vi elkészítéséhez szükségünk volt az egyenes egyenletére, és annak kiszámítására.

9.1. A háromszög alakzat pont egyenes távolságának számítása

A probléma egyszerűsége miatt ezt a műveletet kézzel, papíron végeztem el, és a vi- on már csak paraméterezés műveletét kellett elvégezni.

A pont egyenes távolság kiszámításához használt képlet:

$$|A \cdot x_0 + B \cdot y_0 + C| \div \sqrt{(A^2 + B^2)} = D$$



39. ábra: A háromszöghöz tartozó hibaszámító Vi Front Panel

A program működése mind a három egyenesre kiszámítja a pont távolságát, de számunkra csak a legkisebb érték szükséges. Ehhez a három értékből tömböt készítettem, és egy Array Max & Min művelet segítségével kiválasztottam a legkisebbet. Megszokott módon a vi újrafelhasználáshoz a konnektor bekötésével zárult az alprogram elkészítése.

A hibaszámítás indikálásra használt eszköz csak két esetben szolgáltat érdemi információt, így egyéb esetben pedig az indikátor Waveform Chart kimentérea konstans nullát adunk.

A Radio gombon történő változtatást kezelő Event Structure háromszög és négyszög választásával egy konstans átvadás is történik, aminek értéke egy másik eseménykezelő struktúrában kiválasztja, hogy melyik egyenes és pont távolság számítására alkalmas alprogram hívása történjen meg.

10. Kiértékelés

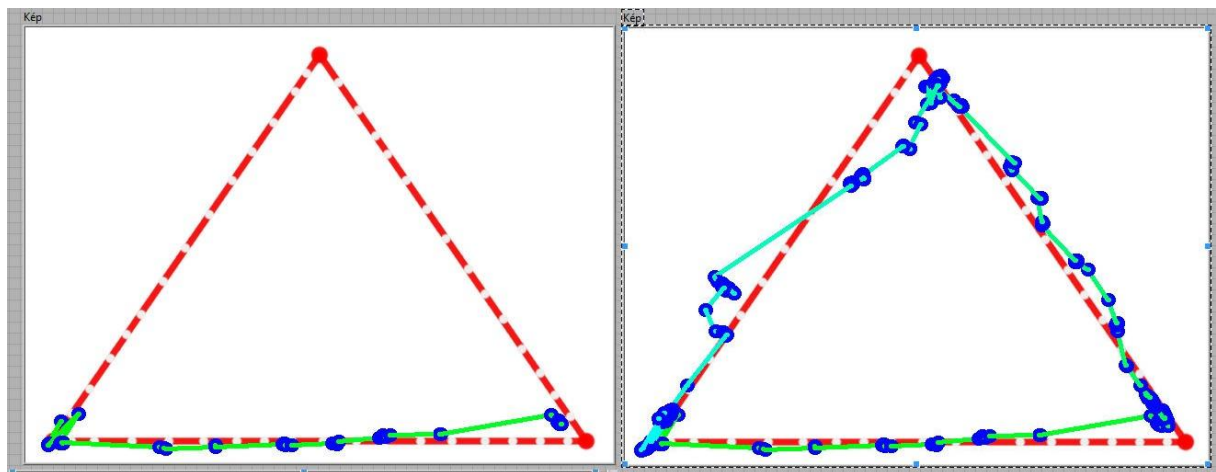
A korábbi fejezetekben már megtörtént a részeredmények ismertetése. A bementi képek feldolgozása megtörtént és a szükséges műveletek elvégzése után ezeket a főprogram Front Paneljén tekinthetjük meg. A főprogram vi ismertetése során bemutattam milyen lehetőségek közül választhatunk. A kalibrációs pont gombnyomás esetén megtekinthetjük, a kalibrációs pontok szemmel történő lekövetését.

A megtekinthető ábrák közül a rádiógombokkal választhatjuk ki, melyiket szeretnénk megtekinteni. A feldolgozott adatokat három indikátor segítségével egy Image, egy Waveform Chart és ahol számoltatunk hibát, ott egy XY Graph mutatja.

