



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Egészségügyi mérnök MSc

Természettudományi Kar

Atomfizika Tanszék

Baumgärtner Margaréta

**PUPILLAKÖVETÉS
MEGVALÓSÍTÁSA EGEREKET
VIZUÁLISAN STIMULÁLÓ
RENDSZERBEN**

KONZULENS

Dr. Maák Pál

BUDAPEST, 2024

Tartalomjegyzék

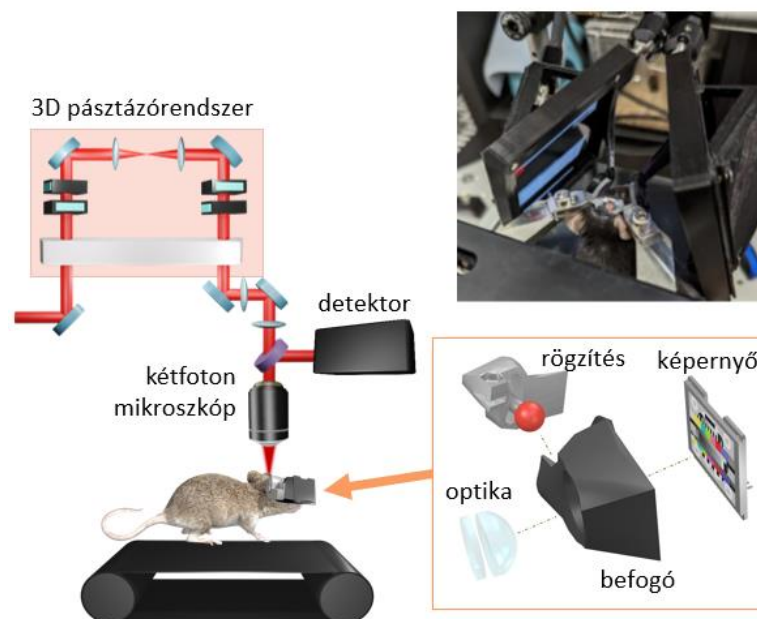
1 Bevezetés	3
2 Célkitűzés	5
3 Irodalomkutatás, hasonló alkotások	6
3.1 Az egér szemmozgásai	6
3.2 Egér video-okulográfia	8
3.3 Szoftveres megoldások áttekintése	10
4 Módszerek	11
4.1 Optikai tervezés	11
4.2 A kiválasztott szoftver bemutatása	12
4.3 Saját kiegészítések	15
4.4 Kamera kiválasztása és szoftveres kezelése	17
4.5 Kísérleti rendszer felépítése	19
4.6 Kombinálás vetítőrendszerrel	20
5 Eredmények	22
5.1 A pupillakövető szoftver működési folyamata	22
5.2 Tesztelés előre felvett videókon	24
5.3 Kísérleti eredmények élő egereken	25
5.4 Kombinálás vetítőrendszerrel	30
6 Megvalósítás alatt álló fejlesztési lehetőségek	32
6.1 Tekintet követése	32
6.2 A pupillakövető szoftver működésének fejlesztése	33
7 Összefoglalás	35
Irodalomjegyzék	37
Függelék	39

1 Bevezetés

A látott információ központi idegrendszeri feldolgozása, a vizuális tanulás folyamata és a látás viselkedésre gyakorolt hatásának vizsgálata mind fontos kutatási területek az idegtudományban. A kísérletekben gyakran alkalmaznak egereket modellállatokként. A sikeres vizuális stimulációhoz figyelembe kell venni az egér látórendszerének sajátosságait, különös tekintettel rendkívül széles látóterének lefedésére és a térbeli mélységérzékelés biztosítására.

A kutatócsoportunk által fejlesztett vetítőrendszer két kisméretű képernyőn két különböző képet vetít közvetlenül az egér szemébe, rekonstruálva a binokuláris látást és lehetővé téve a térbeli mélységérzékelést, valamint teljes kontrollt biztosítva az állatot érő vizuális ingerek felett. Diplomamunkámban ezen rendszert fejleszttem tovább, hatékonyabb optikát tervezek és tesztelem az elkészített eszközt. Az általam továbbfejlesztett vetítőrendszer a számítógépes mérések alapján az egér teljes látóterét lefedi, a látótér mentén állandó képminőséget biztosít, valamint kompakt elrendezésével megkönnyíti a kísérletek kivitelezését. Az eszköz tesztelése élő állatokon jelenleg is folyamatban van.

A kísérleti elrendezésben a vetítőrendszer tartalmaz szemenként egy kisméretű képernyőt, melynek fényét egy-egy lencse fókuszálja az egér szemébe (1.1. ábra). Az elemek rögzítésére 3D nyomtatott tartószerkezet szolgál. Az egér feje rögzített, a kísérlet során nem mozdul el, a mozgás érzetének fenntartása céljából egy futószalag van a lába alatt, a vetítés pedig a futás sebességével szinkronizált. A koponyába beültetett üveglakon keresztül mikroszkóppal követhető az elsődleges vizuális központ agysejtjeinek fluoreszcens aktivitása, úgy, hogy az objektív pontosan a munkatávolságba kerül a megfelelő agyterülettel. A kísérletek alatt párhuzamosan rögzítik az elsődleges látókérgi neuronok aktivitását kétfoton-mikroszkópos felvételekkel, valamint az állatok viselkedését. Közvetlen összefüggés mérhető ki a vetített információ, az egér motorikus válasza, illetve az egyes sejtek aktivitása között. A kísérletek célja leírni a tanulási folyamat során fellépő változásokat a látókérgi neuronok kapcsolatai között. A neuronhálózatok átalakulása szükséges ahhoz, hogy a tanulás hatása a viselkedésben is megmutatkozzon.



1.1. ábra: A kutatócsoportunkban alkalmazott kísérleti elrendezés vázlatos rajzon és fényképen

Tudományos diákköri munkám célja a vetítőrendszer kiegészítése pupillakövetéssel, ennek szoftveres és hardveres kidolgozása. A pupilla méretének változása információt nyújt agyi háttér folyamatokról, a pupilla pozíciójának méréséből pedig rekonstruálható a tekintet iránya. Ennek felhasználásával további információk szerezhetők a vizuális ingerek feldolgozásáról és a tanulási folyamatról, valamint a tekintet irányának ismerete felhasználható a vetítés irányításában is.

A kutatás a BME Atomfizika Tanszékén működő Biophysics of visual systems, perception and information processing (Vizuális rendszerek biofizikája, érzékelése és adatfeldolgozása) kutatócsoportban folyik, amely a BME Szociológia és Kommunikáció Tanszékével együttműködésben a szem optikáját és érzékelési jellemzőit vizsgálja. A projekt során kutatócsoportunk együttműködik a BrainVisionCenter Nonprofit Kft.-vel, célunk a tanulási mechanizmus idegi hátterének megismerése, kísérleti állatok látásmechanizmusa és az ezekhez kapcsolódó agyi folyamatok vizsgálata által.

2 Célkitűzés

Eddigi munkám során részt vettem az egérszem optikai modelljének létrehozásában, részletesen megvizsgáltam a biológiai kísérletekben jelenleg alkalmazott vetítőrendszer képalkotását, és továbbfejlesztettem azt úgy, hogy a vetített kép az egér teljes látóterét jó minőségű képpel lefedje.

Jelen dolgozatban bemutatott munkám célja a vetítőrendszer kiegészítése pupillakövetési funkcióval, lehetővé téve ezzel annak vizsgálatát, hogy a vetített vizuális ingerek hatására hogyan változik a pupilla mérete és a tekintet iránya.

A rendszer szoftveres megvalósításához az alábbi lépések szükségesek:

- a tudományterületen alkalmazott szoftveres megoldások megismerése,
- ezek közül a megfelelő specifikációkkal rendelkező kiválasztása, ha ilyen nincs, akkor valamelyik létező megoldás átalakítása, legvégső esetben új szoftver fejlesztése,
- szoftver tesztelése előre felvett videókon.

A pupillakövetés hardveres összeállításának lépései:

- korábbi optikai modelljeim felhasználásával a rendszer optimális elrendezésének meghatározása,
- a szükséges eszközökkel szemben támasztott követelmények meghatározása, az eszközök kiválasztása és beszerzése,
- a megtervezett rendszer összeállítása,
- a kész rendszer tesztelése elsőként egérszem modellen, majd élő egereken.

Fontos szempont annak megvizsgálása, hogy az elkészített pupillakövető rendszer mechanikai és optikai szempontból integrálható-e az általam korábban tervezett vetítőrendszerbe. Kutatómunkám végső célja az, hogy a vetítés és a pupillakövetés az állatkísérletekben párhuzamosan, egymást nem befolyásolva tudjon hatékonyan működni.

3 Irodalomkutatás, hasonló alkotások

3.1 Az egér szemmozgásai

A szem mozgásainak alapvető funkciója a tekintet irányítása, valamint mozgás közben a látott tárgyak képének stabilizálása a retinán. A szemmozgások az agyi funkciók széles skálájára, érzékszervi, motoros és kognitív funkciókra is betekintést nyújtanak, ezért mérésük fontos eszköz az idegtudományi kutatásokban [1]. Az emlősök szemmozgásai rendkívül sokfélék, többféle szempont alapján csoportosíthatóak és jellemezhetőek. A főemlősök és az egerek közötti különbségek fő oka, hogy az egerek szemmozgásai minden esetben fejmozgásokkal csatoltan, azokkal egyidejűleg fordulnak elő [2]. Az egerek a mozgó objektumokat tehát nem a szemeikkel, hanem a fejükkel követik [3].

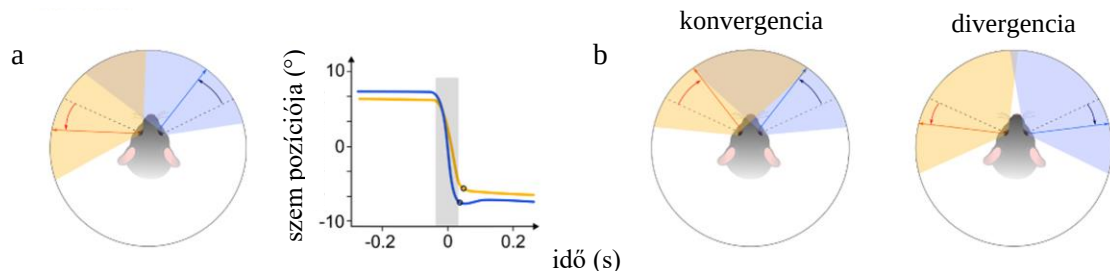
A tekintetet stabilizáló szemmozgások szerepe a fejmozgások kompenzálása, a retinán létrejövő kép stabilizálása és ezzel éles látás lehetővé tétele [3]. Két fázisból állnak, a lassú fázisban a tekintet távolodik a megfigyelt objektumtól, a gyors fázis visszahelyezi a tárgyat a tekintet középpontjába [2]. A tekintetváltó mozgások a szemet egy új célobjektumra irányítják.

A szemmozgások kategorizálhatóak az alapján is, hogy kiváltásuk akaratlagos vagy reflexes, egerekben minden akaratlagos szemmozgás reflexesen is létrejöhet [2]. Az egerekben is jelenlévő akaratlagos szemmozgások a vergencia és a szakkádok, reflexesek pedig a vesztibulo-okuláris reflex és az optokinetikus reflex. A szemmozgásokat magyarázó ábrákon a szemek látómezői sárga és kék színnel vannak jelölve, a szaggatott vonal a vizuális tengelyt jelöli. Az optikai áramlás a látótér pontjainak látszólagos elmozdulását jelenti, oka a szem és a környezet relatív elmozdulása [4].

A szakkádok a tekintet fixációi közötti gyors tekintetváltások létrehozásáért felelősek, lehetnek vízszintes, függőleges vagy ferde irányúak [5]. A 3.1. ábra a. részén egy óramutató járásával ellentétes irányú szakkád látható, a színes vonalak a szem végső pozícióját jelölik. A grafikonon a szemek pozícióját ábrázolták az idő függvényében, szürkével kiemelve a szakkád alatt végbemenő gyors változást.

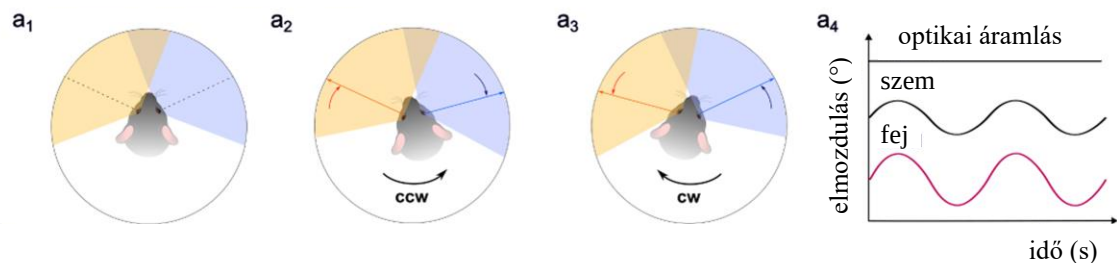
A vergencia alatt a szemek ellentétes irányba mozognak (3.1. ábra, b.), az ilyen szemmozgásokat nem konjugálnak nevezik. Konvergencia során a szemmozgások

temporális-nazális irányúak, a binokuláris látótér kiszélesedik. Divergencia esetén a nazális-temporális irányú mozgások a binokuláris látótér keskenyedését eredményezik [2].



