Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Műszertechnikai és Automatizálási Tanszék

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

AUTOMATIZÁLT VESSZŐDETEKTÁLÓ RENDSZER

Szerzők: **Fridrik Balázs**
villamosmérnök BSc. szak, IV. évf.

Mikola Máté Patrik
mérnökinformatikus BSc. szak, IV. évf.

Konzulens: **Dr. Schuster György**
egyetemi docens

Tartalom

Absztrakt 4

Bevezetés 5

1. Az ötlet elhelyezése 6

1.1 Olimpiai lézeres megvalósítás 7

1.2 Olimpiai képfeldolgozási megvalósítás 7

1.3 Saját módszertan 8

2. Logikai felépítés 9

2.1 Kezdeti változtatások 10

3. Hardver 12

3.1 Mechanika 12

3.1.1 Vesszőfogó 12

3.1.2 Tartó konzolok 14

3.1.3 Rögzítések 15

3.2 Elektronika 16

3.2.1 Raspberry 16

3.2.2 Kamerák 17

3.2.3 Trigger - piezo 18

3.3 Eszközvédelem 21

4. Szoftver 22

4.1 Tesztkódok 23

4.1.1 Kamerakezelés 23

4.1.2 Halszem-effektus 23

4.1.3 Méretezés 24

4.2 Detektálás 24

4.2.1 Objektum detektálás 25

4.2.2 Canny éldetektálás 25

4.2.3 Éldetektálás mellőzése 26
4.3 Környezeti zajok kezelése 27
4.3.1 Konvolúciós élsimítás 27
4.3.2 Képek differenciálképzése 27
4.4 A végleges kód 28
5. Tesztelés 30
5.1 Szoftveres lefutás 30
5.2 Pontosság 31
6. Fejlesztési lehetőségek, potenciál 32
6.1 Hardveres fejlesztések 32
6.2 Szoftveres fejlesztések 33
6.3 Fejlesztési potenciál 33
Összegzés 34
Irodalomjegyzék 35
Mellékletek 37

Absztrakt

Napjainkban soha nem látott mértékekben alkalmazunk elektronikát automatizálási célokra. Az automatizált eszközök körülvesznek minket hétköznapjainkban, így a munkánkban, hobbijainkban és a sport tevékenységeinkben is. A TDK dolgozat egy saját elképzelés alapján működő és a valóságban is kivitelezett automatizált vesszőfogó táblára épül. Célja az eszközünk prezentálása mellett bemutatni a modern íjászatban használt automatizálási eszközök egy szegmensének korszerűségét és felépítését. A jelenleg használatban levő különböző rendszereket átfogólag összehasonlítani a saját ötletünkkel. Továbbá körültekintően kitérni a logikai és fizikai felépítéseik különbségére. Végkifejletként elhelyezni az alkalmazott technikát a napjaink íjászatában használt vessződetektáló, automatizált találat megjelenítő rendszerek között.

Az projekt célja egy használható és olcsó automatizált íjász vesszőfogó megépítése, mely a gyűjtött adatokkal hozzájárulhat egy hobbi vagy akár versenysport íjász edzéseinek effektivitásához. A projekt komplexitása miatt első lépésben csak egy vessző helyzetének meghatározása volt a cél, a dolgozat ezt fogja bemutatni. Kitér a felmerülő problémákra, igényekre és ezeknek a megoldásaira. Részletesen bemutatja a hardver felépítését, anyaghasználatát és ezeknek a fontosságát. Továbbá cél bemutatni az eszközben rejlő lehetőségeket, továbbfejlesztési potenciált.

A megépített céltábla lokalizálja a belelőtt vessző helyzetét és eltárolja azt. A vessző pozíciójának meghatározásához egy a céltáblára szerelt kamerarendszert használunk. A Python nyelven írt szoftver a képfeldolgozáshoz az OpenCV és más nyílt forráskódú könyvtárakat használja.

Bevezetés

A dolgozat kivonatában már olvasható volt, hogy egy saját elven működő és használható automatizált íjász segédeszközt szerettünk volna megalkotni. Az ötletnek személyes vonatkozása is van, mivel egyikünk (F.Balázs) több, mint 8 éve íjászkodik amatőr szinten. A vessződetektáló íjászcél nem újdonság a sport számára azonban folyamatosan jelennek meg új módszerek és fejlesztések a már használatos technikákhoz. Az olimpiai íjászatot leszámítva számtalan íjászati stílus létezik, melyek kevésbé ismertek és kevesebb modern technológiával rendelkeznek. A sport azon szegmensében és szintjén, amiben személyes tapasztalatokkal is rendelkezünk, egyáltalán nem jellemző az automatizált eszközök használata. Célunk egy gyakorlati, elérhető, prototípus szerű eszköz megalkotása volt, ami pótolná az ilyen eszközök hiányosságát ebben a szegmensben. A TDK dolgozatunkban a vessződetektálás és hardveres megvalósítás bemutatása mellett szeretnénk a jelenleg is használatban levő eszközök között is elhelyezni az ötletet és a megvalósított eszközt.

1. Az ötlet elhelyezése

A problémák kérdésköre széles, jellemzően attól függ, hogy a táblát milyen célra és milyen felhasználási módra szeretnénk elkészíteni. Azonban vannak közös pontok, ilyen például a pontosság. Magától értetődő, hogy mivel a pontosság mértékével versenyeznek a sportolók, így a vessződetektáló, megjelenítő rendszer sem mehet ennek rovására. Az élsportban használt eszközök pontossága akár kisebb, mint 1 mm is lehet, ez egy átlagos vessző vastagságának (5-6 mm) csupán $\frac{1}{5}$-e. A hobbisportban nem szükséges ennyire pontos eszközök használata, azonban a nyílvessző méretének sokszorosával megegyező pontatlanságok már nem férnek bele ebbe a kategóriába sem. Az általunk meghatározott célpontosság emiatt egy vessző vastagságának kétszeresére (1,2 cm) esett, ez az érték egy általános céltáblán – nem versenykörülmények között – elenyészőnek mondható.

A másik számottevő tényező az elérhetőség. Itt még nagyobb szakadék áll az élsport, illetve a hobbisport között. Az általunk készített prototípus egy elérhető árú eszközt szeretne bemutatni, mely nem többszöröse egy gyakorló íjászfelszerelés összegének. Az említett összeget meghatározni nehezen lehet a különféle íj és íjászati típusok miatt. Viszonyítási alapnak a személyes gyakorló felszerelést vettük alapul, ami egy (Kassai) visszacsapó íjból, alumínium gyakorlóvesszőkből és egy polifoam vesszőfogóból áll. [1]

A hobbi íjászatban nem feltétlenül egységesek az edzéseken megjelenő íjászfelszerelések és körülmények. Emiatt érdek volt a hordozhatóság és a széleskörű használhatóság is, különböző íjakhoz és íjászpályákhoz. Ahhoz, hogy a felsorolt problémákat és megoldásaikat jobban kifejthessük, először el kell helyeznünk az ötletünket, illetve annak alkotó elemit az íjászat eszköztárában.

Abból kifolyólag, hogy az általunk megalkotni kívánt céltárgy egy sporthoz köthető, a technikai tulajdonságok előtt elengedhetetlen sportolói szemszögből is megvizsgálni a kitűzött célt. A valódi használhatósághoz, illetve ahhoz, hogy megfelelően elhelyezzük a projekt egyediségét és praktikusságát, meg kell nézni milyen igények és nehézség állnak fenn a kigondolt ötlettel kapcsolatosan. Az igényekhez hasonlóan, a nehézségek is széles spektrumon mozognak a sport intenzitásától függően. Akármilyen sporteszközt nézünk, nagyon nehéz – általában nem is cél – olyan optimalizált eszközt gyártani, amely mind az élsport, mind a hobbisport igényeit is képes kielégíteni egyidőben. Emiatt érdemes egészen a kezdő, hobbi íjászattól a profi, olimpiai íjászat szintjéig vizsgálni a tényezőket, majd meghatározni melyik véglet felé érdemes igazítani a projekt kimenetelét.

A hobbi íjászatban használt íjász eszközöknek és kiegészítőknek egyik legmeghatározóbb tulajdonságuk az áruk. Itt nem elsődleges cél a tökéletes személyreszabhatóság, a maximális pontosság, ellentétben az élsporttal. Fontosabb inkább az eszközök hordozhatósága, elérhetősége és a fogyóeszközök pótolhatósága. Az élsportban minden a pontosságról és az adott sportág kihívásainak legyőzéséről szól. Így nem csak az íjakban, vesszőkben és kiegészítőkben vannak nagy különbségek, hanem a vesszőfogók és az edzések módjában is.