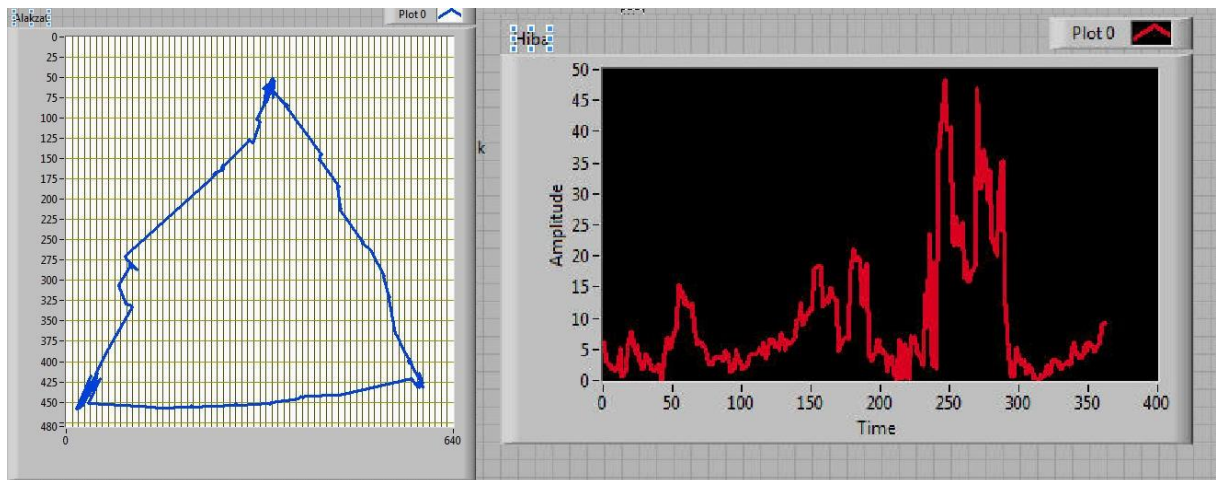
Az Image indikátoron figyelemmel kísérhetjük, hogy az alany tekintete milyen útvonalon halad végig a képen. Továbbá azt is, hogy egyes részleteket hányszor és mennyi ideig nézett meg. A Waveform Chart annak megmutatására szolgál, hogy a megtekintett kép nélkül is lássuk a kirajzolódó alakzat képét.

Két helyen végzünk hibaszámítást, itt a valós és a mért pontok távolságát indikáljuk képkockánként.

Az eredmények a mérést követően megtekinthetőek a 40. -49. ábráig. Érdekesség, hogy a medve nevű képen felfedezhető egy kiugrás, amit egy külső személy megjelenése okozott a mérésben, ezért is említettem, hogy fontos a megfelelő hely megválasztása és a nyugodt környezet.