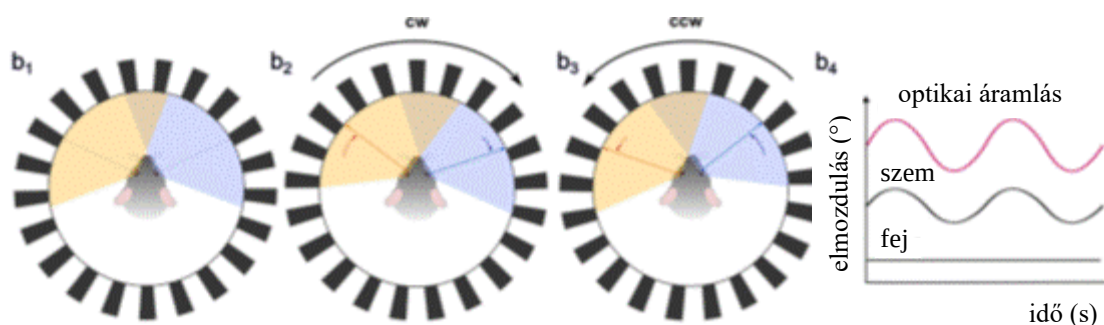
3.1. ábra: a. Szakkád és b. vergencia sematikus ábrája [2]

A vesztibulo-okuláris reflex a tekintetet stabilizálja úgy, hogy a szemek a fejjel ellentétes irányba mozognak. A reflex működését rögzített fejjű vagy testű egerekben a 3.2. ábra mutatja be. Kezdetben a szem nyugalmi helyzetben van (a_1), majd a fej óramutató járásával ellentétes (CCW, a_2) vagy megegyező (CW, a_3) irányú elmozdulása konjugált (a két szemben azonos irányú) szemmozgásokat okoz, amelyek az elfordulás irányával ellentétesek. Ennek eredménye (a_4), hogy nincs optikai áramlás, a látótér nem változik, a szemek és a fej szinkronizáltan mozognak.



3.2. ábra: Vesztibulo-okuláris reflex sematikus ábrája [2]

Az optokinetikus reflex akkor jön létre, ha a retinán a látómező nagy része elmozdul [6]. A 3.3. ábraán a szem kezdetben nyugalmi helyzetben van (b_1), majd a fekete-fehér sávok elfordulása a fordulással azonos irányú szemmozgásokat eredményez (b₂ és b₃). Ebben az esetben az optikai áramlás a szemmozgásokkal szinkronizáltan változik, a fej nem fordul el.



3.3. ábra: Optokinetikus reflex sematikus ábrája [2]

Mivel egerekben a szem- és a fejmozgások csatoltak, különböző szemmozgások figyelhetők meg rögzített fejű és szabadon mozgó állatokon. Az eddig leírt változatos mozgástípusok csak a fej szabad mozgása esetén tudnak létrejönni. Rögzített fejű egereken egyedül kismértékű, szakkádokhoz hasonló horizontális szemmozgások figyelhetők meg, amelyek mindig megkísérelt fejmozgással társulnak [7]. A fej rögzítése azonban gyakran szükséges megkötés a biológiai mérésekben, mivel a megfelelő pontosságú adatgyűjtés sokszor csak így lehetséges, például kutatócsoportunkban a sejtszintű mikroszkópos mérések is így történnek. A vetítőrendszer, ezáltal a tervezett szemkövetés is rögzített fejű állatokon kerül alkalmazásra, ezért várhatóan nem fogunk látványos szemmozgásokat érzékelni.

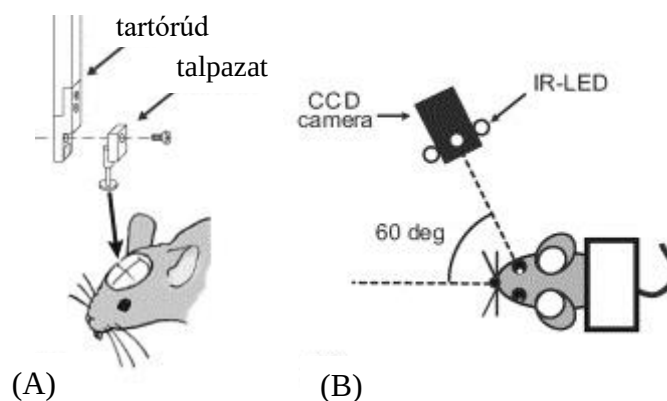
A szemek mozgásán túl a pupilla méretének időbeli változása is értékes kimeneti paramétere a szemkövető kísérleteknek. Ezt elsősorban a fényerősség határozza meg, ezen kívül azonban hatással van rá az idegi aktivitás és magasabb kognitív folyamatok (például figyelem, emlékezés, fájdalom) is [8]. A pupillaméret változásának időbeli mérése tehát alkalmazható az agy állapotának jellemzésére, fajtól függetlenül [8].

3.2 Egér video-okulográfia

A szemmozgások és a pupilla tágultsága a humán neurológiai vizsgálatoknak is fontos paraméterei, így nem meglepő, hogy méréseikre számos módszer alakult ki az évek során. Az egyik legmegbízhatóbb, nem invazív technika a video-okulográfia, amelyben az adatok rögzítése kisméretű videokamerákkal, a videó feldolgozása pedig valamilyen számítógépes szoftverrel történik [9]. A módszert, bizonyos módosításokkal, állatkísérletekben, egereken is alkalmazzák. A mérések fő nehézségét az egér szemének nagyon kicsi mérete jelenti [1], átmérője csupán 3,4 mm [10] körüli. Az okulográfia előnye a viselkedésalapú kísérletekkel szemben a kvantitatívabb elemzés lehetősége,

mivel a szenzoros képi bemenet és a kimeneti szemparaméterek is könnyen mérhetőek [11].

Az egér video-okulográfia kivitelezésének fontos aspektusait de Jeu [11] protokollja alapján foglalom össze. A szemmozgások méréséhez az egér fejét rögzíteni kell, az ehhez szükséges tartóeszközt a koponyába kell műteni (3.4. ábra, a.). A pupilla megvilágítása infravörös fényforrással lehetséges, ami nem zavarja az egeret, mivel nem lát ezen a hullámhossztartományon [12]. Az okulográfiás rendszer alapja szemenként egy, az egér szemére fókuszált infravörös CCD kamera, valamint infravörös fényforrások (3.4. ábra, b.). A kísérleti rendszer része még egy gramfon, futószalag, vagy egyéb felület, amin az egér mozogni tud. A vizuális információ vetítése itt egy nagyméretű henger alakú képernyőre történik, ami a futószalagot körülveszi, saját elrendezésünkben ehelyett saját vetítőrendszerünket fogjuk használni. A vetített kép frissítése a gramfon elmozdulásával szinkronizáltan történik. A kamera a szem mozgásait két dimenzióban látja, azokat csak elmozdulásként érzékeli. A kísérletek előtt kalibráció szükséges, ami alapján a mért elmozdulás visszaalakítható a szemgolyó elfordulásaivá. A videó feldolgozásához egy szemkövető szoftver szükséges, a mérés kimeneti adatai a fényforrások szaruhártyán történő visszaverődéseinek térbeli pozíciói (a továbbiakban cornea reflexiók), a pupilla pozíciója és mérete. Ezek az információk a gramfon vagy futószalag pozíciójával (ami az egér mozgásáról nyújt információt) és a vetítés paramétereivel együtt kerülnek kiértékelésre. Az adatok feldolgozásakor figyelembe kell venni a pupillakövető rendszer késleltetését is. A cornea reflexiói a kalibrációhoz szükségesek, valamint felhasználásukkal a tekintet iránya is meghatározható [13].



3.4. ábra: Egér video-okulográfiás kísérlet vázlatos rajza [14] A. A fej rögzítésének egy lehetséges módja, B. Kamera és infravörös fényforrások elhelyezése

3.3 Szoftveres megoldások áttekintése

Számos különböző video-okulográfiához kifejlesztett pupillakövető szoftver fellelhető, léteznek mind kereskedelmi, mind ingyenes formátumúak. Elsősorban olyan megoldásokat kerestem, amelyek nyílt forráskódúak, hogy szükség esetén működésük módosítható legyen. A szoftverek nagy része embereken történő alkalmazásra készült, viszont fontosnak találtam olyat választani, amit egéren is teszteltek. A fellelt szoftvereket változatos paradigmájú programnyelvek és algoritmikus megoldások jellemezték. A következőkben az általam áttekintett lehetőségek közül csak azt a néhányat szeretném röviden bemutatni, amelyek közül a továbbiakban használt szoftver kiválasztásra került.

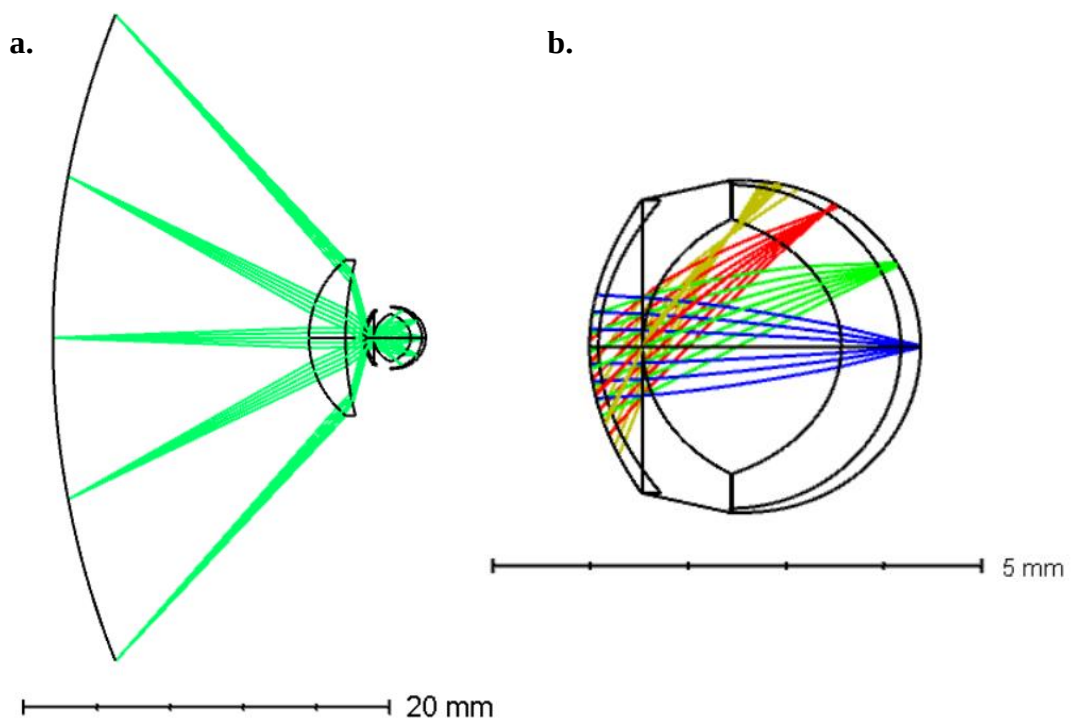
A *PyGaze* egy nyílt forráskódú szoftvercsomag Pythonhoz [15], amivel kevés programozással lehet pupillakövetési kísérleteket végezni, hátránya azonban, hogy csak bizonyos gyártók emberi alkalmazásra készített szemkövető rendszereivel kompatibilis, önmagában nem használható. Az *Open Iris* [16] C# programnyelvben készült, kamerakép és korábban felvett videó feldolgozására is alkalmas, ami a kutatásunk számára előnyös funkció. Mindemellett jól kidolgozott grafikus kezelőfelülettel rendelkezik. Továbbá binokuláris szemkövetést is lehetővé tesz azáltal, hogy egyszerre két kamerát is tud kezelni. Állatokon való alkalmazhatóságát azonban semmilyen adat nem támasztja alá. Dedikált egér video-okulográfiához készített képfeldolgozó szoftvert Sakatani és Isa [14], amely alkalmas a hirtelen, gyors szemmozgások követésére is. A szoftver LabView-ban készült, ezt a programnyelvet a kutatócsoportunkban nem használják, ami tudásmegosztási kihívást jelent, amennyiben a pupillakövető szoftvert a későbbiekben egy nagyobb rendszerbe kell integrálni. Az *EyeLoop* [17] Pythonban készült, kifejezetten kutatási célokra fejlesztett pupillakövető alkalmazás. Emberre, egérre és más állatokra is validálták, a vizsgált állatfajnak megfelelően kiválasztható, hogy az algoritmus kört vagy ellipszist illesszen a pupillára. Alkalmas felvett videó utólagos elemzésre is. A kevésbé kidolgozott grafikus kezelőfelülete ellenére, alkalmazhatósági előnyeinek köszönhetően a kutatás során ezt a szoftvert alkalmaztam.

4 Módszerek

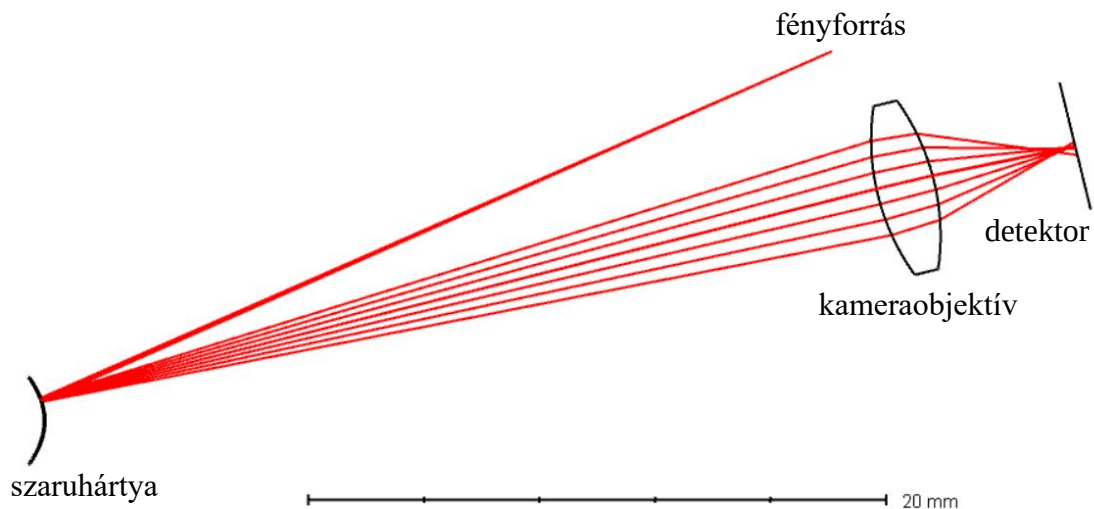
4.1 Optikai tervezés

A pupillakövető rendszer megtervezését optikai modellezéssel kezdtem. Ennek az általam korábban megtervezett vetítőrendszerrel együtt kell működnie, amelynek modellje egy görbített képernyőből és a belőle érkező sugarak fókuszálására egy pozitív meniszkuszlencséből áll (4.1. ábra, a.). A rendszerek tervezéséhez saját egérszem modellemet is alkalmaztam, amelyet korábban készítettem irodalmi adatok alapján [10], [18], (4.1. ábra, b.). Optikai tulajdonságai mellett a szemmodell látótere is megfelel az egérének, ennek nagysága $160-180^\circ$ [19] között van egy szemre.