A vesszőfogókra és a vessződetektálásra fókuszálva érdemes megnézni a sport csúcsának, azaz az olimpiai íjászat technológiájának arzenálját. Itt a versenyzők egy 122 cm átmérőjű 75-80°-ban döntött, rétegzett polietilénhab (pl.: Etilén–vinil-acetát (EVA) szintetikus hab) körtáblára lőnek, 70 méteres távolságból. A táblára papír és vinil célkörös lap van felfogatva melyen tíz egyenlő méretű körgyűrű tagolja a kapható pontokat. A betalált vesszők helyzetét sokáig csupán szemmel vizsgálták, majd 2016-ban a riói olimpián debütált először egy lézeres alapú vesszőmeghatározó rendszer. [2] [3]

1.1 Olimpiai lézeres megvalósítás

A rendszer 0,2 mm-es pontossággal tudta megmondani a vessző pozícióját. Elsősorban a nézők számára fejlesztették, annak érdekében, hogy a lövés pillanatában megjeleníthető legyen a kijelzőkön a találat pontos helye. Működése a lézeres távolságmérés, illetve a lézerfüggönyös árnyékdetektáláson alapult. A technika, pontossága ellenére nem maradt sokáig az élvonalban. A nehéz keretek és technológiai integrációja miatt, a debütálása után a következő olimpián már új módszertant használtak. [2]

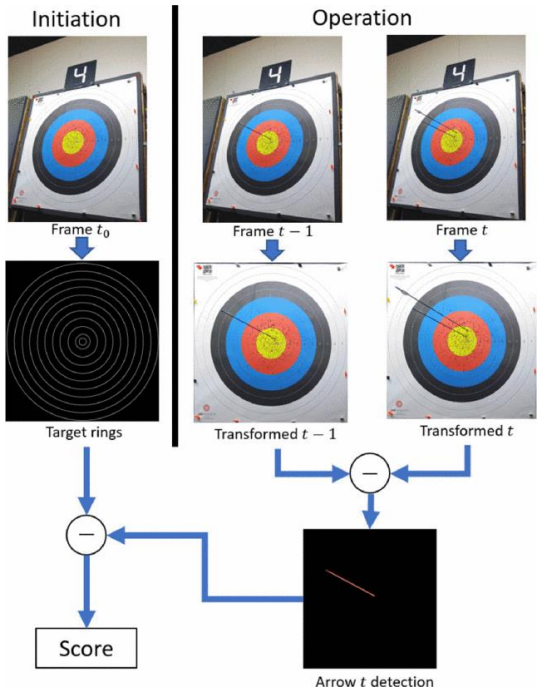
1.2 Olimpiai képfeldolgozási megvalósítás

A legújabb módszer 2021-ben a tokiói olimpián került bemutatásra, amely már kamerákra támaszkodva, a képfeldolgozás segítségével határozza meg a vessző helyzetét. Ez a technológia már jóval könnyebben volt integrálható az olimpiai íjászatban. A speciális kamerarendszer itt a táblával szemben, külső perspektívából figyelte a táblát. A professzionális kamerákkal készült felvételekből, objektumdetektálással pontosan megkereshető a vessző és annak becsapódási pontja. A kamerarendszer előzetes kalibrációt követően képes ezt a pontot nagy pontossággal elhelyezni a tábla síkján. A kamerarendszert a tokiói olimpia után a nagyközönség számára is elérhetővé tették, azonban az ára (~ 2500 USD) és a komplexitása messze nem a hobbi vagy haladó sportot célozza meg. [2] [4] [5]

1.3 Saját módszertan

Jól látható, hogy az automatizált vesszőfogók elképzelése nem kiforrott, egyelőre csak az élsport számára érhető el, ott sem a „hétköznapi” gyakorlásokhoz. Többek között emiatt is kezdtünk el ezzel az ötlettel foglalkozni. Így alakult ki végleges célként egy olyan automatizált nyílvessző meghatározó rendszer építése, ami árban elérhető a hobbi sportolók számára is és segít nekik a mindennapi edzések során leadott találatok kielemzésével.

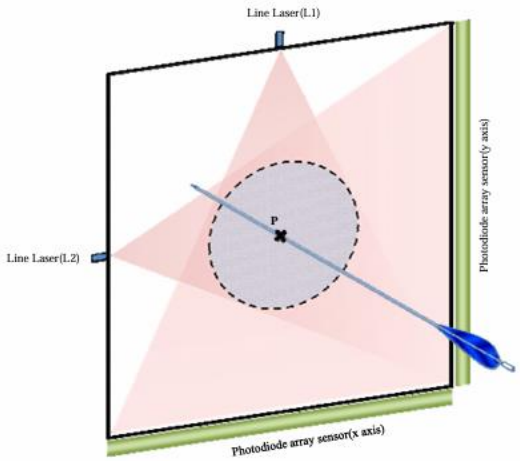
A célkitűzés után legelőször magát a megoldás módszertanát kellett kiválasztani. Kettő fő csoportba lehetett sorolni a felmerülő lehetőségeket. Az egyik főcsoport a kamerát használó különböző képfeldolgozási módszerek voltak. Ide tartozik például a korábban említett, olimpián jelenleg is használatos, külső perspektívából való kamerás megfigyelés. A kamerát ebben az esetben különböző helyekre is pozícionálhatjuk. Rögzíthetjük közvetlenül a táblához, helyezhetjük a táblától elkülönítve is egy fix betekintési szögben. Az elhelyezés lényege, hogy a kamera rálásson a céltábla felületére. Az olimpiai íjászatban talán legkézenfekvőbb módon, a céltáblával egyenes vonalban, 3 méteres távolságban van elhelyezve a kamerarendszer. A kamerás megvalósítás előnyeihez sorolható a hardveres feltételek egyszerűsége (általában), illetve a javarészt szoftveres logika átkonfigurálhatóságának lehetősége. [2] [6] [7]



1. ábra - Képfeldolgozással megvalósított vessződetektálás [7]

A második főcsoport a szenzoros megoldásokat csoportosítja. Ebben az esetben nem képfeldolgozás által, hanem különböző szenzorok értékváltozásai alapján lokalizálják a vesszőt. A módszertant tovább bontva lehetséges lézeres távolságméréssel lokalizálni a vesszőt,

ahol az adott szenzoroktól való távolság határozza meg a vessző pozícióját. Illetve lehetséges lézerfüggöny alkalmazásával a függönyben képzett árnyéksávok metszetét keresni. Ennél a módszernél a vessző által kitakart lézersáv fogja meghatározni annak helyzetét. Bármelyik lézeres meghatározást is nézzük, magasszintű precizitás kell a hardveres felépítéshez. A szenzorok tulajdonságai is meghatározóak, így árazás vagy épp komplexitás terén szintén az élsport lehet a célközönség. [8]



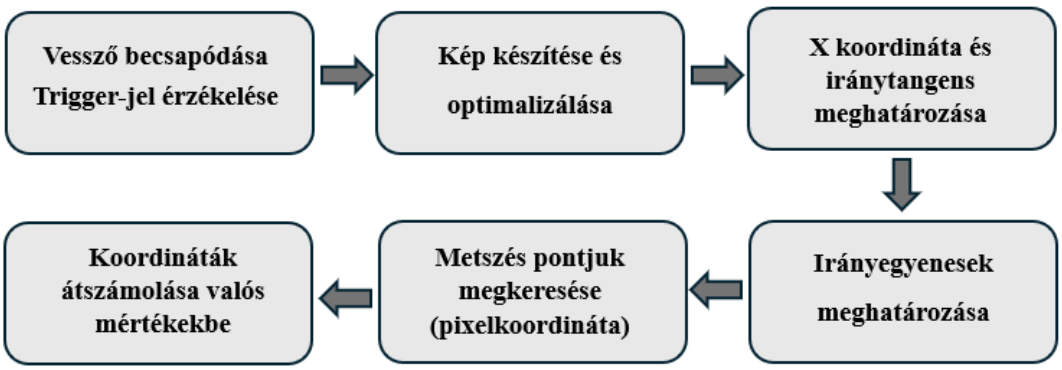
2. ábra - Lézeres módszertannal megvalósított vessződetektálás [8]

A kezdeti ötleteléskor azonban felmerült az ultrahangos vagy épp a mágneses alapon való detektálás is. Az ultrahangos módszer a tábla anyaga, inhomogenitása miatt nem lehetséges. A vessző becsapódási pontjainál az anyag összetömörül, így nem megvalósítható ez a módszer. A mágneses detektálás a vessző hegyét találná meg a tábla anyagában. Ennek egyik hátránya, hogy a különböző becsapódási szögek esetén a hegy nem lesz egyvonalban a tábla síkjában levő becsapódási ponttal. A mágneses detektálás az érzékelő panel elhelyezése miatt se lehetséges megoldás, mivel legjobb pozíciója szerint, azaz tábla hátoldalán veszélyeztetve van az erősebb, táblán áthaladó lövésektől. Az általunk használt módszer a felsorolt lehetőségre alapozva merít mind a 2 főcsoport eszköztárából. Célja, hogy ne bonyolódjon bele a lézeres módszertan erős hardveres komplexitásába, viszont a kamerák alkalmazása esetén se legyen szükség professzionális gépekre.