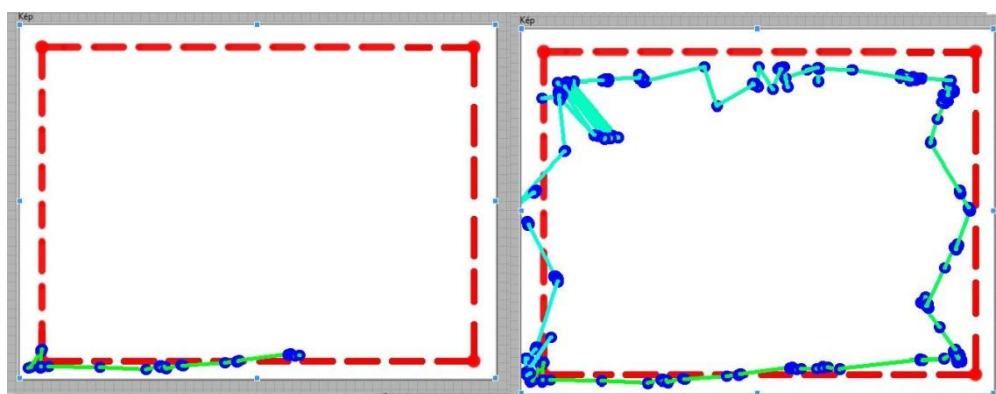


40. ábra: A kirajzolás és annak végeredménye a háromszögre

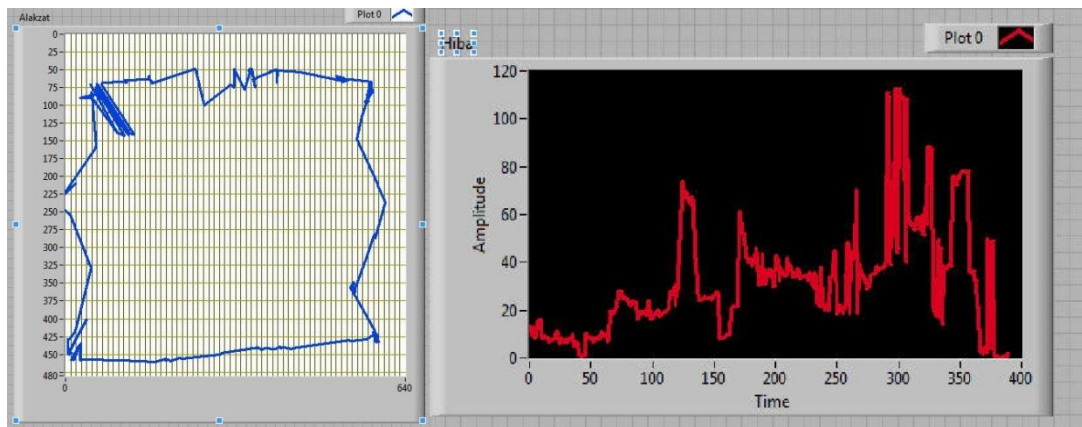


41. ábra: A három alakzata és hibagörbéje

A Négyszög alakzatra mért pontok kirajzolása:

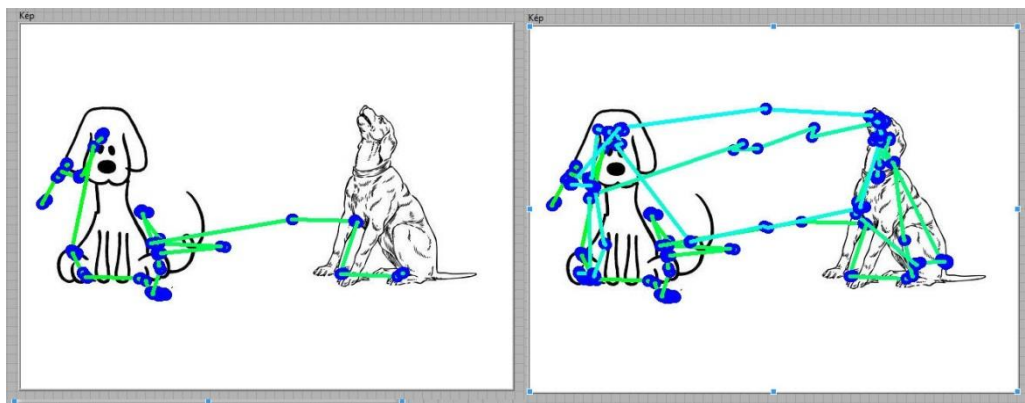


42. ábra: A kirajzolás és annak végeredménye a négyszögre

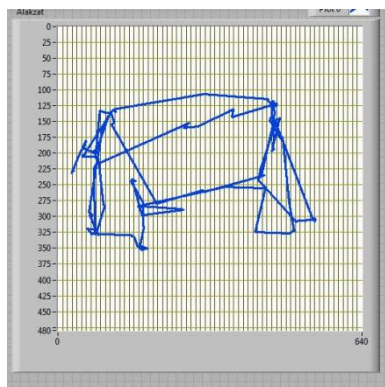


43. ábra: A négyszög alakzata és hibagörbéje

A Kutyák nevű kép kirajzolása:

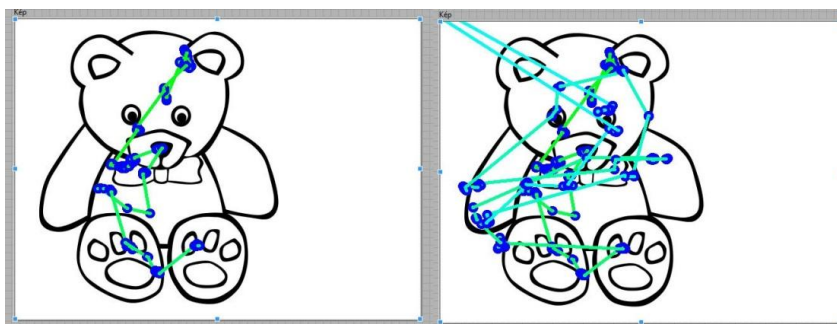


44. ábra: A kirajzolás és annak végeredménye a Kutyák nevű képre

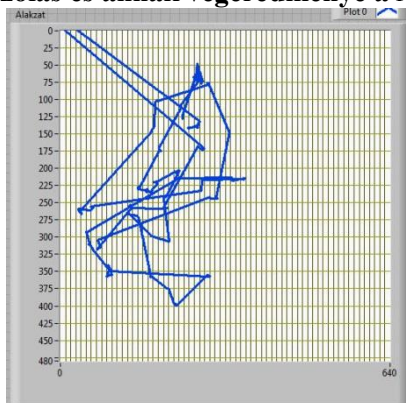


45. ábra: A kutyák nevű kép kirajzolt alakzata

A Medve nevű kép kirajzolása:

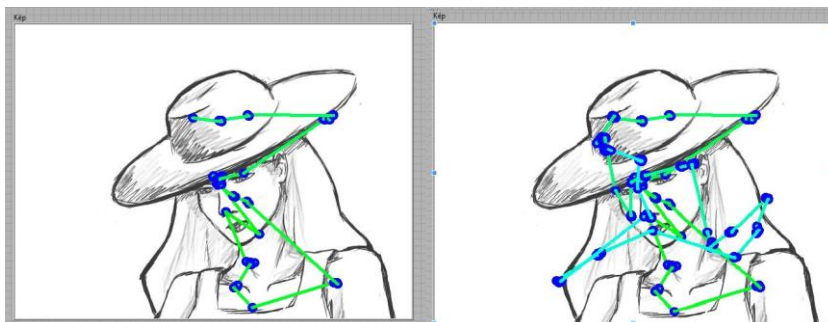


46. ábra: A kirajzolás és annak végeredménye a Medve nevű képre

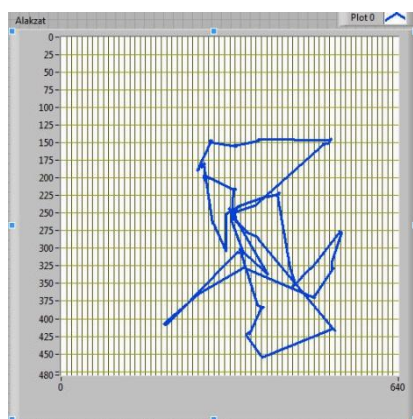


47. ábra: A medve nevű kép kirajzolt alakzata

A Nő nevű kép kirajzolása:



48. ábra: A kirajzolás és annak végeredménye a nő nevű képre



49. ábra: A Nő nevű kép kirajzolt alakzata

11. Konklúzió

Céлом volt egy olyan alkalmazás megalkotása, amely képes az emberi szem mozgásának követésére, a tekintet útjának indikálásra az időben. A rendszer megalkotása közben felhasználtam saját eszközöket és a National Instruments termékeit. Az eredmények igazolják, hogy megbízható méréseket végezhetünk, és az új eszközökkel autonóm mérő egységet készíthetünk.

11.1. Eredmények

Az általam megalkotott mérési kör, valamint egyéni mérési eljárás eredményekkel alátámasztva megmutatja, hogy kellően pontos méréseket sikerült végrehajtanom, és a most folyó kutatásoktól nem sokban eltérő mérési eredményt produkál. Az alkalmazott az eszközök és szoftverek kínálatát kihasználva, személyre szabva és továbbfejlesztve közel automata rendszer sikerült megalkotnom. A pályamunkámat a 2013. évi NI Pályázaton, a szakmai zsűri 1. helyezettel jutalmazta és elismerte, továbbá lehetőségem volt egy nemzetközi szimpóziumon is bemutatni az eredményeimet.