Először a pupillakövetést a vetítőrendszer modelljében szerettem volna megvalósítani, azonban a tervezés bonyolultsága miatt végül nem a vetítőrendszert egészítettem ki a pupillakövetéssel, hanem arra egy külön tervet készítettem, amely a szemmodellből csak a szaruhártyát tartalmazza egy tükröződő felületként. A rendszerben a kamerát egy gyűjtőlencse és egy detektorsík modellezi, a fényforrás a kamera mellett helyezkedik el, a szemtől 3 cm távolságra (4.2. ábra).



4.1. ábra: a. Vetítőrendszer optikai modellje, b. egérszem modell



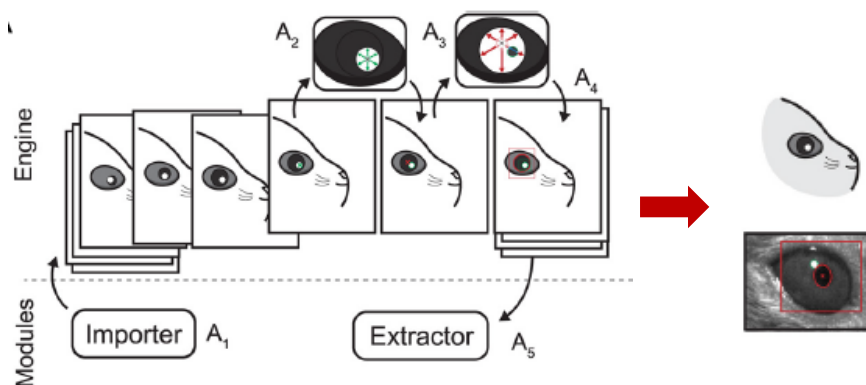
4.2. ábra: Pupillakövető rendszer optikai modellje

A két modellezett rendszer egymásba integrálása egy olyan dichroikus tükörrel valósítható meg, amelyik a látható fényt (a képernyő fényét) átengedi, az infravörös fényt (a szemet megvilágító LED-ek fényét) pedig visszaveri.

4.2 A kiválasztott szoftver bemutatása

A következőkben a kiválasztott szoftver működését és fő funkcióit ismertetem a fejlesztők, Arvin és munkatársai [17] alapján.

Az *EyeLoop* két fő egységből áll: egy motorból (*Engine*, 4.3. ábra, A_{2-4}) és több külső modulból. A motor hajtja végre magát a detektálást: felismeri a képkockákon a pupillát és a cornea reflexiókat. A külső modulok a motor működését támogatják és az adatkezelésért felelősek: az *Importer* modulok (4.3. ábra, A_1) a feldolgozandó képsorozatot importálják a rendszerbe, az *Extractor* modulok (4.3. ábra, A_5) pedig adatokat gyűjtenek és exportálnak. Egy további modul a grafikus felhasználói felület



4.3. ábra: EyeLoop szoftver működésének áttekintése [17]

(*Graphical User Interface, GUI*), amely billentyűparancsok és egérkattintások használatával lehetővé teszi a felhasználók számára az interakciót a szoftverrel.

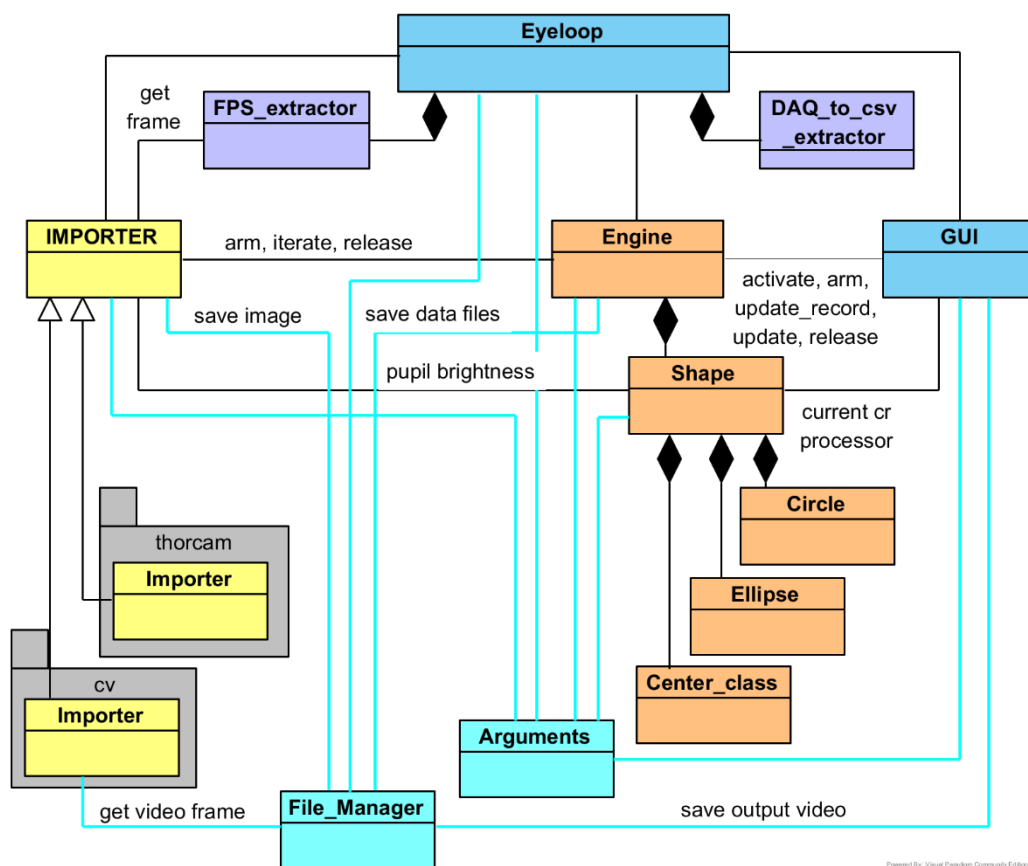
A motor a képeket képkockaként dolgozza fel. A beérkezett képkockán elsőként binarizálást és Gauss-simítást hajt végre, ezek paramétereit a felhasználónak kell beállítania. Az ideális beállítások megtalálásához a grafikus felhasználói felület vizuális visszacsatolást biztosít, a feldolgozott képeken a változtatások rögtön megjelennek. A cornea fényreflexióinak és a pupilla középpontjának koordinátáit egér-rákattintással kell meghatározni, majd ezen becslés alapján a szoftver érzékeli ezek kontúrjait. Az érzékelés alapja, hogy a binarizált képen a pupilla, illetve a reflexiók fehér pixelekből állnak, az első sötét pixel megjelenése tehát az objektumok szélét jelenti. A kontúr pontjainak felhasználásával a szoftvert ellipszist illeszt a pupillára, melynek paraméterei az *Extractor* által létrehozott kimeneti fájl részét képezik. A pupilla méretének és elmozdulásának követéséhez csak a pupilla bináris képére van szükség, a cornea reflexióinak a későbbiekben, a tekintet irányának meghatározásában lesz jelentőségük. A következő képkocka beérkezésekor a motor az előző kép eredményeiből kiindulva megismétli a számításokat. Ha egy képkockán nem található a pupillának megfeleltethető ellipszis, a motor a képkockát pislogásként jelöli meg.

Habár a szoftver működése alapvetően megfelelt céljaimnak, több esetben is szükség volt lényegi módosításokra. Ehhez részleteiben is meg kellett ismernem a programkódot és megérteni annak pontos működését.

A program objektumorientált szemlélettel készült, ezért működésének részletesebb, rendszerszerű bemutatásához UML osztálydiagramot készítettem. Az UML (Unified Modelling Language) informatikai rendszerek felépítésének vizuális megjelenítésére szolgál. Az osztálydiagram tartalmazza a program osztályainak nevét, azok attribútumait (mezőit), operációit (metódusait) és az objektumok közötti kapcsolatokat [20]. A kapcsolatok erősségétől függően többféle típust különítünk el [20], amelyek közül a diagramban használtakat a 4.1. táblázat foglalja össze. A program összetettsége miatt az *EyeLoop* objektumai számos attribútummal és operációval rendelkeznek, a teljes osztálydiagram bonyolult és nagyméretű, ezt az 1. függelék tartalmazza. A 4.4. ábrán látható egyszerűsített diagram az osztályok neveit és a közöttük fellépő kapcsolatokat ábrázolja.

kapcsolat neve	jelentés	jelölés
öröklődés	a leszármazott objektum öröklí az ősosztály összes jellemzőjét és annak hozzáférési szintje szerint módosíthatja is azokat	
asszociáció	az objektumok kapcsolatban vannak egymással	
aggregáció	egy objektumot több másik objektum is tartalmazhat, és külön is elérhetőek	
kompozíció	a tartalmazott objektum csak egy tartalmazó objektumon belül létezik és érhető el	

4.1. táblázat: Az UML objektumdiagramon ábrázolt kapcsolatok jellemzői



4.4. ábra: Az EyeLoop program egyszerűsített UML objektumdiagramja

A rendszer fő részei az osztálydiagramon különböző színnel vannak jelölve: sárgával az *Importer* objektumai, narancssárgával az *Engine* és lilával az *Extractor* objektumok. A *File_Manager* felelős a bemeneti feldolgozandó videó és a kimeneti fájlok kezeléséért, az *Arguments* pedig a bemeneti paramétereket kezeli és tárolja, amelyek

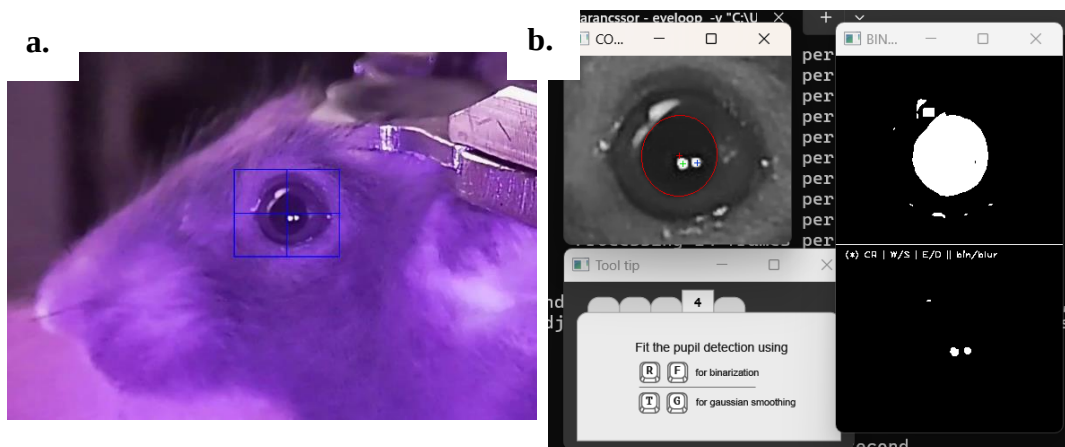
alapvetően meghatározzák a különböző egységek működéseit. Szerteágazó funkcióik révén ezzel a két objektummal a programnak szinte minden része kommunikál. A *GUI* az *Engine*-nel való asszociációs kapcsolata által irányítja annak működését. Az *Extractor*-ok csak az *Eyeloop* objektumon belül léteznek, így közöttük kompozíció van, akárcsak az *Engine* részei között. Az *IMPORTER*-nek két különböző leszármazottja van, ezek közül egy adott futás során csak az egyiket használja, a program indításakor meghatározottaknak megfelelően. A *cv Importer* működik illesztőprogramot nem igénylő kamera használata esetén, vagy előre felvett videó feldolgozásakor, a *thorcam Importer*-t pedig kifejezetten az általam választott Thorlabs kamera kezelésére készítettem.

4.3 Saját kiegészítések

A program áttekintése és tesztelése során több olyan funkciót is találtam, amelyek vagy nem működtek megfelelően, vagy kimaradtak a rendszerből, viszont alkalmazásuk jelentősen megkönnyíti a szoftver alkalmazását és a detektálás pontosságát.

A pupilla és a szaruhártya reflexióinak érzékeléséhez szükséges egy kezdeti manuális kijelölés egérekattintással, ami az *Engine* általi kijelölés alapja. Ez a kódban implementálva van, azonban a kattintás érzékelése nem. Ezért a *GUI* objektumon belül létrehoztam egy *MouseCallback* függvényt (az OpenCV könyvtár egy beépített függvényét alkalmazva), amely folyamatosan követi az egér koordinátáit, és kattintáskor értékül adja azokat a *GUI cursor* tulajdonságának.

A tesztelt kamerák látótere a szemhez közel, néhány cm távolságban is nagy, a képkockák a szemén kívül nagy mennyiségű, a pupillakövetés szempontjából felesleges hátteret is tartalmaznak. Ugyanakkor kísérleti tapasztalataim alapján a bináris kijelölés pontosságát javítja, ha az algoritmusnak kisebb képen kell megtalálnia a pupillát és a cornea reflexiókat. Ezen probléma megoldására kiegészítettem a programot egy olyan lépéssel, amellyel minden futtatás kezdetén ki lehet választani a kamera képének azt a területét, amelyen a szem található (4.5. ábra, a.), és a továbbiakban elegendő ezt a területet, a ROI-t (Region of interest) vizsgálni. Ennek megvalósításához az *Importer* objektumban az openCV könyvtár *selectROI* függvényét használtam. Az első képkocka betöltésekor a program megkéri a felhasználót, hogy az egér lenyomva tartásával és húzásával jelölje ki a vizsgálandó területet. A kiválasztott terület koordinátáinak felhasználásával minden további képkocka körülvágott formában kerül feldolgozásra (4.5. ábra, b.).



4.5. ábra: a. Szem területének kijelölése a videón, b. a további feldolgozás csak a kiválasztott képterületen történik

A pupillakövetési kísérletekben alapesetben az egér pupillája sötétebb a környezeténél, ezért a pupilla illesztéséhez a kép binarizálása során annak színeit invertálni kell, hogy az eredeti kép sötét pixeljei a bináris képen fehérnek jelenjenek meg. Kutatócsoportunkban emellett végeznek olyan biológiai méréseket is, amikor az egér fejébe világítanak be lézerrel, ekkor a szem világít, a pupilla tehát a környezeténél világosabb lesz. Ebben az esetben változtatni kell a pupilla küszöbölésén, mivel ekkor a nagyobb intenzitású, világos pixeleknél kell fehérnek maradniuk a bináris képen. Ennek megoldására beiktattam egy plusz döntési lépést a program kezdetén, ahol a felhasználó kiválaszthatja, hogy az adott felvételen milyen intenzitású pixelek jellemzik a pupillát. Sötét pupilla esetén a *D* (dark) billentyűt kell lenyomni, ekkor a kép invertálása megtörténik, világos pupilla esetén a *B* (bright) billentyű lenyomásával az invertálás kimarad.