2. Logikai felépítés

Annak érdekében, hogy a nyílvessző helyzetét meg tudjuk állapítani a táblában, a tábla síkját és a becsapódási pontot kell vizsgálnunk. A tábla síkját 2D-s formaként kezelve feloszthatjuk koordinátarendszer-szerűen. A koordináta-rendszer egységét a legkisebb detektálható elemre

kell választani, ez – a képfeldolgozás esetén – a pixel. A pixel alapon feloszlott 2D-s alakzaton, az ötletünk alapján, koordináta alapon határoznánk meg a vesszőt. A vessző helyzetét tehát kettő pixelkoordináta (X_{pixel}, Y_{pixel}) határozna meg, amit megfelelő átváltásokkal valós koordinátákra alakítva kapjuk meg a vessző pozícióját (X_{real}, Y_{real}). A koordináta alapú helyzetmeghatározás alapvetően a lézeres módszertanban jelenik meg, ahol például a lézerárnyékból számítják a koordinátát. A mi esetünkben a készített kameraképekből szeretnénk kinyerni a szükséges koordinátákat. Emiatt optimális, ha a kamerák párhuzamosak a tábla síkjával, mivel így csupán egy koordinátát „lát" egy adott kamera. Ennél a módszernél legalább kettő kamerára van szükség, hogy a vesszőt kettő külön pozícióból figyelje. A vessző becsapódásakor szükséges valamilyen „trigger-jel", mely hatására a kamera képet készít. A kamera által készített képen először a vesszőnek a becsapódási pontját kell megkeresni. A megírt kód megkeresi ezt a pontot, és a pont X koordinátájából iránytangenst számít. A kapott szögből és a kamera saját pozíciójából irányvektort lehet húzni a kamera pozíciójából kiindulva. A kapott irányvektor meghatároz egy olyan sávot, amelyen biztosan rajta van a vessző. A kapott irányvektor egyenletének átalakításával megkaphatjuk annak az egyenesnek az egyenletét, amelyen a vektor elhelyezkedik. Amennyiben ugyanezt a műveletsort végrehajtjuk egy másik pozícióba állított kamerával, kettő különböző egyenletű egyenest kapunk, amelyeken biztosan szerepel a vessző helyzete. Ha megfelelően pozícionáltuk a kamerákat, mind a valóságban, mind az elméleti koordináta rendszerünkben, akkor a kapott egyenesek metszéspontjánál lesz a vessző pontos helyzete.



3. ábra - Saját módszertan logikai folyamatábrája [saját ábra]

2.1 Kezdeti változtatások

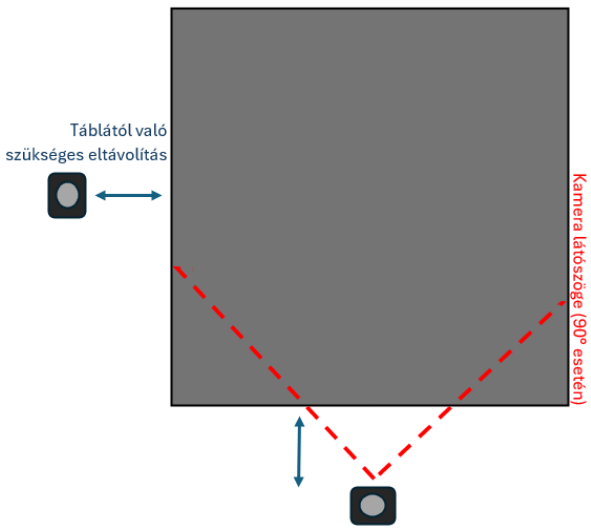
A fent leírt metódus szerint kettő darab kamera szükséges a helyzetmeghatározáshoz. A kezdeti tervezésnél ezek alapján kétkamerás rendszerre terveztük volna a projektet, azonban

több akadályba is ütköztünk. Az alapgondolat szerint a kamerák a négyzet (vesszőfogó) oldalfelezőinél lettek volna, annak két szomszédos oldalánál. A felvetés szerint a kettő kamerát az oldaltól való kellő eltávolítás után, saját képeik alapján tudtak volna az X és Y koordinátát meghatározni. A felépítéssel az első probléma a kamerák látószögének optimalizálásával volt. A széleslátószögű kamerák Halszem-effektusa miatt pont az általunk vizsgálni akart tartományokba esett volna a torzulás. A kislátószögű kamerák esetében a tábla teljes vagy túlzó lefedése érdekében, olyan mértékben kellett volna távolítani a kamerákat, hogy vagy pontosság, vagy a magas képfelbontási igény miatt a költségvetés rovására ment volna az elképzelés. Az alapelképzelés tovább gondolásakor esett a választás a sarkokba elhelyezett kamerákra. Előnye a látószögek jobb kihasználhatósága, így a felvetett problémák egy részének megoldása is.

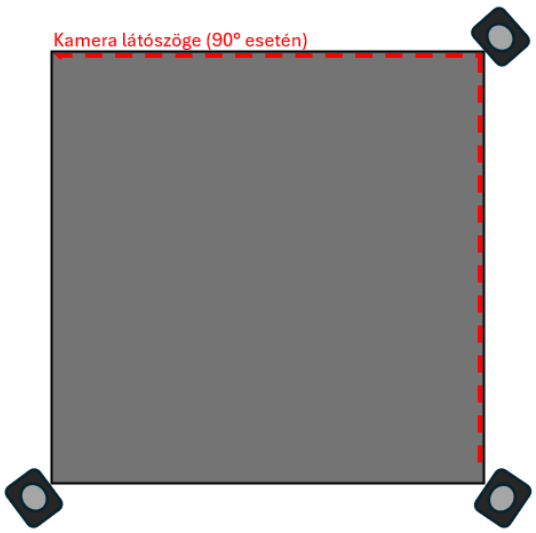
A következő döntés a széleslátószögű kamerák használata volt. Ezeknél a kameráknál a látószög akár 180° is lehet, azonban cserébe erős „Halszem-effektussal" rendelkeznek. A Halszem-effektus széleslátószögű (halszem) objektívek esetén a kép szélén keletkező torzulást jelenti. A sarkokba való kamera elhelyezésnek köszönhetően viszont, az alaptorzítás a vizsgált tartomány szélére esett. Itt már kis szakaszon lehetett vele számolni és korrigálni. [9]

A végleges logika felépítésekor a következő problémakör a több vessző által képzett árnyékok, vakfoltok. A projekt egyik említett célja volt, hogy hobbi felhasználáshoz alkalmas módon, akár 5-6 vessző helyzetének egyidejű meghatározására is képes legyen. A harmadik vessző helyzeténél azonban már fellép az a probléma, hogy az előző kettő vessző árnyékába esik, így egyik kamera sem lát rá. Azonban teljes kitakarás nélkül is nagyban ronthatja a pontosságot, amennyiben nem vagy nem megfelelően lát rá valamelyik kamera az adott vesszőre. Továbbá a kamerák táblához való közelsége miatt, az is nagy vakfoltot eredményezhet, ha a kamera elé, ahhoz kellően közel érkezik be a vessző. Ezen problémák kiküszöbölése érdekében döntöttünk egy harmadik kameraállás mellett. A harmadik kamera segítségével szinte teljesen kizárhatjuk a vakfoltokat, 5-6 vesszőmennyiség esetében. A túl közeli találatok pontosságát a kamerák távtartásával és a harmadik segédkamera segítségével javítottuk. A legtávolabbi érvényes találatok pontosságát elméletben a kamerakép felbontása és a vesszőtől mért távolsága határozza meg. Utóbbin javítani szoftveresen nem lehet, csak is a hardver módosítása, fejlesztésével lehetne. A végleges tervezetben ezek alapján már három kamerás rendszerben szándékoztuk megvalósítani a projektet, amelyben kettő főkameraként és egy segédkameraként funkcionál. A két főkamerával történik alapvetően a vessződetektálás, míg az egyik főkamerával szemben elhelyezett mellékkamera csak szükség esetén használatos.

Az utolsó logikai kérdés a kamerákkal kapcsolatosan a kamerák dőlésszöge, síkból való kiemelkedése volt. A tervek alapján teljesen párhuzamos elrendezést szerettünk volna a tábla síkjával, azonban az első tesztek és látképek alapján tudtuk csak véglegesen, hogy számunkra milyen értékek lennének kedvezőek. A tábla síkjával tisztán párhuzamos álláson igazítva, minimálisan felfele buktattunk a kamerákon, hogy megfelelően láthassák a tábla felületét. Annak érdekében, hogy a tábla mögött és mellett levő objektumok ne zavarhassanak be a készített képekbe, szükséges volt elszeparálni a tábla széleit a háttértől. Ehhez egy kontrasztos (fehér) színű keretet építettünk a vesszőfogó széleihez. A kontrasztos keret 5 cm-rel emelkedik ki a tábla síkjából, ezzel könnyen vizsgálható háttérterületet biztosítva a legtávolabbi találatokhoz is. A készült képeket később szoftveresen levágtuk a keret felső határánál, ezzel megkapva a véglegesen vizsgálandó fotót.



4. ábra - Kezdeti elrendezési terv [saját ábra]



5. ábra - Végleges kamera elrendezés [saját ábra]

3. Hardver

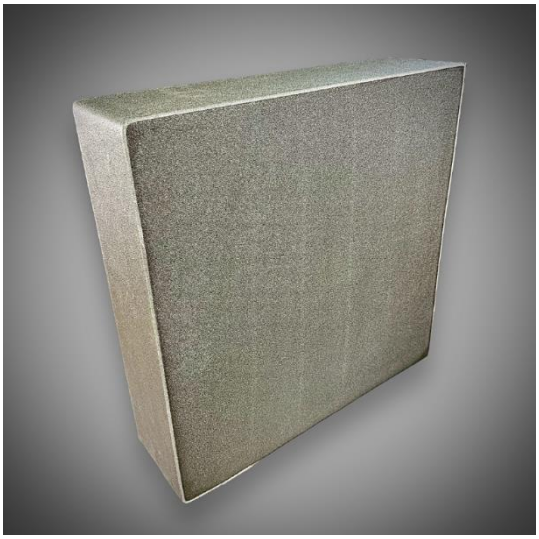
A prototípus szinten megépített projektnél is jelentős szerepet kaptak a hardveres tulajdonságok. A megfelelő kutatások és egyéni tapasztalatok segítségével sikerült olyan alkotó elemeket összeválogatni, melyek minden téren megfelelnek a projekt céljainak.