11.2. Jövőbeni tervek

Céлом, hogy a rendelkezésre álló erőforrások felhasználása mellett, egy teljesen automata rendszert alkossak meg, amelyben a szerepel kiterjesztett hibakezelés, valamint felhasználóbarát környezetet és alkalmazási felületet hozzak létre.

Céljaim között szerepel tovább a Pécsi Tudományegyetem orvostani karával történő kooperáció, hogy a bevezetőben említett vizsgálatokat, a saját fejlesztésű alkalmazásommal végezhessem el.

11. Ábrajegyzék

1. ábra: A fejtámasz	- 9 -
2. ábra: NI 1774C Smart Camera.....	- 9 -
3. ábra: Deformált és ideális minta.	- 10 -
4. ábra: Az elkészült mérőkör alannyal.....	- 11 -
5. ábra: A mérési kör.....	- 11 -
6. ábra: A mérési kör két változata	- 12 -
7. ábra: Kalibrációs tábla NI Labview fejlesztőkörnyezettel előállítva	- 13 -
8. ábra: Kalibrációs pontok.....	- 14 -
9. ábra: Alakzat készítése GIMP-ben.....	- 14 -
10. ábra: Az alkalmazás állapotdiagrammja.....	- 18 -
11. ábra: A Vision Builder AI használat közben	- 19 -
12. ábra: Az előforduló verzió kezelési problémák	- 20 -
13. ábra: NI Vision üdvözlő képernyője	- 21 -
14. ábra: A Process Images ablak	- 23 -
15. ábra: A Brightness Setup felépítése	- 24 -
16. ábra: A Pattern Matching Setup felépítése	- 24 -
17. ábra: A Pattern Matching Edit Template funkciója	- 25 -
18. ábra: A minta keresés algoritmus Vision folyamatábrája.....	- 25 -
19. ábra: A mintakeresési művelet bemutatása az első kalibrációs ponton	- 25 -
20. ábra: A mintakeresési művelet bemutatása a hetedik kalibrációs ponton	- 26 -
21. ábra: A Find Circular Edge Setup felépítése	- 27 -
22. ábra: A Find Circular Edge eredménye	- 27 -
23. ábra: A kör keresési algoritmus Vision folyamatábrája	- 27 -
24. ábra: A körkeresési művelet bemutatása az első kalibrációs ponton	- 28 -
25. ábra: A körkeresési művelet bemutatása a hetedik kalibrációs ponton.....	- 28 -
26. ábra: A LabVIEW Connector 7	- 29 -
27. ábra: Elérési út átadása a Mintakereső VI számára.....	- 30 -
28. ábra: A ROI Descriptor képe a Front Panelen	- 31 -
29. ábra: A ROI Descriptor frissítése a kapott koordinátákra	- 32 -
30. ábra: A koordináta transzformációs VI Front Panel	- 33 -
31. ábra: A koordináta transzformációs VI Block diagrammjának részlete.....	- 33 -
32. ábra: Az x és y irányú eltolások valamint az eltolási vektor mező	- 34 -

33. ábra: A Főprogram Radio gombos választófelülete.....	- 34 -
34. ábra: A kalibrációs pontok ábrázolása.....	- 35 -
35. ábra: A rádió gombos választás Block Diagrammon.....	- 35 -
36. ábra: A kiválasztott kép beolvasása	- 36 -
37. ábra: A mért pontok kirajzolására alkalmas Draw Point és Multiple Line	- 36 -
38. ábra: A szín változtatására szolgáló ciklus	- 37 -
39. ábra: A háromszöghöz tartozó hibaszámító Vi Front Panel	- 37 -
40. ábra: A kirajzolás és annak végeredménye a háromszögre	- 39 -
41. ábra: A három alakzata és hibagörbéje.....	- 39 -
42. ábra: A kirajzolás és annak végeredménye a négyszögre	- 39 -
43. ábra: A négyszög alakzata és hibagörbéje.....	- 40 -
44. ábra: A kirajzolás és annak végeredménye a Kuttyák nevű képre.....	- 40 -
45. ábra: A kuttyák nevű kép kirajzolt alakzata	- 40 -
46. ábra: A kirajzolás és annak végeredménye a Medve nevű képre	- 41 -
47. ábra: A medve nevű kép kirajzolt alakzata.....	- 41 -
48. ábra: A kirajzolás és annak végeredménye a nő nevű képre	- 41 -
49. ábra: A Nő nevű kép kirajzolt alakzata	- 41 -