A szoftver működésének fontos paraméterei az *Extractor*-ok révén kerülnek rögzítésre, amelyek a pupillakövetés leállítását követően több kimeneti fájl is generálnak, ezeket a 4.2. táblázat foglalja össze. A pupilla kalibrációs, valamint a binarizálási és simítási paramétereket tartalmazó konfigurációs fájlok későbbi futtatásokkor betölthetőek és felhasználhatóak, hogy ezek a tulajdonságok a mérések között ne változzanak. A fájlok típusa *.npy*, ez egy standard bináris fájlformátum, az adatszerkezetek formáját, az adatok típusát és értékeit tárolja. A feldolgozott videó *mp4* formátumban kerül kimentésre. A szoftver futásának fontos paramétereit és a *logger* üzeneteit egy szöveges dokumentum tartalmazza. Ez információt nyújt a használt *Importer*-ekről és *Extractor*-okról, az elemzett videófájl vagy a használt kamera nevééről, a kiválasztott ROI koordinátáiról és a követés kezdetének időpontjáról. A biológiai

kísérletekben fontos, hogy a mért pupillajelek időben összeilleszthetők legyenek az állatot ért ingerekkel és az egyéb mért kimeneti jelekkel, ezért lehetővé tettem markerek rögzítését, a kimentett dokumentumban a markerek sorszáma és a rögzítés időpillanata látható.

Annak meghatározásához, hogy a pupilla követéséről mely adatok és milyen formátumban kerüljenek rögzítésre, a szoftver jövőbeli felhasználóit, kutatócsoportunk biológusait kérdeztem. Javaslaik alapján a program exportálja az indítás óta eltelt időt, a cornea reflexiók koordinátáit pixelben mérve, és a pupillára illesztett ellipszis paramétereit: a fél kis- és nagytengely hosszát (pixel), az ellipszis területét terület (pixel²), és az ellipszis dőlésszögét (°), ami a fél nagytengely vízszintessel bezárt szögét jelenti. A pixeleken mért méretek a későbbiekben szükség esetén átalakíthatóak, ehhez azonban figyelembe kell venni, hogy a szemmozgások nem valódi translációs elmozdulások, hanem szögelfordulások [11], az átszámításhoz kalibráció szükséges. Az adatok a *DAQ_to_csv_extractor* alkalmazásával .csv (vesszővel tagolt fájl) kiterjesztésben kerülnek kimentésre, mert ez a fájlformátum sokrétű további feldolgozást tesz lehetővé.

fájlnév	kiterjesztés	leírás
kimeneti fájl	vesszővel elválasztott szöveges (.csv)	mérési adatok cornea reflexiókról és pupilláról
logger fájl	szöveges (.log)	mérés paramétere
kimeneti videó	.mp4	a feldolgozott videó
pislogási kalibrációs fájl	.npy	pislogási paraméterek
paraméter fájl	.npy	binarizációs és simítási paraméterek

4.2. táblázat: Kimeneti fájl típusok összefoglalása

4.4 Kamera kiválasztása és szoftveres kezelése

Ahhoz, hogy a pupillakövetés az optikai vetítőrendszerrel egybeépíthető legyen, olyan kisméretű kamerára van szükség, amelynek látótávolságát néhány cm-ig lehet csökkenteni, valamint érzékelnie kell az infravörös hullámhossztartományt. Többféle típusú egyszerű, kisméretű, USB kamerát teszteltem, azonban fix fókusz távolságuk miatt a szükséges kis távolságban nem alkottak éles képet az egér szeméről. A megoldást a Thorlabs CS165MU/M monokróm 1,6 megapixeles kamera és a Schneider-Kreuznach Cinegon 1.4/8-0512 objektív alkalmazása jelentette.

A kiválasztott kamera kezeléséhez illesztőprogram szükséges. A *Thorcam* szoftver a gyártó oldaláról ingyenesen letölthető, a kamera első tesztjei során ezt használtam, *EyeLoop*-ba integrálásához pedig létrehoztam egy saját *Importer* objektumot a kamera kezelésére. A kamera programozásához a gyártó által biztosított segédletet [21] használtam, ez tartalmazza a kamera működtetésének szükséges lépéseit és az alkalmazandó függvények leírásait.

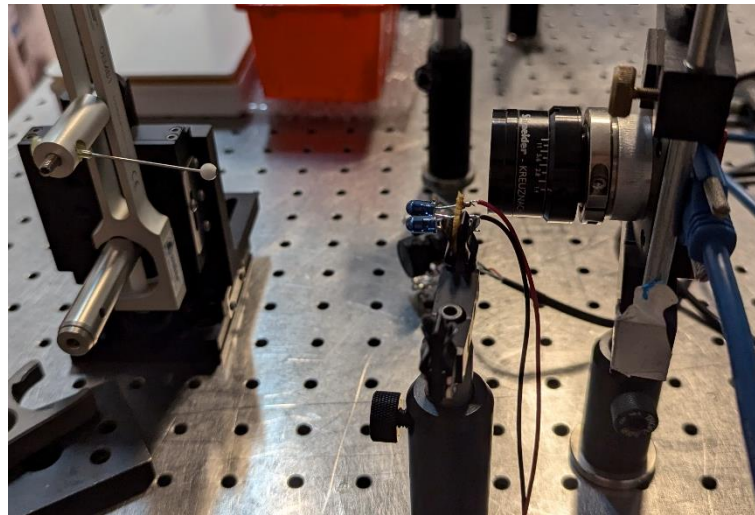
A kísérleti körülmények fényviszonyai sokfélék lehetnek, előzetes ismeretek nélkül a kamera expozíciós idejét nem lehet előre meghatározni. Ennek megoldására iteratív autoexpozíciós algoritmust [22] implementáltam, ami a szoftver indításakor kamera által rögzített kép átlagos pixelintenzitása alapján iteratívan kiszámolja a megfelelő expozíciós időt, ezáltal lehetővé teszi az alkalmazkodást különböző megvilágítottságú kísérleti körülményekhez. A kiszámított értéket a felhasználó igényei szerint módosíthatja. Az algoritmus a T expozíciós idő függvényében modellezi a képernyő $f(T)$ fényességét, úgy, hogy minden iterációban kiszámolja az adott expozíciós idővel létrejövő kép pixeleinek átlagos intenzitását. Az algoritmus addig iterál, amíg meg nem találja azt a T expozíciós időt, amire az $f(T)$ átlagos fényesség megfelel az F_t célértéknek:

$$T_{n+1} = \frac{[Y_{fixed} - f(T_n)] \cdot T_n \cdot T_{max}}{[F_t - f(T_n)] \cdot T_n - \alpha}, n = 0, 1, 2 \dots$$

ahol $\alpha = (Y_{fixed} - F_t) \cdot T_{max}$, F_t az elérni kívánt átlagos pixelintenzitás, jellemzően a dinamikus tartomány közéértéke, Y_{fixed} értéke pedig a maximális intenzitás plusz egy. T_{max} nem lehet nagyobb, mint egy kép megjelenítésének ideje, ami a másodpercek alatt megjelenített képkockák számának reciproka. A Thorlabs kamera 10 bites képeket készít, ezért $F_t = 512$ és $Y_{fixed} = 1024$. A vizsgált terület (ROI) kijelölését megelőzően a szem jellemzően csak kis részét foglalja el a kameraképnek, ha a háttér nagy sötét területként jelenik meg, akkor az átlagos pixelintenzitás F_t értékének csökkentésére lehet szükség. A kamera képkockasebessége a teljes képet használva 34,8 fps, ezért $T_{max} = \frac{1}{34,8 \text{ frame/s}} = 0,029 \text{ s}$. A ROI meghatározása jelenleg utólag, a kamera objektumtól függetlenül történik, amennyiben nagyobb képkockasebességre lenne szükség, célszerűbb lenne ehelyett a ROI-t a kamera kezelése során beállítani.

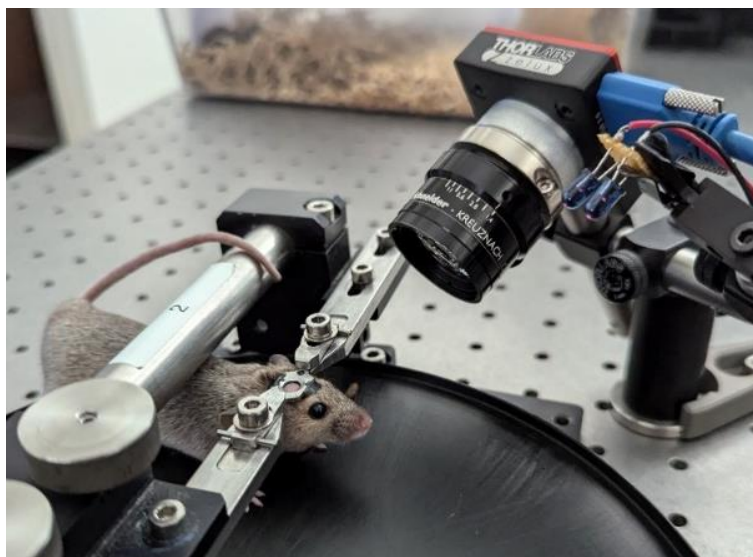
4.5 Kísérleti rendszer felépítése

A kísérletekben a szem megvilágításához 2 darab infravörös LED-et használtam, amelyek egymás mellett, közvetlenül a kamera mellett helyezkedtek el. Mivel élő egerek a teljes tesztelési folyamat során nem álltak rendelkezésemre, 3 mm átmérőjű, fehér hungarocell gömbből szemmodellt készítettem, amire fekete alkoholos filccel egy pupillát rajzoltam. A szemmodellt egy tűre szúrva rögzítettem, majd a kamera elé, tőle körülbelül 5 cm távolságra helyeztem (4.6. ábra). A kamera fókusztávolságát beállítottam úgy, hogy a szemmodell legyen fókuszban. Ezt az első, egyszerű kísérleti elrendezést a szoftver fejlesztése közben használtam, különböző funkciók, például a ROI beállításának tesztelésére.



4.6. ábra: Első kísérleti elrendezés szemmodellel

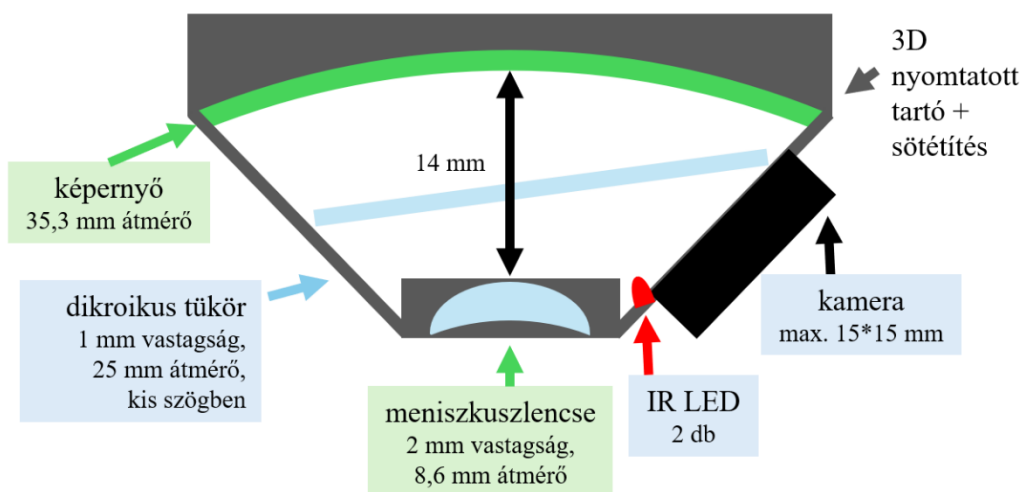
Miután megbizonyosodtam a kísérleti elrendezés és a szoftver megfelelőségéről, a szemmodell pozíciójába élő állatot helyeztem (4.7. ábra), amelynek feje rögzített volt, és egy gramofonon állt, ami mozgási lehetőséget biztosított számára (a mozgásnak a pupillakövetés szempontjából nem volt jelentősége). Kétféle típusú, világos és sötét szőrű egéren is teszteltem a szoftver működését. Ezek szeme különböző mértékben pigmentált, a sötét színű egerek esetén a szem és a pupilla között kisebb a kontraszt, ami nehezítheti a pupilla körvonalainak megtalálását.