3.1 Mechanika

3.1.1 Vesszőfogó

A vesszőfogó kulcskérdés a projektben, ennek megválasztása elsőre egyszerűnek tűnhet, azonban megannyi dolognak kell megfelelnie a választott anyagnak, formának és méretnek.

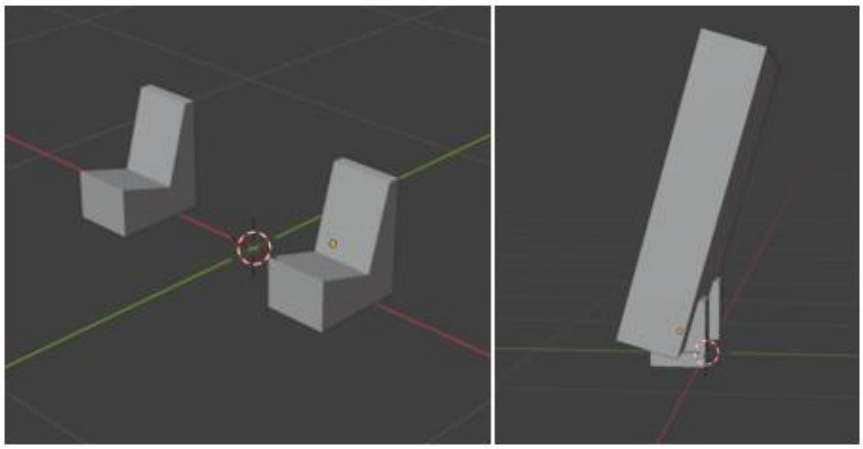
Anyagát tekintve egyre jobban szűkül a piac, az élsportban a 2000-es évek eleje óta csakis EVA és más nagy sűrűségű habokból készült célokat használnak. A hobbisportban szintúgy erősen a polifoam vesszőfogók dominálnak. A polifoam vesszőfogó előnyeihez sorolható a strapabíróság, a csekély súlya és az elérhetősége is. A projektünk szempontjából továbbá nem hátrány, hogy anyagából adódóan könnyen rögzíthetőek hozzá különböző alkatrészek. Nem véletlen, hogy az olimpiai, a hobbi sport és a vadászati célokra készített gyakorló célok is ezt az anyagcsaládot pártolják elsősorban. A másodlagosan elérhetőek szalma, préselt szalma, karton célok is, ezek azonban tulajdonságaik nagytöbbségében mind alulmaradnak a modern polifoam kockák mellett. Méretek és formák tekintetében már sokkal szélesebb skálán mozognak a táblák, céljuknak megfelelően specializálva. Az olimpiai íjászatban kerek, míg a hobbisportban és szabadgyakorlások esetében gyakran a négyzetes táblákat pártolják. A vadászati gyakorlások céljából kerülnek elő elsősorban a komplikáltabb 3D-s formák, a különböző állatok, speciális testek formájában. További fejtörést okoz az íjak és azok erejének széles palettája. Míg egy gyerek gyakorló visszacsapó íj 16-22# (font) kihúzási erővel rendelkezik, addig egy felnőtt vadászreflex íj ennek akár három-négyszere is lehet (60-70+ font). Egy professzionális csigás vadászíj által kilőtt nyílvessző akár 300 km/h feletti sebességet is elérhet. Sok esetben döntést kell hoznia a felhasználóknak is, hogy íjász felszereléseiket, eszközeiket kellően összehangolják. Számunkra fontos volt, hogy a táblát széleskörűen lehessen használni, íjtól és erejétől függetlenül (hétköznapi kereteken belül). A felsorolt anyagtechnikai előnyök és a negatívumok hiánya miatt projekthez polifoam négyzet vesszőfogót választottunk. Egy az általunk választott rétegzett EVA-hab tábla egyaránt használható gyerek és erősebb felnőtt íjakhoz is (font értékek). A vadászathoz használt reflex és csigás íjakból leadott lövés kisebb távon már áthaladna a vesszőfogón, azonban így is nagy szegmens sikerül lefedni a választott anyaggal és mérettel. A négyzetes szerkezetnek és a polifoam fúrhatóságának köszönhetően egyszerűbben lehetett előkészíteni a projekt fizikai részleteit (kamerák és a többi eszköz rögzítése). A táblára ezen felül különböző célformák, lőlapok nyomtathatók, melyekre kalibrációval a későbbiekben igazítható a szoftver is. A végleges vesszőfogó egy Freddo márkájú 80x80x16,5 cm-es szürke polifoam tábla lett. [6] [1] [10] [11]



6. ábra - Felhasznált vesszőfogó [1]

3.1.2 Tartó konzolok

A vesszőfogó kiválasztása után, annak rögzítésére kellett megoldást találni. A rögzítés kritériuma a minél kisebb elmozdulás és a hordozhatóság volt. Ennek megfelelően szintén egy egyszerű tartószerkezetet szerettünk volna. Végül egy kisebb talapzatra és egy hátsó támasztóra esett a választás. Ezek könnyen leszerelhetők a tábláról, a hordozhatóságot segítve. Az olimpiai és sport állványok általában robusztusabb, háromlábú, magasított koncepciót szoktak támogatni, azonban hobbi felhasználásra, illetve a prototípushoz praktikusabb volt egy költséghatékony és hordozható megvalósítást kitalálni. A talapzat tartó elemeit szintén 3D nyomtató segítségével készítettük el. A hátsó támasztónak egy csiszolt fa nyél szolgál.



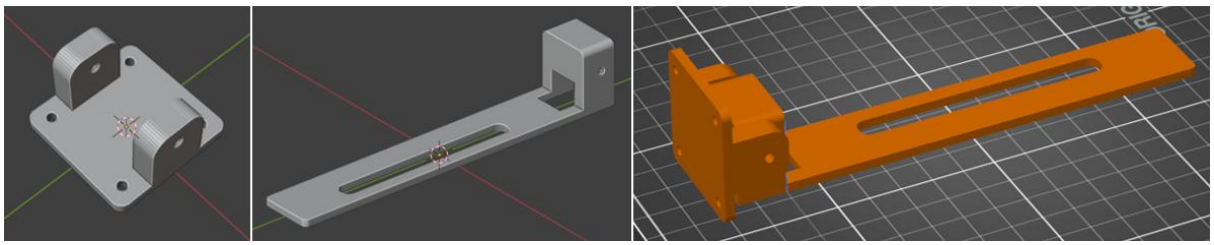
7. ábra – Talapzat [saját ábra]

Az alapok lefektetése után nekiállhattunk egyedivé tenni a meglevő hardveres feltételeket, illetve felműszerezni a vesszőfogót. A koncepció 3 darab kamera rögzítését követelte meg. A kamerák elhelyezkedése három sarokba a tábla síkjával nagyjából párhuzamosan történt. Ennek

megvalósításához egy keretet készítettünk a tábla köré, amelyre már könnyen rögzíthetőek voltak a kamerák. A keretet falécből, sarokvasakból építettük, rögzítését csavarokkal oldottuk meg a táblához. Az itteni anyaghasználat és rögzítési megoldások inkább prototípus jellegűek, mint véglegesek. Költségesebb anyaghasználattal, külön rögzítő modulokkal és a kamerák védelmével még tovább lehetne fejleszteni az eszközt, azonban elsődleges céljait így is tökéletesen ellátja. Az első próbalövések és korrigálások után kifejezetten stabil és jól funkcionáló keretet sikerült építeni a tábla köré.

3.1.3 Rögzítések

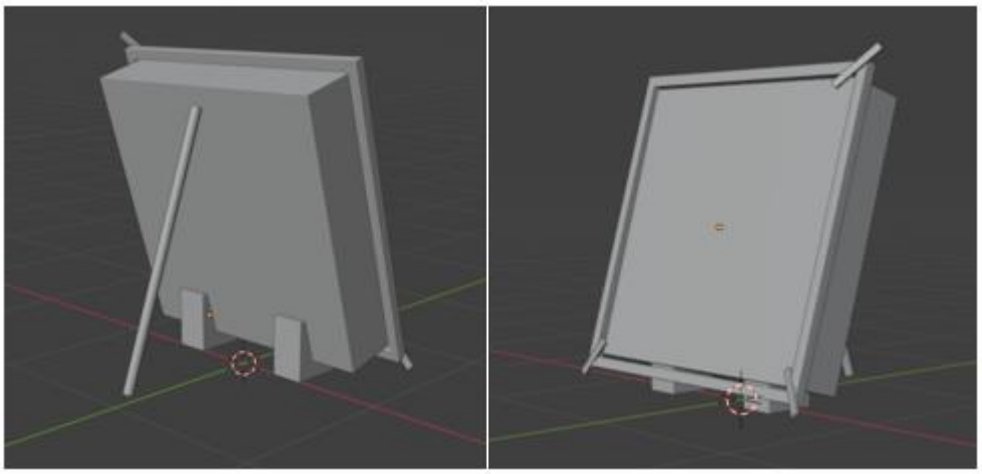
A kamerák rögzítésére, a sarokra laposvasakból toldásokat csavaroztunk, így a fém lapokon pozícionáltuk és rögzítettük a kamerákat végleges helyükön. Az így kapott síneken jelentősen könnyebben tudtuk pozícionálni a kamerákat a megfelelő látkép érdekében. A kameramodulok előre szerelt NYÁK-okon, USB csatlakozóval érkeztek. Nagy figyelmet kellett fordítani a modulok rögzítésére, hogy a kezdeti beállítás és konfiguráció után ne mozdulhassanak el a vesszők becsapódása hatására sem. Ennek kivitelezése érdekében egyedileg terveztünk hozzá tartó modulokat. A 3D-s modellek a Blender 4.2-es programmal készültek. A nyomtatást egy Creality Ender 3-as 3D nyomtató végezte, fekete PETG filamentből. A tervezett műanyag modulok lehetőséget adtak, hogy a vasalaton, mint sínen mozgassuk a kamerákat.