12.Hivatkozások:

- P. Müller - A. Schiffer: *Development of human eye-movement tracking system*, Ninth International PhD & DLA Symposium, 21-22 October 2013, Pecs, Hungary
- Müller P.: *Szemmozgás követő rendszer fejlesztése*, Szakdolgozat, PTE PMMIT, Műszaki Informatika BSc Szak, 2012, Pécs, Hungary
- Müller P.: *Az emberi tekintet követésére alkalmas rendszer megvalósítása LabView-ban*, PTE PMMIK TDK Konferencia, 2012, Pécs, Hungary
- Müller P.: *Szemmozgás követésére alkalmas rendszer fejlesztése*, NI Pályázat 2013, 2013, Debrecen, Hungary

13. Források

- [1] Eye Tracking in User Experience Design, J.R.Bergstorm, A.J. Schall, 2014
- [2] NI VISION-NI Vision Assistane Tutorial. June 2009,372228L, ni.com
- [3] IMAQ-IMAQ Vision for Lab VIEW™ User Manual, June 2003 Edition, Part Number 322917B-01, ni. com
- [4] <http://eye-tracking.hu/blog/eyetrackingrol-bevezetes/>

- [5] Taupe: Visualizing and analyzing eye-tracking data, Benoît De Smet, Lorent Lempereur, Zohreh Sharafi, Yann-Gaël Guéhéneuc, Giuliano Antoniol, Naji Habra, 11 February 2012, Science of Computer Programming 79 (2014) 260–278
- [6] Detection and tracking of eyes for gaze-camera control, Shinjiro Kawato, Nobuji Tetsutani, Image and Vision Computing, Volume 22, Issue 12, 1 October 2004, Pages 1031-1038
- [7] Eye/eyes tracking based on a unified deformable template and particle filtering, Yali Li, Shengjin Wang, Xiaoqing Ding, 25 February 2010, Pattern Recognition Letters 31 (2010) 1377–1387
- [8] Using eye-tracking technology to investigate the redundant effect of multimedia web pages on viewers' cognitive processes, Han-Chin Liu, Meng-Lung Lai, Hsueh-Hua Chuang, 8 July 2011, Computers in Human Behavior 27 (2011) 2410–2417
- [9] A review of using eye-tracking technology in exploring learning from 2000 to 2012, Meng-Lung Lai, Meng-Jung Tsai, Fang-Ying Yang, Chung-Yuan Hsu, Tzu-Chien Liu, Silvia Wen-Yu Lee, Min-Hsien Lee, Guo-Li Chiou, Jyh-Chong Liang, Chin-Chung Tsai, Educational Research Review, Volume 10, December 2013, Pages 90-115
- [10] Dissociating spatial and letter-based word length effects observed in readers' eye movement patterns, Jarkko Hautala, Jukka Hyönä, Mikko Aro, 12 June 2011, Vision Research 51 (2011) 1719–1727
- [11] Robust real time eye tracking for computer interface for disabled people, Alberto De Santis, Daniela Iacoviello, Computer Methods and Programs in Biomedicine, Volume 96, Issue 1, October 2009, Pages 1-11
- [12] <http://nrc.hu/modszerek/eyetracking>
- [13] <http://nrc.hu/modszerek/eyetracking/szemkamera#alkalmazas>
- [14] <http://nrc.hu/modszerek/eyetracking/szemkamera#modszerek>
- [15] <http://www.eyetracking.hu/projekt.html>
- [16] http://nyf.beckground.hu/incoming/kepfeldolgozas/FA_KJjegyzet2.pdf
- [17] <http://hungary.ni.com/labview>
- [18] <http://en.wikipedia.org/wiki/LabVIEW>
- [19] <http://mit.bme.hu/~krebesz/oktatas/vimia312/labVIEW07meres31ab01.pdf>
- [20] http://e-oktat.pmmf.hu/bevezetes_grafikus_programozas