4.7. ábra: Pupillakövető kísérlet elrendezése vilgos szőrű egérrel

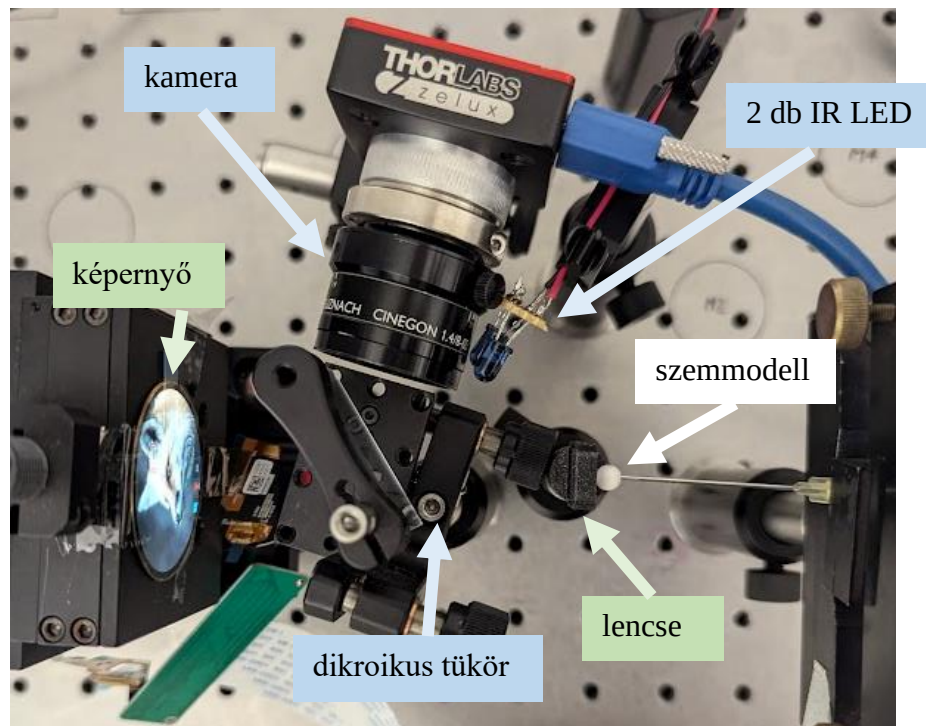
4.6 Kombinálás vetítőrendszerrel

A pupillakövetés és a vetítés egyidejű megvalósításához a pupilla mérésének a vetítőlencsén keresztül is a megfelelő pontossággal kell működnie. A két rendszer integrálása dikroikus tükör alkalmazásával lehetséges (4.8. ábra). Ezt olyan szögben kell elhelyezni a vetítőrendszerben a képernyő és a meniszkuszlencse között, úgy, hogy az optimális képernyő-lencse távolság megmaradjon. A kamera és a 2 db infravörös LED a vetítőrendszer oldalán, a 3D nyomtatással készült tartóba rögzítve helyezkednek el. Méréseim alapján a fényforrások ideális pozíciója a kamera mellett, a meniszkuszlencse felőli oldalon van, ha ez esetleg mechanikai problémák miatt nem valósítható meg, akkor megfelelő szögben a kamera alá vagy fölé is kerülhetnek.



4.8. ábra: Integrált vetítő- és pupillakövető rendszer tervezett elrendezése

A megtervezett elrendezés első megvalósítása 4.9. ábraán látható, ami a koncepció működésének tesztelésére szolgál. A kísérleti összeállításban a vetítőrendszer elemei egymás előtt helyezkednek el: a szemmodelltől 2 mm távolságra van a 3D nyomtatott lencsetartó a pozitív meniszkuszlencsével, előtte egy tartóban ferdén van a dikroikus tükör, ezt pedig a kerek képernyő követi, ami ebben az elrendezésben még nincs meggörbítve. A dikroikus tükör mellett, oldalt található a kamera, oldalán a fényforrásokkal.



4.9. ábra: Pupillakövető- (elemei kék háttérrel) és vetítőrendszer (zölddel) elrendezés felülnézetből

A 4.9. ábraán látható elrendezés több szempontból sem felel meg a rendszer méretével szemben támasztott követelményeknek. Az egyedi 3D nyomtatott tartó hiánya, valamint a kamera és a dikroikus tükör nagy mérete miatt a lencse és a képernyő távolsága az optimális 14 mm-nél jelentősen nagyobb, így a vetítés minősége nem megfelelő. A probléma megoldásához csökkenteni kell a pupillakövető rendszer elemeinek méretét, valamint aprólékosan megtervezni a tartóelemeket, hogy a pupillakövetés és a vetítés is megfelelően tudjon működni a kisméretű rendszerben. A szükséges kisméretű elemek beszerzése és a 3D nyomtatott tartómechanika tervezése folyamatban van.

5 Eredmények

5.1 A pupillakövető szoftver működési folyamata

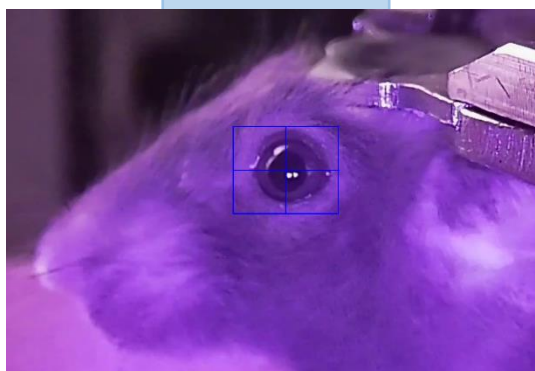
Az elkészült szoftver működési folyamata az 5.1. ábraán látható, amely az eredeti *EyeLoop* funkcióit és saját kiegészítéseimet is tartalmazza. A folyamat három jól elkülöníthető lépésre tagolódik: indításkor konfigurálni kell a ROI-t, a pupilla fényességét és a kamera expozíciós idejét; a kijelölés során a felhasználónak be kell állítania a kép binarizálásának optimális paramétereit; a követés során pedig az optimalizált bináris kép alapján a szoftver ellipszist illeszt a pupillára. A követés leállítását követően a kimeneti adatok mentésre kerülnek, majd a program leáll. Az 5.1. ábra jobb oldalán az is látható, hogy az egyes lépésekben a felhasználónak hogyan, milyen billentyűparancsokkal, illetve helyenként kattintással kell irányítania a szoftvert.

Követés közben markerek rögzíthetőek, ezek rögzítésének időpontja a kimeneti logger fájlban jelenik meg, felhasználhatóak a későbbi elemzés során. A markerek használata nagyban megkönnyíti a felvételek elemzését, mivel így pontosan tudjuk, hogy meddig tartott a paraméterek beállításának fázisa - ez a szakasz természetesen még nem szolgál releváns eredményekkel - és mikortól kezdődött a követés. A markerek felhasználhatóak arra is, hogy a detektálás mérés közben észlelt pillanatnyi hibáit jelöljük, hogy azok a későbbi elemzés során figyelmen kívül hagyhatóak legyenek.

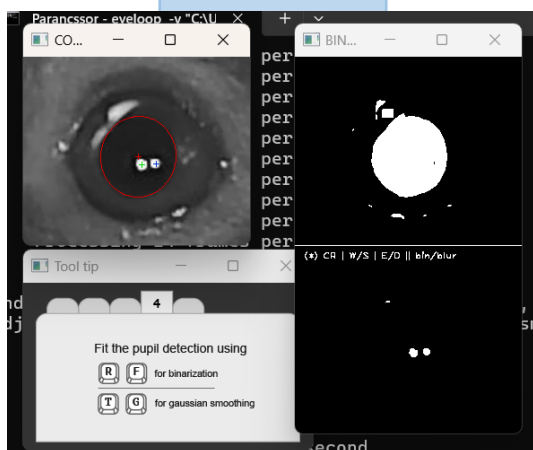
A továbbiakban ismertetett eredmények alapján a program megfelelően működik, a parancssorban olvasható információk alapján az egymást követő lépések jól végrehajthatóak. A pontos érzékelés érdekében fontos a lehető legkisebb ROI kiválasztása, ami csak a szemet és közvetlen környezetét tartalmazza, valamint a binarizálás és a simítás paramétereinek ideális kiválasztása, utóbbiak nagyon nagy mértékben hozzájárulnak a helyes követéshez, és ahhoz, hogy a szoftver a követési fázis alatt ne veszítse el a pupillát.

A szoftver alkalmazhatóságának jövőbeli fejlesztése során előnyös lenne visszalépést lehetővé tenni az egyes lépések között, valamint egy olyan grafikus felhasználói felületet létrehozni, amelyik nem a parancssorban, hanem egy jobban látható ablakban írja ki az utasításokat a felhasználó számára.

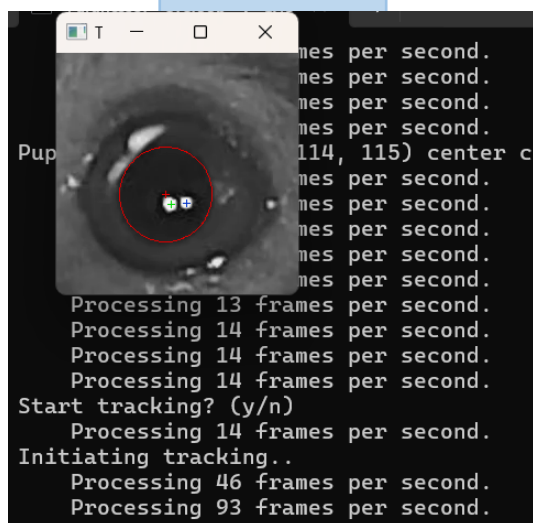
INDÍTÁS



KIJELÖLÉS



KÖVETÉS



lépés	irányítás, billentyűk
1. expozíciós idő automatikus kiszámítása	-
2. ROI beállítása	egér, ENTER
3. pupilla fényességének megadása	B: világos D: sötét
4. expozíciós idő automatikus korrigálása	-
5. expozíciós idő módosítása	L: +1 ms K: -1 ms → megerősítés: X

6. cornea reflexiók kijelölése	egér-rákattintás
7. cornea reflexiók bináris képének beállítása	W: küszöbérték ↑ S: küszöbérték ↓ E: simítás ↑ D: simítás ↓
8. pupilla középpontjának kijelölése	egér-rákattintás
9. bináris pupillakép paramétereinek optimális beállítása	R: küszöbérték ↑ F: küszöbérték ↓ T: simítás ↑ G: simítás ↓

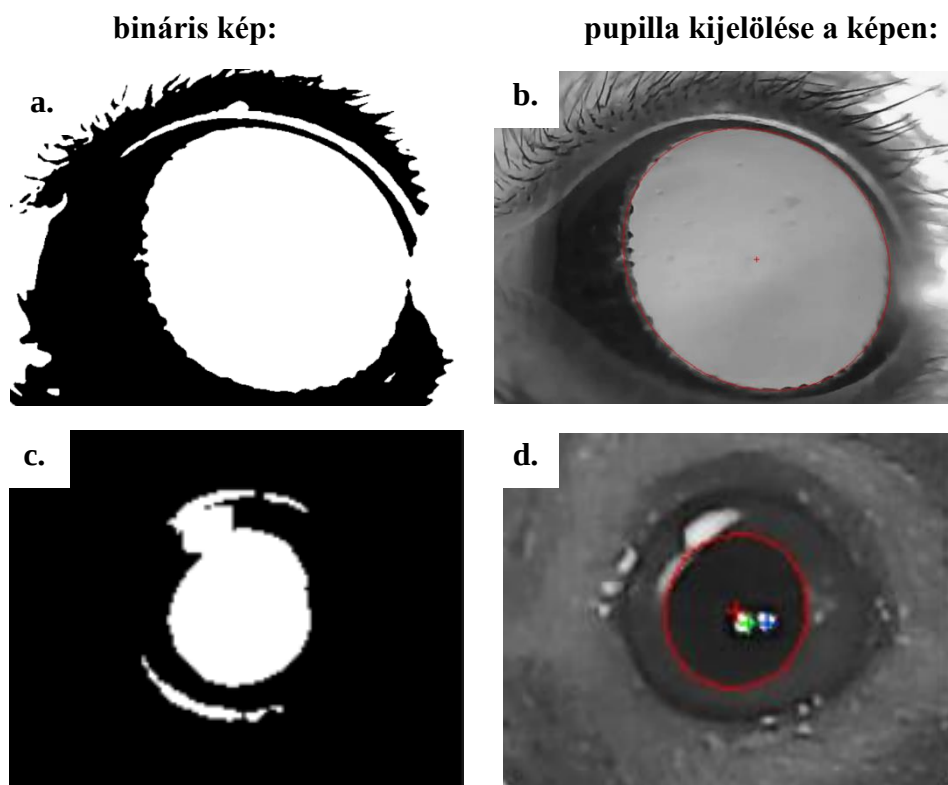
10. követés elindítása	Z → megerősítés: Y → mégsem: N
11. követés, markerek rögzítése	M
12. követés leállítása	Q

5.1. ábra: A kész szoftver működésének fő lépései, részletes folyamata, és ezek felhasználói kontrollja (az opcionális lépések kék háttérrel jelölve)

5.2 Tesztelés előre felvett videókon

A szoftver működését a biológiai méréseket megelőzően előre felvett videókon teszteltem. Az egérről normál körülmények között készített videót egy beépített infravörös megvilágítást tartalmazó kamerával készítettem. A kamera nem volt fókuszálható, ezért nagyobb távolságba, körülbelül 20 cm-re kellett helyezni az egér szemétől. A világító pupillát tartalmazó felvételek nem saját készítésűek, mert ilyen kísérletben még nem volt lehetőségem részt venni.

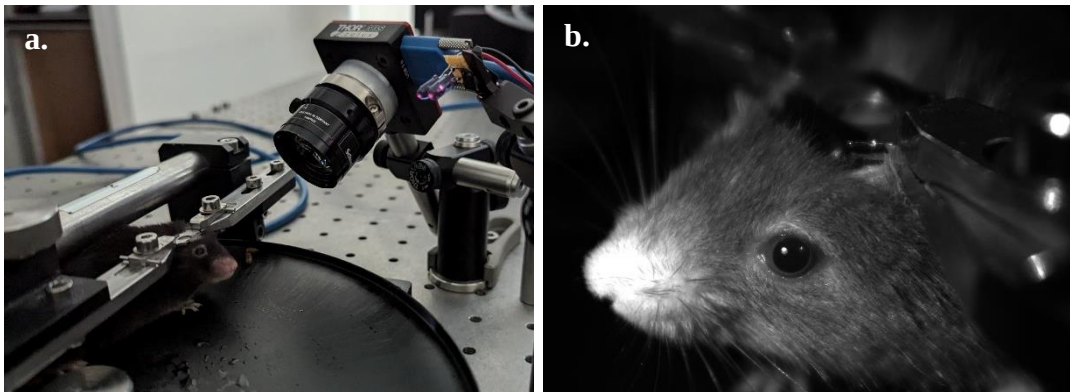
A szoftverbe általam beépített döntési lépéssel sötét és világos pupilla kezelése is sikeres volt, a bináris képen mindkét esetben a pupilla látszik világos pixelekkkel, és a paraméterek megfelelő beállításával a pupilla pontosan detektálható és követhető (5.2. ábra). A világító pupillás felvételen több nagyobb kiterjedésű, hirtelen szemmozgás fordult elő, ezek esetében a szoftver volt, hogy néhány képkockán elveszítette a pupillát. Továbbá világos pupilla esetén a szaruhártyáról visszaverődő fénypontok nem látszanak, viszont ez nem okoz problémát, mivel a pupilla detektálása ezektől függetlenül történik.