*8. ábra - Kamera rögzítő modul [*saját ábra*]*

Ahogy a csatolt képen látható, a modulok feje billenthetőre lett tervezve, így a kamera nézetét is tudtuk igazítani a tábla síkjához képest. A látkép beállításánál figyelembe kellett venni a kamerák távolságát a legközelebbi és legtávolabbi beérezhető találati ponthoz, és a kameraképek szimmetrikusságat a tábla átlóira. Továbbá azt, hogy minden kamera a tábla teljes síkját (kétoldali szélét) is lássa. A bepozícionálás után meglett a kameráknak végső helye. A végleges rögzítéseket egyszerűen metrikus csavarokkal és csavaranyákkal végeztük, ezzel is csökkentve az elmozdulás esélyét. A keret és kamerák véglegesítésével az elektrotechnikai eszközöket kellett összeépíteni.

9. ábra - Pozícionált kamera látképe [saját ábra]



10. ábra - Hardveres elemek összképe [saját ábra]

3.2 Elektronika

3.2.1 Raspberry

Az elektronikát nézve a projekt „agya” egy Raspberry Pi 4B egykártyás számítógép. A megfelelő számítógép kiválasztását, itt is kellő utánajárás előzte meg. A klasszikus mikrokontrollerek teljesítménye nem volt elegendő. A szükséges számítási teljesítmény és memória méreteiről más hasonló projektekből, tanulmányokból tájékozódtunk. A teljesítményi oldalról a Raspberry-k, azoknak is az újabb szériás verziói kerültek be a látóterünkbe. A „mini PC” -k árazása és teljesítményük túlzása miatt kikerültek a számítási körökből (pl.: NVIDIA Jetson Nano, Intel NUC). A végleges Raspberry modell konfigurációja egy Quad core 64-bit ARM-Cortex A72 processzorból, egy VideoCore VI 3D Graphics GPU-nól és 8Gb LPDDR4 RAM-ból épül fel. A kellő tárhelyet egy Samsung 128Gb-os SD kártyával biztosítottuk. A számítási kapacitása az elégséges kategóriába esik. Az eszközcsaládot előszeretettel használják hasonló képfeldolgozással foglalkozó projektekben. A hardvereszköz Linux-os környezete és

Python nyelv előnyben részesítése számunkra szintén pozitívum volt. Az rendszer felállítása során célul tűztük ki, hogy a projekt fejlesztését ne csak a hardver mellől lehessen végezni, hanem távolról is elérhető legyen a felület bármelyikünk számára. Ennek fő oka a tábla mérete és a kamerák fix helye volt. Erre a célra a TeamViewer program segítségével hostoltunk elérést a Raspberry-re. A friss telepítés után csatlakoztattuk a Pi-t egy interneteléréssel rendelkező hálózathoz, majd telepítettük a TeamViewer 15.55.3 H verziójú távoli asztal elérésére alkalmas programot. Ezt követően az első szoftveres lépéseket, például a kameraképek megjelenítését már távolról saját számítógépről végezhettük. A projekt fejlesztése során nagy előrelépés volt, hogy nem kellett minden alkalommal személyesen a hardver közelében lenni. [12] [13] [14] [15]

11. ábra - Raspberry Pi 4B [12]

3.2.2 Kamerák

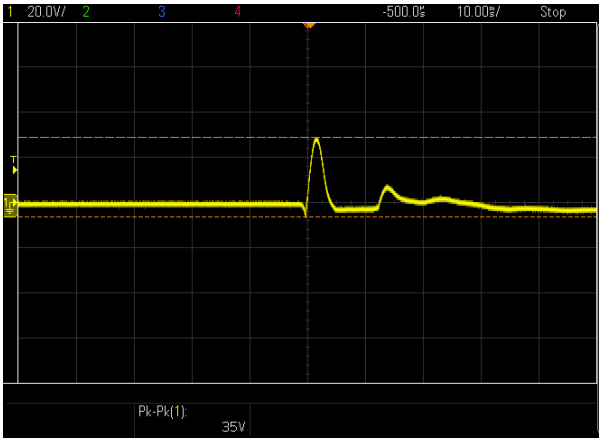
A projekt „szeme”-ként egy FIT0892 DFROBOT kameramodul szolgál. A kamera 1280x720 px felbontásra képes. A kamera látószöge 140°. A kamera által képviselt árkategóriában kiemelkedő, más rosszabb felbontású vagy drágább kameramodulok/webkamerákhoz képest. A felhasznált három kamera ugyanolyan. A Raspberry márkának megtalálhatóak saját kameramoduljai is, ezek azonban nem széleslátószögű modulok. Továbbá hasonló árkategóriában sokkal jobb tulajdonságokkal rendelkező kamerák állnak rendelkezésre. A Raspberry által használt MIPI CSI interfész sem volt kedvező, itt inkább egy USB csatlakozóval rendelkező modult részesítettünk előnyben. [12] [16]

12. ábra - FIT0892 DFROBOT kameramodul [29]

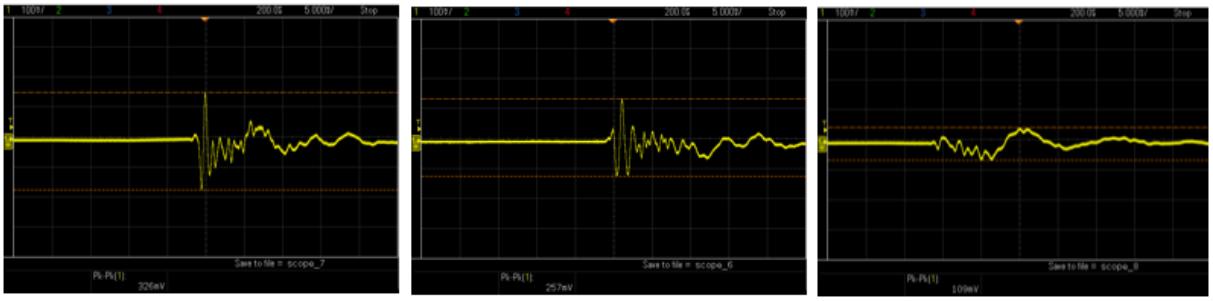
3.2.3 Trigger - piezo

A kép készítését szükséges egy kiváltó eseményhez kötni. A kiváltó eseménynek érdemes a vessző becsapódását választani. A becsapódás érzékelését többféleképpen is lehet érzékelni, lehetséges magukkal a kamerákkal, illetve szenzoros alapon. A kamerákkal megvalósított élőképes figyelőrendszer relatív nagy erőforrást emészt fel és fennáll a téves triggerelés is, az esetlegesen képbe nyúló, nem vizsgálni kívánt objektumok esetében. Mi a szenzoros alapú megoldás mellett döntöttünk. A legoptimálisabb a becsapódás megállapításához egy piezo-mikrofon. Ezek az eszközök olyan piezoelektromos anyagokon (például bárium-titanát) alapulnak, melyek képesek elektromos feszültséget generálni mechanikai nyomás vagy rezgés hatására. A nyílvessző becsapódása erős hang és rezgéshullám jár, emiatt a szenzort a tábla hátuljára vagy oldalára erősítve biztosan érzékelhetjük a becsapódás időpontját. Az általunk használt 20 mm átmérőjű piezo lapka saját méréseink alapján akár 35 V feszültséget is képes leadni kellő erősségű rezgés hatására. A lapka elhelyezkedésétől és a találat helyétől függően a nyílvessző becsapódása 50-180 mV közötti értéket produkál saját méréseink szerint.

13. ábra - 20 mm piezo mikrofon [28]



14. ábra - piezo max. kimeneti feszültsége - oszcilloszkóp ábra [saját ábra]



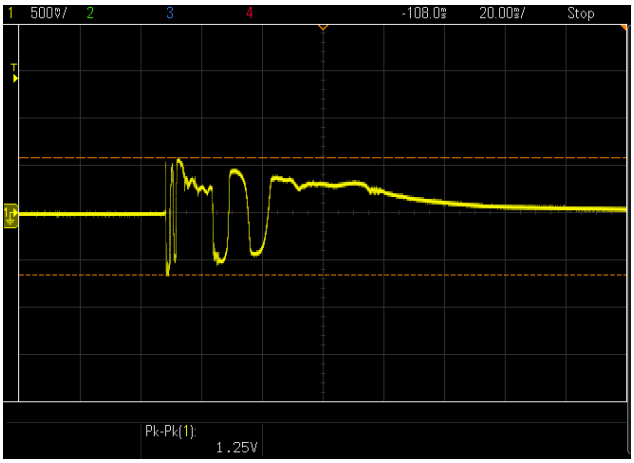
15. ábra - piezo kimeneti feszültség értéke találat esetén - oszcilloszkóp ábra [saját ábra]

A piezo bekötését nehezítette, hogy az általunk használt Raspberry modul nem rendelkezik analóg port-tal. Ennek a problémának a kiküszöbölése egy ADC átalakítóval, vagy különböző kapcsolóüzemű áramkörökkel és komparátorokkal lehetséges. A kapcsolás megvalósítható műveleti erősítő segítségével, azonban a tranzisztorok megléte miatt egy többlépcsős földelt emitterű tranzisztoros kapcsolás mellett döntöttünk.