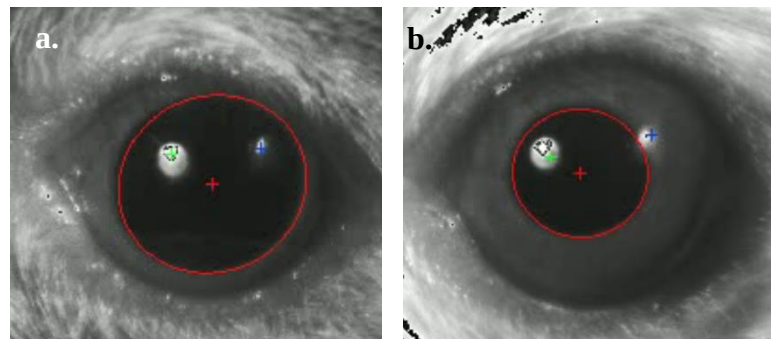
5.2. ábra: Világos (a, b,) és sötét (c, d,) pupilla detektálása bináris képek alapján

5.3 Kísérleti eredmények élő egereken

Miután az előre felvett videókon a pupillakövetés megfelelő pontossággal működött, a szoftvert és a pupillakövető rendszert élő egereken is teszteltem, egyelőre vetítőrendszer alkalmazása nélkül (5.3. ábra, a.). A kamera felvételei élesek, a pupilla detektálásához megfelelő minőségűek (5.3. ábra, b.) voltak, a két darab infravörös LED elegendő megvilágítást biztosított. A szivárványhártya és pupilla között mindkét típusú egér esetében elegendő kontraszt volt a pupilla határainak kijelöléséhez (5.4. ábra).



5.3. ábra: a. Kísérleti elrendezés pupillakövető rendszerrel és sötét szőrű egérrel, b. a kamera által látott kép



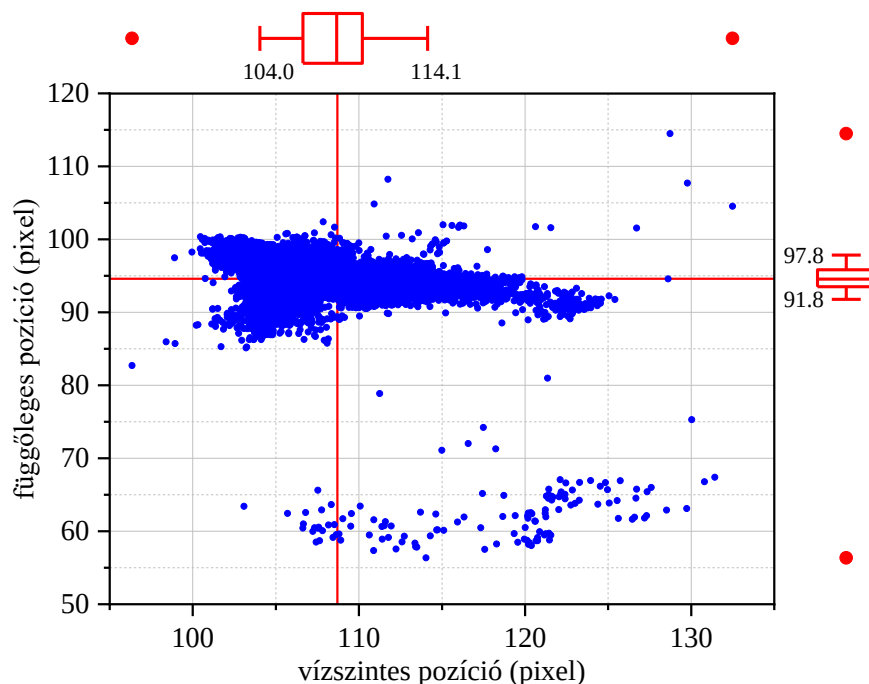
5.4. ábra: Pupilla kijelölése fekete (a.) és szürke (b.) egéren

A pupillakövető mérések 12,5 percig tartottak, ennyi idő alatt megfigyelhetők fluktuációk a pupilla méretében, valamint a pupilla elmozdulásai is. A kísérletek elején jelentős időt vett igénybe a bináris kép paramétereinek optimális beállítása, ami elengedhetetlen a későbbi pontos követéshez. Ez alatt a pupilla detektálása még pontatlan, ezért az eredmények kiértékelésekor a mérések első 100 másodpercét nem vettem figyelembe. Az adatok elemzése mindkét egér esetén ugyanazon módszerekkel történt, a továbbiakban együtt kerülnek bemutatásra (F: fekete egér, SZ: szürke egér). A kijelölt ROI mindkét esetben körülbelül $200 \text{ pixel} \cdot 180 \text{ pixel}$ méretű volt (F: $200 \cdot 182$, SZ:

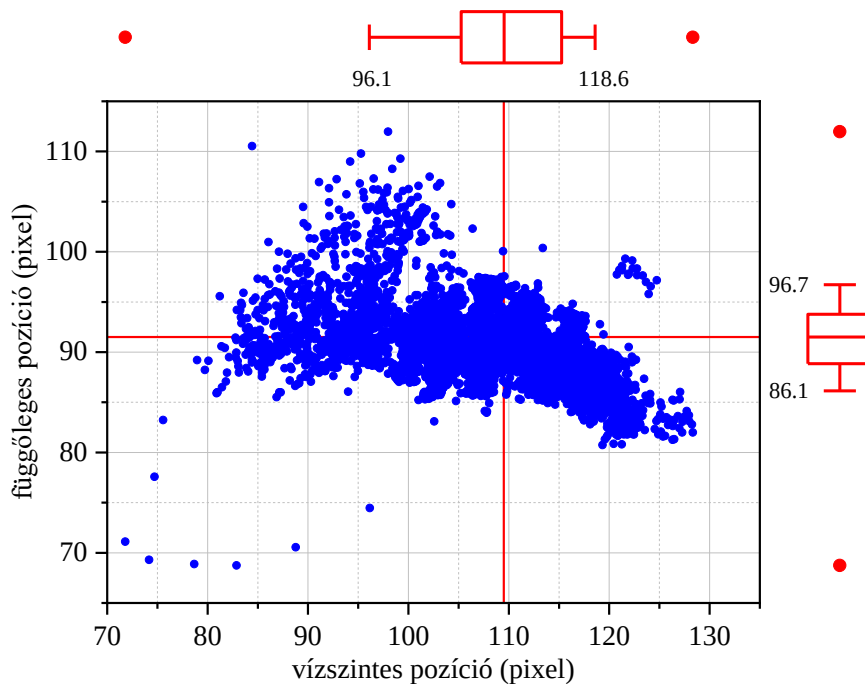
208·186), és mivel a két egér mérésében a kamera és az állat térbeli pozíciója nem változott, a pixelben mért eredmények összevethetőek egymással.

A statisztikai elemzéshez a mérési adatokat dobozdiagramon ábrázoltam, ami egy szemléletes eszköz a minta eloszlásának jellemzésére. A grafikon ábrázolása kvantilisok alapján történik, amik tetszőleges arányban osztják a mintát, és minél nagyobb a kvantilisok távolsága, annál nagyobb a minta szórása. A téglalap határai az alsó (a minta 25%-a nála kisebb értékű, Q_1) és a felső (a minta 75%-a nála kisebb értékű, Q_3) kvantilisok, a kinyúló vonalak pedig az 5%-os (P_5) és a 95%-os (P_{95}) kvantilisok között húzódnak. A vonalakon túli minták kilógó adatokként külön pontokkal jelölhetők, a minimum és a maximum értékeket is megjelenítettem a dobozok tartományainak szélein. A számszerű statisztikai adatok, amik alapján a grafikonok készültek, 2. függelékben megtalálhatóak.

Az átlagos pupillapozíció a szürke egérenél ($108,7 \pm 3,3$ pixel; $94,6 \pm 3,1$ pixel), a fekete egérenél ($109,5 \pm 6,9$ pixel; $91,5 \pm 3,4$ pixel) voltak. A mért pupillapozíciókat ábrázolva (5.5. ábra, 5.6. ábra) megfigyelhető, hogy a pozíciók változása vízszintes irányban mindkét esetben nagyobb terjedelmű (P_5 - P_{95} terjedelem, SZ: 22,5 pixel, F: 10,1 pixel), mint függőleges irányban (P_5 - P_{95} terjedelem, SZ: 10,6 pixel, F: 6,0 pixel), ez összhangban van azzal az elvárással, hogy a rögzített fejű egerek fő szemmozgásai



5.5. ábra: Fekete egér pupillapozíciói, az átlagos pozíció piros vonalakkal jelölve

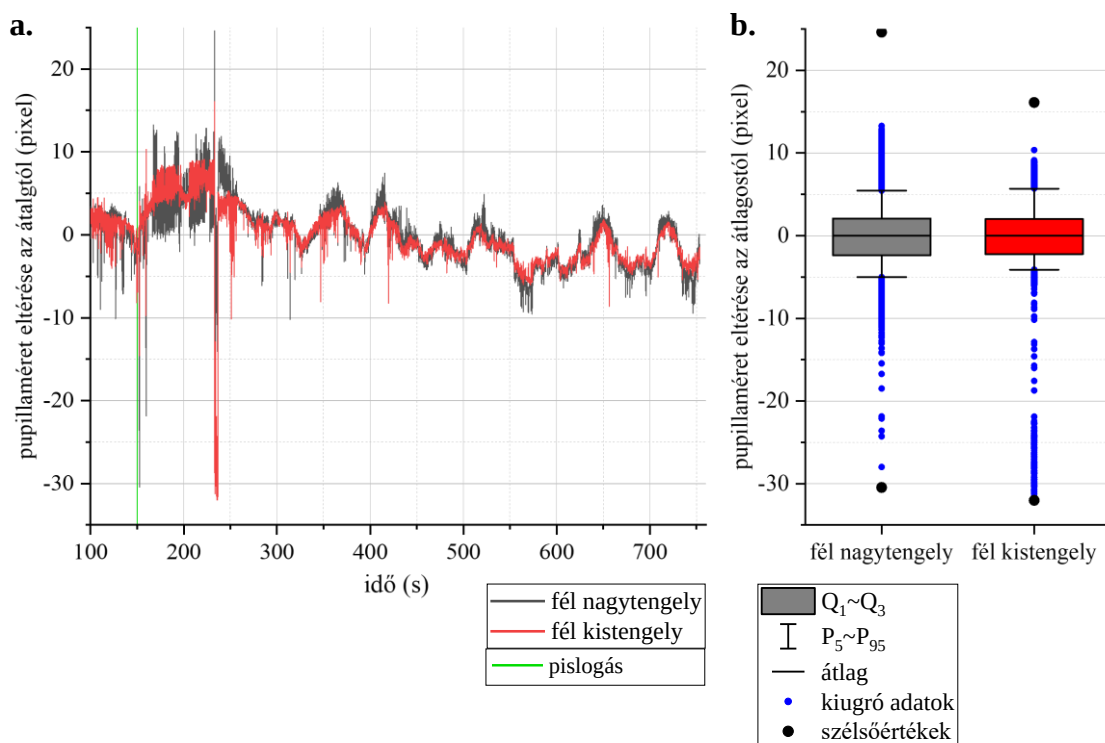


5.6. ábra: Szürke egér pupillapozíciói, az átlagos pozíció piros vonalakkal jelölve

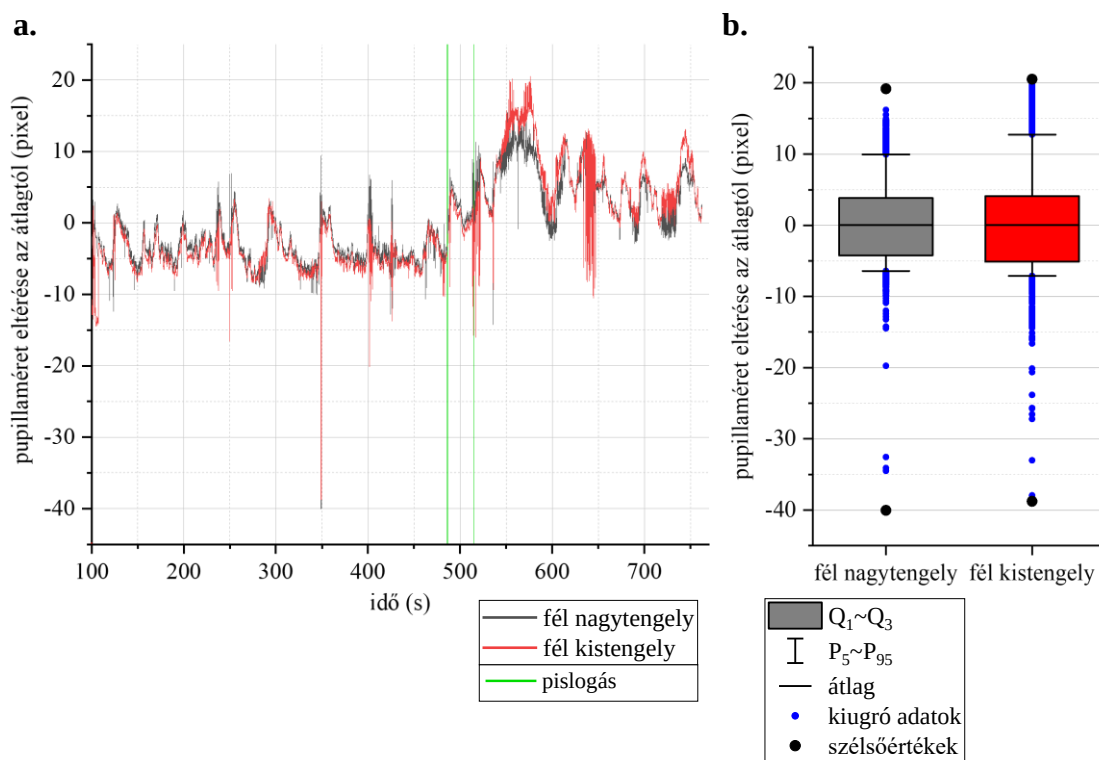
horizontális irányúak. A szürke egér szemmozgásai a mérés közben rögzített videón láthatóan nagyobb terjedelműek a fekete egérénél, ezt a mérési adatok is alátámasztják.