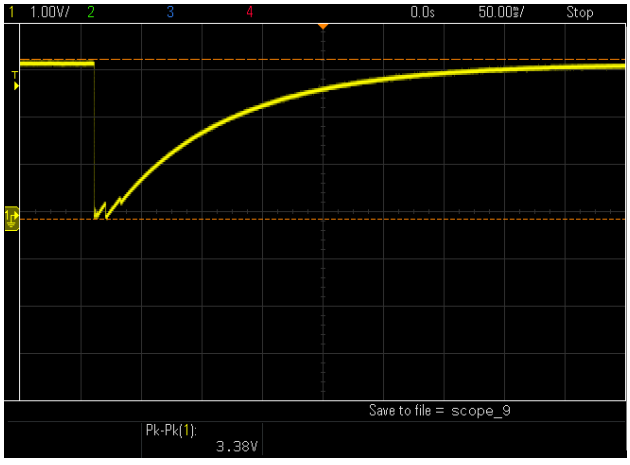
A kapcsoláshoz felhasznált elemek:

- 1db 20 mm piezo mikrofon
- 2db - BC547 NPN tranzisztor
- 1db BC557 TNP tranzisztor
- 4db ellenállás (3db 470 Ohm, 1db 10kOhm)
- 1db elektrolit kondenzátor (10 µF)
- 3db lineáris potenciométer (10k)
- 2 darab 4148-as dióda
- próbapanel, kábelek

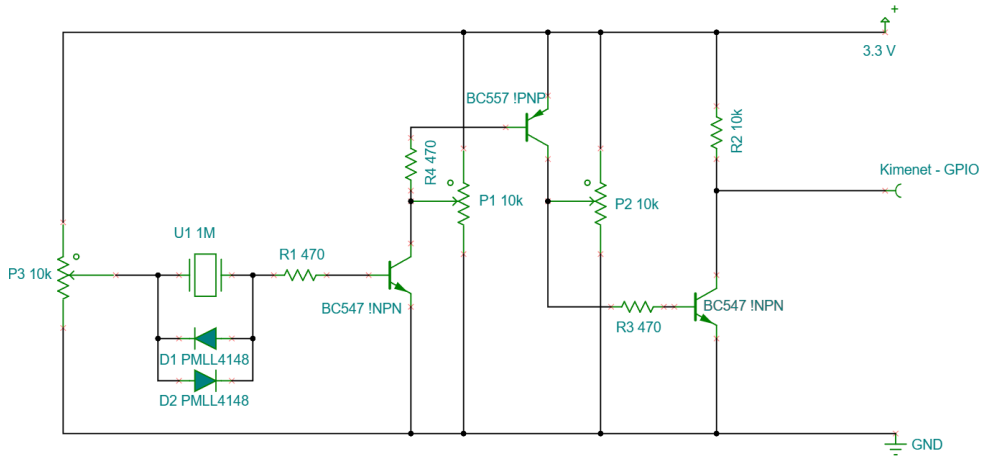
A kapcsolóüzemű tranzisztoros áramkör a Raspberry egyik 3,3 V-os és a 17-es GPIO pin-jét használja. A GPIO port egészen addig HIGH állapoton van amíg a vessző be nem csapódik, ekkor az áramkör a kimenetét a GND-re (logikai LOW-ra) húzza. A szoftver számára ez lesz a trigger esemény. A kapcsolásba került potenciométer segítségével a feszültségosztás értékét állíthatjuk. A feszültségosztó értékének módosításával ráhatásunk van a trigger érzékenységének állítására. A kettő dióda bekötése biztonsági okokat szolgál. A GPIO port már 5 V hatására is károsodhat, ezért volt szükséges egy alul-felül vágó áramkörre, ahol a párhuzamosan bekötött diódákkal 1,25 V-értéknél levágjuk piezo által kibocsájtott értéket, így biztosan nem kerül 3,3 V-nál nagyobb érték a GPIO port-ra. A piezo lapka felhelyezése után véglegesen kalibráltuk az érzékenységet vesszőbecsapódására, ezzel is szűrve téves trigger jeleket. [17] [18]



16. ábra - piezo diódával levágott max. kimeneti feszültsége [saját ábra]



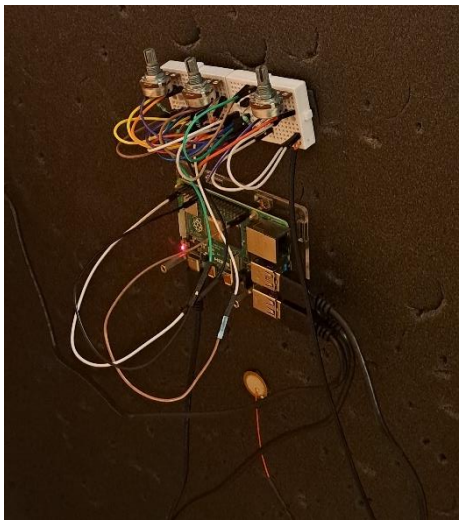
*17. ábra - Áramkör kimeneti feszültség értéke találat esetén [*saját ábra*]*



*18. ábra - Végleges kapcsolás [*saját ábra*]*

3.3 Eszközvédelem

A Raspberry-t egy akril tokban, a tábla hátoldalára rögzítettük. A kameramodulokat USB kábel segítségével csatlakoztattuk hozzá, melyeket szintén a tábla hátoldalán vezettünk el. A táblán épphogy áthaladó lövések - a toknak és a laza kábelvezetésnek köszönhetően - nem tehetnek kárt az eszközeinkben. Azonban, a nem prototípus szintű kivitelezésnél, az eszközeink - beleértve a kamerák védelmét - kulcsfontosságú. A kamerák kivételével elhelyezésük védett dobozban a tábla mellett vagy annak hátoldalán lenne optimális. A kamerák védelmét, elsősorban nem blokkolva, hanem térítőlegesen kellene megoldani, például speciális fém kúpformák segítségével. Ennek segítségével, az esetlegesen a kameramodulra irányuló lövések károkozás nélkül csúsznának le a kúpformáról valamelyik irányba. A táblán esetlegesen áthatoló lövéseknek az erejük csekély, egyedül a csigás-vadászati célokra használt íjak képesek olyan erőhatást kifejteni a vesszőre, hogy az lényeges erővel bírjon a táblán való áthaladása után is.



19. ábra - Vesszőfogó hátoldalán rögzített eszközök [saját ábra]

4. Szoftver

A hardver és a szoftver teljes szétválasztása a mi esetünkben nem volt lehetséges. A hardveres beállítások egy részéhez elképzelhetetlen lett volna a szoftveres segítség mellőzése. A különböző tesztkódok alapján tudtuk például a kamerákat megfelelően pozícionálni. A becsapódás által okozott rezgéseket, stabilitást is csak megfelelő szoftveres segítséggel tudtuk vizsgálni. A Raspberry kezdeti telepítései után először a TeamViewer távolelérés programmal ütköztünk problémába. A Raspberry Pi egy Debian 12 (Bookworm) Linux disztribúció található, ami egy "Linux 6.6.31+rpt-rpi-v8"-as kernelt futtat. A Debian ezen verzióján futó Wayland megjelenítési protokoll egyelőre nem támogatott a TeamViewer által, ezért vissza kellett váltani a régi X11-es megjelenítési protokollra, ahol zökkenőmentesen csatlakozott a program. Programunkat Python 3.11.2-ben fejlesztettük. Csomagjainkat a pip 23.0.1-es verziójú csomagkezelőből szereztük be. A Thonny 4.1.4-1-es verziójú fejlesztői környezetet használtuk a fejlesztés jelentős részére, illetve a tesztelés egészére. Mivel a program képfeldolgozással foglalkozik, így egy részét Windows 11 pro 23H2-alatt Visual Studio Code 1.94.2 és Python 3.12.5-tel fejlesztettük. A program a fejlesztés során párhuzamosan mind a két rendszeren volt tesztelve, a képfeldolgozás szempontjából eltérés nem volt tapasztalható.

A fejlesztéshez az OpenCV 4.6.0 és a NumPy 1.24.2-es csomagot használtuk alapként a képek készítéséhez és feldolgozásához. A könnyebb tesztelhetőség és vizualizáláshoz a tkinter 8.6-os verziójú csomag volt segítségünkre. A GPIO portok eléréséhez, pedig a RPi.GPIO 0.7.1 csomagra volt szükség.

Az OpenCV széleskörben elterjedt és használt könyvtár több ezer optimalizált algoritmust tartalmaz, amelyek kifejezetten képfeldolgozási, objektumfelismerési és egyéb feladatok elvégzésére lett megalkotva. Előnyei közé sorolható a gyorsasága és optimalizáltsága, továbbá a platformfüggetlen használata. Bár eredetileg C++-ban íródott, de Python nyelven is elérhető. A projekt során felhasználtuk az alapmoduloktól (kép betöltése, megjelenítése, mentése) egészen a speciálisabb képfeldolgozási modulokig (Canny éldeketálás, kontúrfelismerés) kínálta lehetőségeket. [15] [19] [20]

A NumPy könyvtára szintén nem ismeretlen a fejlesztők számára. Széleskörű megoldásokat biztosít a numerikus számításokhoz, többdimenziós tömbökhöz és a mátrixok kezelésére. Számunkra elsősorban a binarizált képeken való kereséséhez és a matematikai műveletek elvégzésénél volt rá szükség. [15] [19] [20]

4.1 Tesztkódok

4.1.1 Kamerakezelés

Az első tesztkódok az élő kameraképek megjelenítését célozták meg. A kód első lépéseként a könyvtárak meghívása után a kamerákat inicializálta. A kamerák címének és a megfelelő API (Application Programming Interface) megkeresése után pár függvény segítségével tudtuk olvasni és megjeleníteni az élő kameraképet. Az élőkép a kamerapozícionálást segítette. Ennek befejeztével a képkészítés és annak kielemzése következett. A kódban pár dolgot módosítva láthattuk mennyire torzít az objektív, illetve, hogy vannak-e a különböző tengelyeken elferdülések az általunk kívánt pozíciókhoz képest. Az újabb korrigálások után már kész képekkel dolgozhattunk.

4.1.2 Halszem-effektus

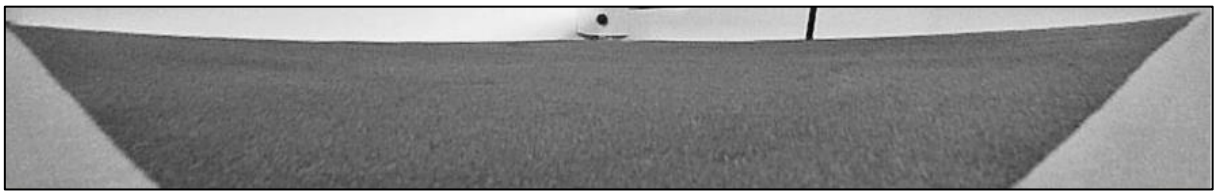
A képeken a vártaknak megfelelően látható volt a kép széle felé jelentkező torzulás (Halszem-effektus). Ennek megfelelő korrigálása kulcsfontosságú a későbbi képlevágások és arányok miatt. Fontos volt, hogy a torzítást ne csupán szemmérték-szerűen korrigáljuk, hanem a valós arányok megmaradjanak a korrigált képen is. A valós arányok elvesztésével nem lehetséges meghatározni a képek alapján a vessző helyzetét a torzult tartományon belül, vagy csak kifejezetten pontatlanul. A korrekcióra lehetőséget ad az OpenCV kamera mátrix paraméteres függvénye. Segítségével a kamera belső paraméterei alapján kezelhetjük és korrigálhatjuk a kamera mátrix torzulását. Elsősorban a radiális torzítást kellett optimalizálni, melyet a függvény kettő szabadon változtatható értékével könnyen megtehettünk. A méretarányok megőrzése végett szükséges a módosított kameramátrixot optimalizálni és a képet aszerint újra korrigálni, erre szintén egy függvény volt segítségünkre.

A kép helyességét egy teszttel ellenőriztük. A teszt során az egyik kamerától egyenlő távolságra félkörívben egyenlő osztásközökre helyeztünk el csavarokat. A készített képen a csavarok távolságának mérésével igazolhattuk, hogy a méretarányokat sikerült megőrizni a teljes spektrumon (a készített félkör teljes ívén). Nyilvánvalóan valamennyi pontatlanság keletkezik a készített új mátrixon is, ez azonban elhanyagolható (néhány pixel nagyságrendű). [9]

20. ábra - Képkorrekció (előtte-utána) [saját ábra]

4.1.3 Méretezés

A torzításmentes képet ezután méretre kellett vágni, hogy csupán a vizsgálni kívánt tartományt lássuk. Ezt szintén manuálisan, tesztek alapján állítottuk be, azonban a későbbiekben automatizálható ez a folyamat is. A cél az volt, hogy a kép jobb és bal oldala a tábla két szélénél legyen levágva, így a későbbiekben könnyebb a valóságos mértékekre való átváltás, illetve a tábla szinte teljes síkját tudjuk a találatokat regisztrálni. A vizsgálandó ablak alsó és felső szélét szintén a tábla teljes síkjára igazítottuk. A felső vágás a hátteret kitakaró keret tetejéhez lett pozícionálva. Az így körülvágott kép megköveteli a kamerák biztos rögzítését, mivel pozícióváltozás esetén újra kell kalibrálni a vizsgálandó ablakot. A megfelelő rögzítés következtében az elmozdulással a tesztek során nem volt problémánk. Az így kapott végleges levágott fotó már csak a kívánt területet mutatta és csupán néhány változtatásra volt szükség ahhoz, hogy a detektálás gondtalanul történjen. A következő lépés a kép szürkeárnylatossá alakítása volt. Ez leegyszerűsíti a kezelt kép színkészletét, így könnyebben lehet a kontrasztokat elkülöníteni. A meghívott könyvtár itt is szolgál egy ilyen célra megírt függvénnyel. A kapott szürkeárnyalatos képpel meg lehetet kezdeni a nyílvessző detektálását a képen.



21. ábra – Előkészített kép [saját ábra]

4.2 Detektálás

A detektálás módszertana kapcsán már több lehetséges megoldást kellett mérlegelni. Elsőként mindenféleképp el kell különíteni a vesszőt a vesszőfogótól. Ehhez azonban először

meg kell találnunk magát a vesszőt a fotón. Több lehetőség közül is válogathatunk, hogy miként szeretnénk ezt megtenni.

4.2.1 Objektum detektálás

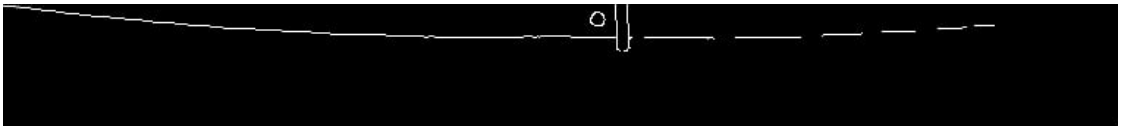
Az első felvetődött lehetőség a konkrét objektumdetektálás használata volt. Ennél a lehetőségnél felhasználhatunk olyan nyíltforráskódú könyvtárakat, amikben rengeteg olyan alapvető tárgy van eltárolva, amiket a könyvtár segítségével a szoftver képes azonosítani. Az egyik legnépszerűbb ilyen könyvtár a „Coco database" (Common Objects in Context). A könyvtár a Microsoft fejlesztésével, sok közreműködővel alkotott meg egy olyan bárki számára elérhető adatbázist, mely rengeteg hétköznapi tárgy határolódoboz kimenetét vagy az objektumszegmentációs elkülönítését segíti. A könyvtár felhasználásával a vesszőt detektálhatjuk és létrehozhatjuk azt a szoftveres határolódobozát mely alapján megkereshetjük a becsapódási pontját (legalsó pont). Bár a könyvtár által használt komplex függvényekre és kódolásra kevésbé lehet rálátni, használata kifejezetten felhasználóbarát, célja, hogy előre mozdítsa az objektumdetektálást. [21] [22]

4.2.2 Canny éldetektálás

Egy másik lehetséges módszer a különböző OpenCV-ben is megtalálható éldetektáló algoritmusok. Az egyik leggyakrabban használt, számunkra is elérhető metódus a Canny éldetektálás. Ez az éldetektálási algoritmus egy összetett metódus. Megkeresi azokat a vonalakat (potenciális éleket), ahol a színek vagy intenzitások változása hirtelen történik, így jelölve ki az objektumok kontúrjait. A logika magába foglalja a nyers kép előkészítését (konvulóció, Gauss zaj szűrés), a gradiensek számítását, a kapott élek nem-maximum elnyomását (Non-max Suppressio), majd hiszterézises küszöbölését. Eredményképpen a megadott hiszterézises küszöböléssel állíthatjuk, hogy mennyire finom élek látszódjanak a módosított képen. [23] [24]

Az általunk előkészített képen végigfuttatott Canny-detektálás - kellő korrekció után - megtalálta a táblába állított vessző kontúrjait. Azonban a nyílvessző fekete és a tábla sötétszürke színe miatt nehéz volt úgy beállítani a küszöbértékeket, hogy a kapott kontúr megfelelő részletességgel látszódjon, azonban a környezet zajai ne legyenek zavaróak. A különböző paraméterek és vesszőpozíciók tesztelése után a nem kívánt detektált élekből csupán a tábla szélének és a szemközti kamera objektívjának körvonala maradt. Az így kapott kép sem volt tökéletes számunkra. A hosszas paraméterezések és a fényviszonyokra való érzékeny viselkedése miatt úgy döntöttünk, hogy megpróbálkozunk más éldetektáló módszerrel is. A cél

a vesszőnek a lehető legjobb elkülönítése volt a háttértől, így másik megoldási metódussal is próbálkozni kezdtünk.