A pupillára illesztett ellipszis fél nagy tengelyének átlagos méretei $49,9 \pm 3,4$ pixel (F) és $42,4 \pm 5,2$ pixel (SZ), fél kistengelyének méretei $46,5 \pm 3,6$ pixel (F) és $40,6 \pm 6,4$ pixel (SZ) pixel. Az ellipszis nagy tengelye a vízszintessel átlagosan kis, negatív szöget zár be, a szögek szórása nagy (F: $-11,4 \pm 16,2^\circ$; SZ: $-7,7 \pm 23,0^\circ$). Az egér pupillájára illesztett ellipsziszről tehát általánosságban elmondható, hogy közel vízszintes orientációjú és szinte kör alakú, a fél nagy tengely csupán néhány pixellel nagyobb a fél kistengelynél.

A pupillaméret változását az idő függvényében ábrázoltam (5.8. ábra, 5.7. ábra), a statisztikai elemzéshez itt is dobozfüggvényt készítettem. Mindkét felvételen megjelennek kiugró értékek, amik jelentősen megnövelik az adatok teljes terjedelmét (SZ: fél nagy tengely: 59,2 pixel, fél kistengely: 59,3 pixel; F: fél nagy tengely: 55,1 pixel, fél kistengely: 48,1 pixel). A P_5 - P_{95} tartományok viszont ennek töredékei (SZ: fél nagy tengely: 16,4 pixel, fél kistengely: 19,8 pixel; F: fél nagy tengely: 10,4 pixel, fél kistengely: 9,8 pixel), tehát a kiugró értékek viszonylag kis számban vannak jelen. A szürke egér esetében a P_5 - P_{95} tartomány szélesebb, ez valószínűleg a fokozott szemmozgások, és a hozzájuk társuló fokozott pupillaméret-változások eredménye. A kiugró értékek valószínűleg hibás detektálás eredményei, mivel a pupilla méretének ilyen



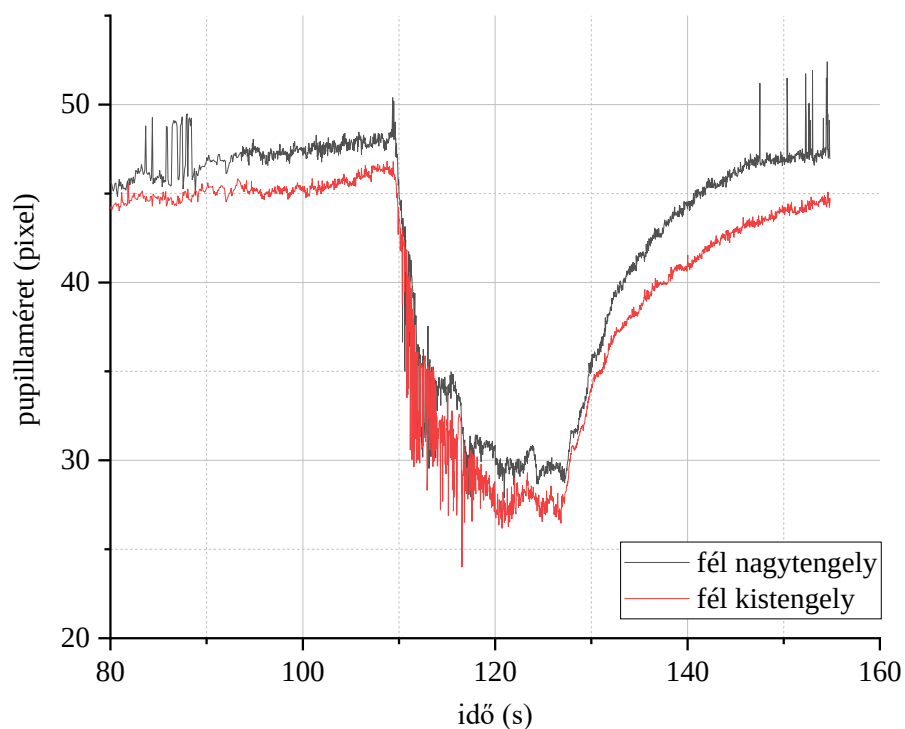
5.7. ábra: a. A pupillára illesztett ellipszis méretének időbeli változása az átlagos mérethez képest, a pislogások függőleges zöld vonallal jelölve, fekete egér. b. A mért adatok statisztikai paramétereinek ábrázolása dobozdiagramon.



5.8. ábra: a. A pupillára illesztett ellipszis méretének időbeli változása az átlagos mérethez képest, a pislogások függőleges zöld vonallal jelölve, szürke egér. b. A mért adatok statisztikai paramétereinek ábrázolása dobozdiagramon.

nagy sebességű változása, valamint az adott felvételeken a néhány pixel nagyságú pupillaméret nem reálisak. Az ábrákon megfigyelhető, hogy a pislogások megjelenése körül (F: 150 s, SZ: 515 s) gyakoriak a kiugró értékek, ezek oka lehet, hogy a szem becsukása és kinyitása során a pupilla érzékelése kevésbé pontos.

A kísérletet végrehajtottam úgy is, hogy a pupilla követése közben 20 cm távolságból egy lámpával megvilágítottam az egér szemét. A pupillára illesztett ellipszis fél kis- és nagy tengelyének változása az idő függvényében a mérés 80. másodpercétől az 5.9. ábraán látható. A fény hatására a pupilla reflexes összeszűkülését várjuk, ez jól megfigyelhető a rögzített adatokban: a kezdeti átlagos pupillaméret $47,3 \pm 0,8$ pixel (fél nagy tengely) volt, ezután a pupilla 7,6 s alatt összeszűkülött $29,9 \pm 0,7$ pixel (fél nagy tengely) méretűre. A fényinger megszűnését követően a pupilla folyamatosan növekedett, majd a szűkülésnél lassabban, 27,3 s alatt tért vissza eredeti mérettartományába.



5.9. ábra: Pupilla méretének változása a fényreflex hatására

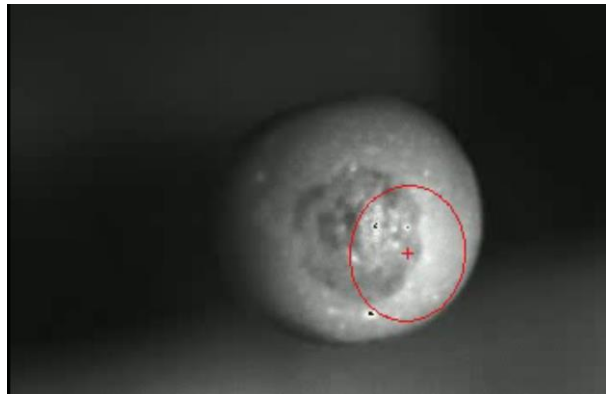
Ebben az alfejezetben bemutattam, hogy a pupillakövetés adataiból hogyan lehet biológiai jelentőségű adatokat kinyerni. A későbbiekben, a mérés többszörös elvégzése után szereshető elegendő tapasztalat ahhoz, hogy a hirtelen szemmozgásokat a mérési hibáktól pontosan el lehessen különíteni, ez alapján a mérési hibák a statisztikai

elemzésben figyelmen kívül hagyhatóak, vagy jelfeldolgozási lépésekkel a hibák hatása csökkenthető, például alul áteresztő szűrés vagy simítás alkalmazásával.

Amit a kamera síkbeli képén a szoftver a pupilla középpontjának elmozdulásaként érzékel, az valójában a szem térbeli elfordulásának eredménye. Ahhoz, hogy az eredményeket szemléletesebb mértékegységekben, az elfordulásokat fokban, a pupilla méretének változásait pedig μm -ben lehessen megadni, ismerni kell a kamera és a szem távolságát, valamint a szem térbeli pozícióját a kamerához képest. Ezek meghatározásához a későbbiekben kalibrációs és térbeli távolságmérési eljárás kifejlesztése szükséges.

5.4 Kombinálás vetítőrendszerrel

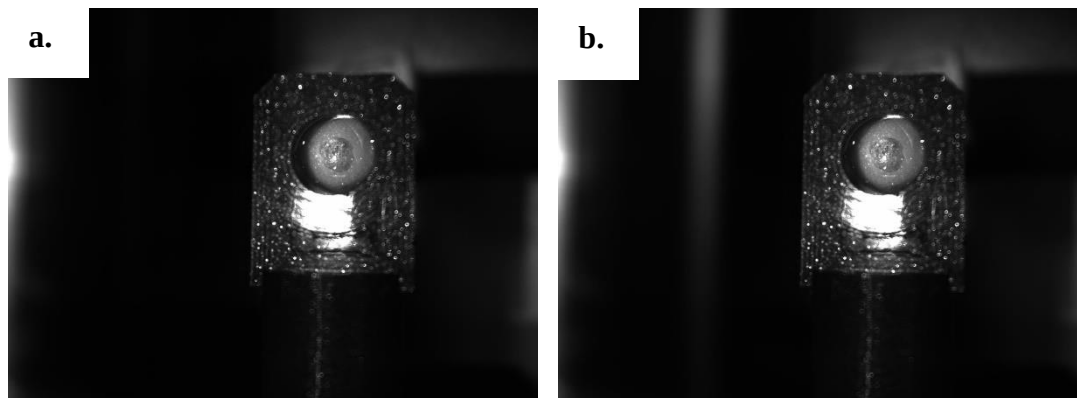
A hungarocell egérszemmodell a mérési elrendezésnek, a szoftver kiegészítő funkcióiak és a kamera beállításainak tesztelésére megfelelő volt, azonban a fekete filccel rajzolt pupilla nem nyelte el elegendő mértékben az infravörös fényt, nem látszott megfelelő kontraszttal a videón, ezért a szoftver nem tudta pontosan követni (5.10. ábra).



5.10. ábra: Rosszul kijelölt pupilla az egérszemmodellen

A pupillakövetés és a vetítőrendszer együttes alkalmazásának legfontosabb kérdése, hogy a két módszer mennyire zavarja meg egymás működését. Mivel az infravörös LED-eket az egér nem látja, a látható fényt pedig a dikroikus tükör szinte teljes mértékben átengedi, a pupillakövetés nem zavarja a vetítést. Eredményeim alapján a képernyő fénye sem zavarja meg a pupilla követését, mert a kamera képén a képernyő fénye egyedül egy keskeny sávban, a dikroikus tükör szélénél, halványan jelenik meg (5.11. ábra). Fontos kísérleti eredmény, hogy a pupilla a szem elé helyezett, a megjelenítő képét a retinára vetítő lencsén keresztül is követhető. Azonban a lencsetartó nagymértékben visszaveri az infravörös fényt, emiatt célszerű lenne esetlegesen a

jövőben más anyagból 3D nyomtatni azt, ugyanakkor a kamera ROI kijelölhető olyan szűkre, hogy a lencsetartó azon kívül essen.



5.11. ábra: Vetítés és pupillakövetés szemmodellen, sötét (a.) és világos (b.) képernyő esetén. A két infravörös LED mindkét esetben megvilágította a szemet.

Eredményeim alapján tehát a két rendszer kompatibilis egymással, az optimális vetítéshez szükséges kompakt elrendezés azonban csak bizonyos komponensek cseréjével érhető el. Ennek megoldását követően sor kerülhet a kombinált rendszer élő egereken történő tesztelésére is.

Ezen adatok felhasználásával geometriai egyenletek segítségével rekonstruálni lehet a szem optikai tengelyét, valamint a tekintet irányát. Elsőként az ismert adatokból meg kell határozni a szaruhártya görbületi középpontjának térbeli pozícióját ($\mathbf{c}$), az egyenletek bonyolultsága miatt ebben a lépésben numerikus számítási módszereket is alkalmazni kell. Ennek ismeretében kiszámíthatóak a szaruhártyán visszaverődő fénypontok koordinátái ($\mathbf{q}_{ij}$ és $\mathbf{r}_j$), a pupilla középpontja ($\mathbf{p}$) és a szem optikai tengelye. Végül a szem térbeli pozíciójának meghatározását követően a tekintet vektora meghatározható.

Az általunk elsődlegesen használni tervezett, szemenként egy kamerát ($j = 1$) és két fényforrást ($i = 1, 2$) tartalmazó rendszer már elegendő információt szolgáltat az egyenletek egyértelmű megoldásához. A tekintetkövetés pupillakövető rendszerbe integrálásához azonban több nehézségre is megoldást kell találni. A numerikus módszerek lelassíthatják a szoftver működését, ezért a tekintet valós idejű meghatározásához fontos a számításokat a lehető leghatékonyabban implementálni. Amennyiben így sem érhető el megfelelő gyorsaság, a tekintet kiszámítására csak utólagosan, előre felvett videókon lesz lehetőség. A szem térbeli pozíciójának meghatározásához kalibráció szükséges, ennek kivitelezése egéren egyelőre kérdéses. Ezen kívül a szükséges paramétereket (szaruhártya görbületi sugara, szaruhártya és csarnokvíz kombinált törésmutatója, szaruhártya és a pupilla görbületi középpontjának távolsága) az egér szemére meg kell határozni, vagy irodalmi adatokat keresni. A kalibrációból nyert adatok továbbá felhasználhatóak lennének a pixeleken mért, síkbeli elmozdulásként meghatározott pupillamozgások átszámítására valódi, rotációs elmozdulásokká.

6.2 A pupillakövető szoftver működésének fejlesztése

A szoftver működése hosszútávon hatékonyabbá tehető lenne a grafikus felület további fejlesztésével, úgy, hogy ne a parancssorból, billentyűlenyomásokkal kelljen irányítani. Ezen kívül fontos lenne visszalépési lehetőséget biztosítani a mérés egyes szakaszai között. Felmerült, hogy markerek elhelyezése helyett a biológiai mérésekkel szükséges lehet valamilyen elektronikus jel alkalmazásával szinkronizálni a szoftvert más kísérleti eszközökkel.

A jövőbeli kísérletekben, a vetítőrendszerbe integrált pupillakövetéssel mindkét szem szinkronizált követésére lesz szükség. Ennek lehetőségét a jövőben a szoftverben

meg kell teremteni, hogy egyszerre két kamerát is tudjon kezelni, két szem paramétereit beállítani és két pupillát követni. Ebben az esetben a program kezelését egy fejlettebb grafikus felület szintén jelentősen megkönnyítené.

7 Összefoglalás

A pupilla mérete idegrendszeri állapotokról, pozíciója pedig a tekintet irányáról szolgáltat hasznos információt a biológiai kísérletekben. Ezért kutatásom célja az volt, hogy a korábban tervezett, egérszemet optimálisan stimuláló vetítőrendszeremet pupillakövetési funkcióval kiegészítsem.

Az egér szemmozgásairól szóló szakirodalom áttekintése alapján rögzített fejű egereken horizontális szemmozgások jelenléte várható. Megismerkedtem a videó-okulográfiás mérések alapjaival, a kísérlet menetével, a szükséges eszközökkel és az azokkal szemben támasztott követelményekkel. A pupillakövetéshez szükséges egy fényforrás és egy kisméretű kamera, amelyek az egér által nem érzékelt infravörös hullámhossztartományban működnek.

A pupillakövetés szoftveres megvalósításához áttekintettem az ingyenes, nyílt forráskódú lehetőségeket, majd kiválasztottam a számomra legmegfelelőbbet: az *EyeLoop* alkalmazható egereken, folyamatos és előre felvett videókon történő követésre is alkalmas, valamint egy egyszerű grafikus felhasználói felülettel is rendelkezik. Saját funkciókkal kiegészítve, mint például a detektálás leszűkítése a szem területére, a szoftver képes pontosan detektálni a pupillát és a szaruhártya fényvisszaverődéseit egerekről készült videókon sötét és világos pupilla esetén is.

A pupillakövető rendszerhez egy kisméretű kamerát és egy fókuszálható objektívet, valamint két infravörös LED-et használtam. A kiválasztott kamera kezeléséhez saját kódot írtam, amelyben a gyártó által biztosított vezérlési függvényeken túl egy iteratív autoexpozíciós algoritmust is implementáltam, ami lehetővé teszi a kamera automatikus alkalmazkodását a kísérlet fényviszonyaihoz.

Az élő állatokon történő teszteléshez két egeret alkalmaztam, melyek szemei különböző mértékben voltak pigmentáltak. Az egerek egy forgó gramofonon álltak, ami biztosította számukra a mozgás lehetőségét, fejük rögzített volt. A vizsgált 10 perces időtartományban megfelelően beállított paraméterek esetén a szoftver tudta követni a pupillát, kevés látványosan kiugró mérési hibával. A rögzített adatok elemzésével bemutattam, hogy felhasználásukkal biológiailag releváns következtetések vonhatóak le a pupilla méretének változásáról és a pupillaközéppont elmozdulásairól. Továbbá a

pupillakövető rendszer alkalmazásával videóra vettem és számszerűsítettem a pupilla méretének változását a fény hatására történő reflexes összehúzódás során.

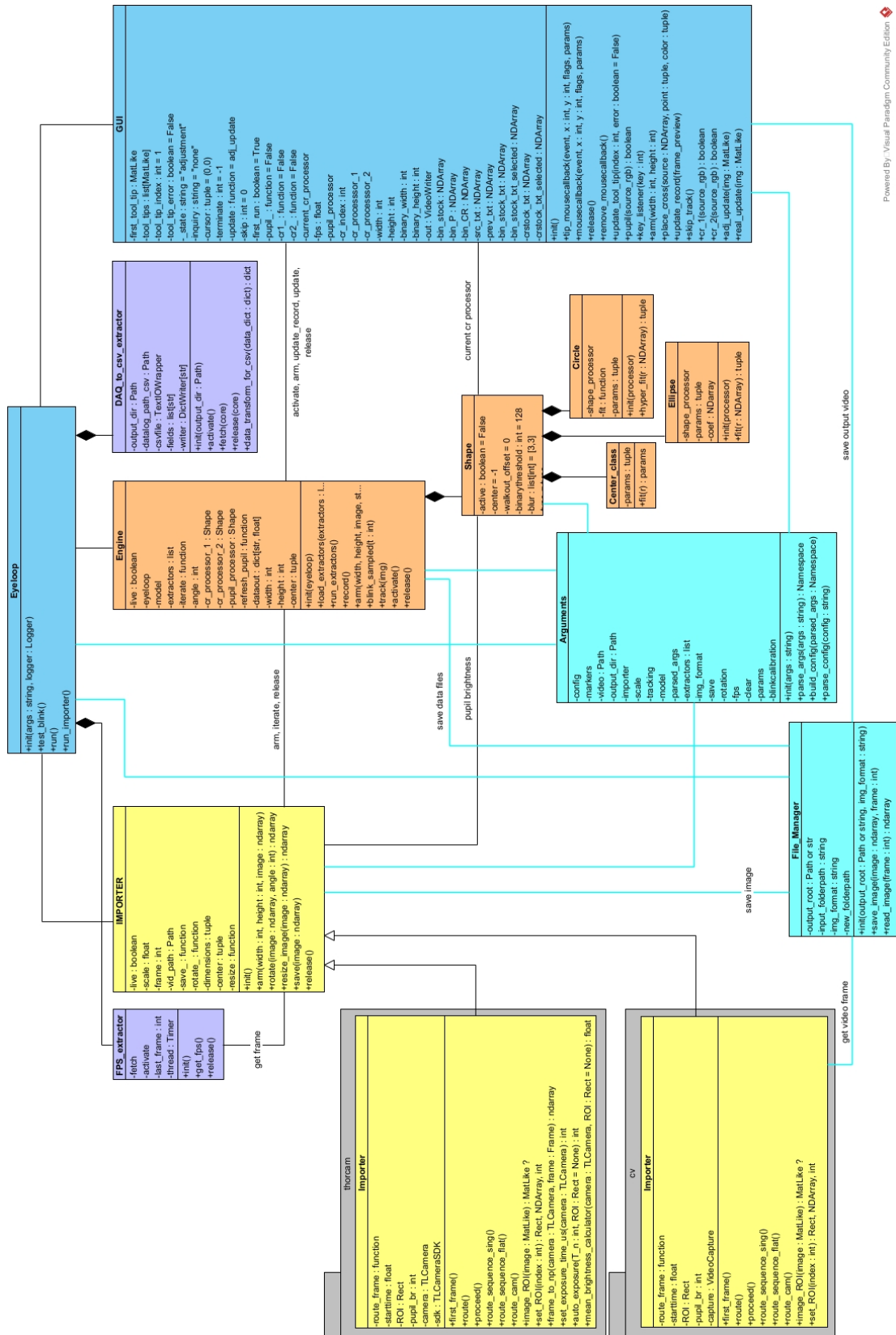
A fejlesztés célja, hogy a pupillakövető rendszer a vizuális információt szolgáltató vetítőrendszerrel is kombinálható legyen, ez egy dikroikus tükörrel megvalósítható. Az összeállított kísérleti elrendezés megmutatta, hogy végezhető vetítés és követés párhuzamosan, egymástól függetlenül, zavaró hatások nélkül. Ahhoz azonban, hogy a vetítőrendszer kis méreteihez a pupillakövetés igazodni tudjon, a rendszer elemeinek kisebbre cserélése és az elrendezés módosítása szükséges. A mechanikai nehézségek megoldását követően lehetséges lesz a kombinált rendszer tesztelése egereken. További terveim között szerepel a szoftver kiegészítése a tekintet irányának kiszámításával, valamint az ehhez szükséges kalibrációs folyamat kidolgozása egerek részére.

Irodalomjegyzék

- [1] H. L. Payne és J. L. Raymond, „Magnetic eye tracking in mice”, *eLife*, köt. 6, o. e29222, szept. 2017, doi: 10.7554/eLife.29222.
- [2] E. Ambrad Giovannetti és E. Rancz, „Behind mouse eyes: The function and control of eye movements in mice”, *Neuroscience & Biobehavioral Reviews*, köt. 161, o. 105671, jún. 2024, doi: 10.1016/j.neubiorev.2024.105671.
- [3] M. Land, „Eye movements in man and other animals”, *Vision Research*, köt. 162, o. 1–7, szept. 2019, doi: 10.1016/j.visres.2019.06.004.
- [4] F. Raudies, „Optic flow”, *Scholarpedia*, köt. 8, sz. 7, o. 30724, júl. 2013, doi: 10.4249/scholarpedia.30724.
- [5] R. S. Hessels, D. C. Niehorster, M. Nyström, R. Andersson, és I. T. C. Hooge, „Is the eye-movement field confused about fixations and saccades? A survey among 124 researchers”, *Royal Society Open Science*, köt. 5, sz. 8, o. 180502, aug. 2018, doi: 10.1098/rsos.180502.
- [6] M. Lappe és K.-P. Hoffmann, „Optic Flow and Eye Movements”, in *International Review of Neurobiology*, köt. 44, M. Lappe, Szerk., in *Neuronal Processing of Optic Flow*, vol. 44. , Academic Press, 2000, o. 29–47. doi: 10.1016/S0074-7742(08)60736-9.
- [7] A. F. Meyer, J. O’Keefe, és J. Poort, „Two Distinct Types of Eye-Head Coupling in Freely Moving Mice”, *Current Biology*, köt. 30, sz. 11, o. 2116–2130.e6, jún. 2020, doi: 10.1016/j.cub.2020.04.042.
- [8] R. Mazziotti és mtsai., „MEYE: Web App for Translational and Real-Time Pupillometry”, *eNeuro*, köt. 8, sz. 5, szept. 2021, doi: 10.1523/ENEURO.0122-21.2021.
- [9] J. M. Furman és F. L. Wuyts, „Chapter 32 - Vestibular Laboratory Testing”, in *Aminoff’s Electrodiagnosis in Clinical Neurology (Sixth Edition)*, M. J. Aminoff, Szerk., London: W.B. Saunders, 2012, o. 699–723. doi: 10.1016/B978-1-4557-0308-1.00032-7.
- [10] S. Remtulla és P. E. Hallett, „A schematic eye for the mouse, and comparisons with the rat”, *Vision Research*, köt. 25, sz. 1, o. 21–31, jan. 1985, doi: 10.1016/0042-6989(85)90076-8.
- [11] M. de Jeu és C. I. D. Zeeuw, „Video-oculography in Mice”, *JoVE (Journal of Visualized Experiments)*, sz. 65, o. e3971, júl. 2012, doi: 10.3791/3971.
- [12] G. H. Jacobs, J. Neitz, és J. F. Deegan, „Retinal receptors in rodents maximally sensitive to ultraviolet light”, *Nature*, köt. 353, sz. 6345, o. 655–656, okt. 1991, doi: 10.1038/353655a0.
- [13] E. D. Guestrin és M. Eizenman, „General Theory of Remote Gaze Estimation Using the Pupil Center and Corneal Reflections”, *IEEE Trans. Biomed. Eng.*, köt. 53, sz. 6, o. 1124–1133, jún. 2006, doi: 10.1109/TBME.2005.863952.
- [14] T. Sakatani és T. Isa, „PC-based high-speed video-oculography for measuring rapid eye movements in mice”, *Neuroscience Research*, köt. 49, sz. 1, o. 123–131, máj. 2004, doi: 10.1016/j.neures.2004.02.002.
- [15] E. S. Dalmaijer, S. Mathôt, és S. Van der Stigchel, „PyGaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments”, *Behav Res*, köt. 46, sz. 4, o. 913–921, dec. 2014, doi: 10.3758/s13428-013-0422-2.

- [16] R. Sadeghi, R. Ressmeyer, J. Yates, és J. Otero-Millan, „Open Iris - An Open Source Framework for Video-Based Eye-Tracking Research and Development”, 2024. április 22., *bioRxiv*. doi: 10.1101/2024.02.27.582401.
- [17] S. Arvin, R. N. Rasmussen, és K. Yonehara, „EyeLoop: An Open-Source System for High-Speed, Closed-Loop Eye-Tracking”, *Front. Cell. Neurosci.*, köt. 15, dec. 2021, doi: 10.3389/fncel.2021.779628.
- [18] M. Hoshino, K. Uesugi, N. Yagi, S. Mohri, J. Regini, és B. Pierscione, „Optical Properties of In Situ Eye Lenses Measured with X-Ray Talbot Interferometry: A Novel Measure of Growth Processes”, *PLOS ONE*, köt. 6, sz. 9, o. e25140, szept. 2011, doi: 10.1371/journal.pone.0025140.
- [19] C. P. Heesy, „On the relationship between orbit orientation and binocular visual field overlap in mammals”, *The Anatomical Record Part A: Discoveries in Molecular, Cellular, and Evolutionary Biology*, köt. 281A, sz. 1, o. 1104–1110, 2004, doi: 10.1002/ar.a.20116.
- [20] „01. előadás - Programozási Ismeretek jegyzet”. Elérés: 2024. szeptember 21. [Online]. Elérhető: <https://okt.inf.szte.hu/progism/eloadas/ea01/>
- [21] „Thorlabs Camera Python API Reference”. Thorlabs Scientific Imaging, 2022. április 26.
- [22] Y. su, J. Y. Lin, és C.-C. J. Kuo, „A Model-based Approach to Camera’s Auto Exposure Control”, *Journal of Visual Communication and Image Representation*, köt. 36, jan. 2016, doi: 10.1016/j.jvcir.2016.01.011.
- [23] E. D. Guestrin, „Remote, Non-contact Gaze Estimation with Minimal Subject Cooperation”, Thesis, 2010. Elérés: 2024. szeptember 26. [Online]. Elérhető: <https://tspace.library.utoronto.ca/handle/1807/24349>

Függelék



1. függelék: Teljes UML osztálydiagram

	fekete egér		szürke egér	
	vízszintes (pixel)	függőleges (pixel)	vízszintes (pixel)	függőleges (pixel)
ROI mérete	200	182	208	186
pupillapozíció				
átlag	108,7±3,3	94,6± 3,1	109,5±6,9	91,5±3,4
minimum	96,4	56,4	71,8	68,7
minimális elmozdulás	-12,3	-38,2	-37,7	-22,8
maximum	132,5	114,5	128,3	112,0
maximális elmozdulás	+23,8	+19,9	+18,8	+20,5
terjedelem	36,1	58,1	56,5	43,2
pupillaméret				
	fél nagytengely (pixel)	fél kistengely (pixel)	fél nagytengely (pixel)	fél kistengely (pixel)
átlag	49,9±3,4	46,5±3,6	42,4±5,2	40,6±6,4
minimum	19,4	14,48	2,4	1,8
maximum	74,5	62,6	61,6	61,1
terjedelem	59,2	59,3	55,1	48,1

2. függelék: Élő egér tízperces pupillakövetés méréseinek statisztikai adatai