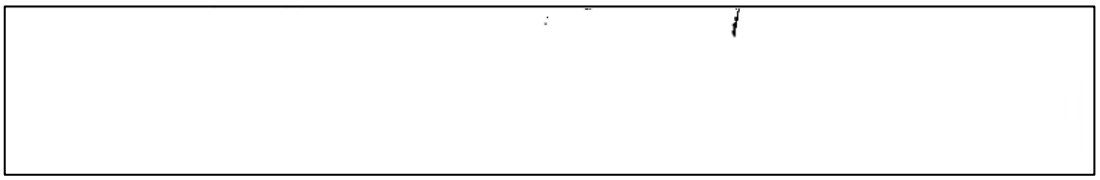


*22. ábra - Canny éldetektálás alkalmazása [*saját ábra*]*

4.2.3 Éldetektálás mellőzése

A harmadik lehetőség, az él és objektum detektáló algoritmusok nélküli elkülönítés volt. Ebben az esetben elsősorban nem a vessző megtalálása, majd kiemelése volt a cél. A logika szerint az előkészített, levágott és szürkearányos képet olyan küszöbértékekkel alakítjuk bináris képpé, mely számunkra kedvező, szinte csak a vessző szerepel rajta. A kép binárissá alakítása küszöböléssel (treshold) történik. Célja, hogy a képen csak fekete és fehér pixelek legyen. A bináris kép vizsgálata még egyszerűbb, hiszen innentől nem kell a küszöbértékek beállításával foglalkozni. A megfelelő küszöbértéket ebben az esetben is úgy kellett megválasztani, hogy a céltárgy jól kivehető legyen, azonban a környezeti zajok ne akadályozzák a detekciót. A küszöbérték megfelelő beállítása után olyan bináris képet kapunk, ahol a vesszőt és néhány környezeti zajt fekete pixelek jelölnek a fehér alapon. A kapott képen látható vesszőnek a legalsó (becsapódási) pontjának az X_{pixel} koordinátájának meghatározása a feladat. Ehhez először listázzuk a fekete pixeleket. A pixelek listázását a np.where függvény segítségével történik, mely megkeresi a megadott értékű (jelen esetben fehér - 255) pixeleket. A kapott értékeket tömbben tároljuk, majd a listázott pixelek közöl megkeressük azt a legutolsó pixelsort (a kép tetejétől nézve) amelyik még tartalmaz fehér pixelt. A legalsó sort a legmagasabb Y_{binpx} koordinátájú fehér pixel fogja meghatározni. A vizsgált legutolsó sorban szereplő pixelek középértékét véve megkaphatjuk az általunk keresett X koordinátát, tehát a vessző becsapódási középpontját. A pontosítás érdekében érdemes nem csupán egyetlen pixelsávot vizsgálni, hanem a megtalált utolsó pixelsor feletti 3 pixelsornyi tartományra kibővíteni az ablakot. A három sorban talált fehér pixelek középértékének koordinátája pontosabb értéket adhat, amennyiben a vessző becsapódási pontja nem teljes a binarizált képen. Ez a módszer nagy pontossággal, relatív egyszerűen működik, azonban csak ideális esetben (amennyiben csak a vessző látható a képen) kaphatjuk meg a nyíl becsapódási pontjának X_{pixel} koordinátáját. Abban az esetben, ha más nem kívánt fehér pixel (zaj) is megtalálható a becsapódási pont alatt vagy azzal egy vonalban, téves helyre lokalizálhatjuk a nyílvesszőt. Továbbá ennél a módszernél is zavaróak a tábla oldalvonalai, hiszen nem kívánatos zajként jelennek meg a képen. Az oldalvonalak eltüntetését fizikailag oldottuk meg. Egy világos színű ragasztószalaggal

ragasztottuk le a tábla perem és a kameratartó keret közti mezsgyét, így az oldalsó élek könnyen a megadott küszöbértéken kívülre kerültek. A tábla felső horizontvonala, illetve adott esetben a szemközti kamera objektíve látszódni fog, azonban a vesszőnek mindenféleképp ezek alatt kell elhelyezkednie, így nem kell őket figyelembe venni. Az esetleges zajok viszont nem hagyhatók figyelmen kívül. A régebbi találatok helyei, nagyobb sérülések a táblán mind megjelenhetnek a bináris kép azon tartományában, ami torzítja a vessző lokalizálását. [25]



*23. ábra - Binarizált kép [*saját ábra*]*

4.3 Környezeti zajok kezelése

4.3.1 Konvolúciós élsimítás

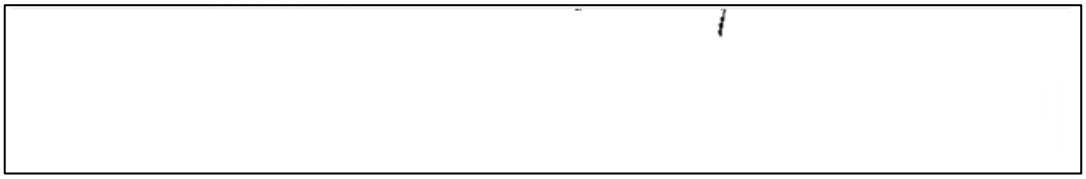
A probléma megoldását a zajok eltávolítása jelenti. Eltávolításuk, az egyik lehetséges módszer a konvolúciós élsimítás, melynek segítségével elhomályosíthatjuk a kép egészét vagy bizonyos részeit, ezzel elérve, hogy a kisebb hibák, zajok ne legyenek kellően kontrasztosak. Ezt a műveletet még a kép binarizálása előtt kell végrehajtani. A felhasznált könyvtárak bizonyos függvényei (np.ones, cv2.filter2D) lehetőséget adnak ilyen képsimításra. A konvolúciós simítás egy adott kernelt (szűrő mátrixot) használ, melyet az egész képen végig léptetünk és a mátrixban határolt elemeket átlagoljuk, így homályosítva a képet és annak éleit. A homályosítással csupán az erős élek (a nyílvessző) maradnak elég kontrasztosak ahhoz, hogy a megadott küszöbértéken belül maradjanak. A homályosítás értékének megválasztása körülményes, mivel a túlzó simítás a távoli találatok pontosságának rovására mehet. [23] [26]

4.3.2 Képek differenciálképzése

Egy másik lehetőség kihasználni, hogy ezek a hibák fix ponton találhatóak és ez alapján kiszűrni őket. Amennyiben készítünk egy nulladik képet, melyen még nem szerepel az adott vessző, a binarizálás után megkapjuk azokat a zajokat, amelyek a vesszőn kívül észlelhetők. Amennyiben ezt a képet összevetjük azzal a bináris képpel, melyen már szerepel a vessző, majd a két kép különbségeket nézzük, akkor tisztán vessző marad az összevetett képen. Ezzel a módszerrel teljesen mértékben szűrhetjük a fix zavaró tényezőket a képről. Ahhoz, hogy ezt a logikát megvalósítsuk, szükséges egy aktuális nulladik kép, ami nem számottevően régebbi a becsapódási fotónál. A fényviszonyok változása miatt nem lehetséges egy előre elkészített mindenkori nulladik képhez viszonyítani. Ahhoz, hogy ezt a viszonyítási képet folyamatosan

frissíteni tudjuk szükség volt egy olyan ciklusra, ami adott időközönként (30 másodperc) felülírja az elmentett nulladik képet. Abban az esetben, ha találat érkezik, ez a ciklus megtörik és a legutolsó állapotú fotóhoz tudjuk hasonlítani a vesszőt tartalmazó másik bináris képet. Az OpenCV (cv2.absdiff) függvénye lehetőséget ad, hogy a két kép különbségeit jelenítsük meg. Ezt egy a pixelekkel végzet XOR logikai művelettel érhetjük el. A logikai kizáró vagy (XOR) ebben az esetben csak azokat a pixeleket tartja meg amik csak az egyik képen jelennek meg, így a mindkettő képen ugyanott megjelenő fix hibákat, zajokat kiszűri. A módszer megköveteli a tábla és a kamerák tökéletes, fix pozícióját, ezzel azonban a tesztek során nem volt problémánk.

A két módszer elméletben együttesen is alkalmazható, hiszen a konvolúcióval módosított kép binarizálása után összevethető egy nulladik képpel. A kettős használatot egyelőre nem teszteltük. A végleges kódban a nulladik képes módszert használtuk. Döntésünket a lehetséges továbbfejlesztési lehetőségekre alapoztuk. Több vessző helyzetmeghatározása esetén a képösszehasonlítás módszerével, elméletben, kevés módosítással a most használt vesszőlokalizációs logika alkalmazható lehetne. Az aktuális találatról készült képeket „logikai kizáró vagy”-gyal összevetve az azt megelőző találat képeivel, mindig csupán egy vessző helyzetét lenne szükséges vizsgálni. A több vesszőre alkalmazhatóságról egyelőre csak elméletben tudunk beszámolni. [7]



*24. ábra - Zajmentesített (kivont) binarizált kép [*saját ábra*]*

4.4 A végleges kód

A kód átláthatósága és tesztelése érdekében a különböző kódrészleteket külön írtuk meg. A szoftver elkészítése közben, - ahogy a dokumentáció is taglalja - számtalanszor kellett kitekintéseket tenni és különböző módszereket tesztelni. A tesztkódok és részkódok egy része teljesen kikerült a szoftverből, míg más kódrészletek részlegesen vagy módosításokkal kerültek bele. A végleges kód összeépítéséhez a kész kódrészletek összeépítése mellett szükség volt egy őket összefogó keretre. A kód összeépítése során egy grafikus felhasználói felületre (GUI) építettük fel a programot, a könnyebb hibakeresés és átláthatóság javítása érdekében. A program futtatásakor a grafikus felületen gombokkal hajthatjuk végre a különböző lépéseket. Az automatikus működés elindításához egy *„Live detection”* gombot hoztunk létre, amely

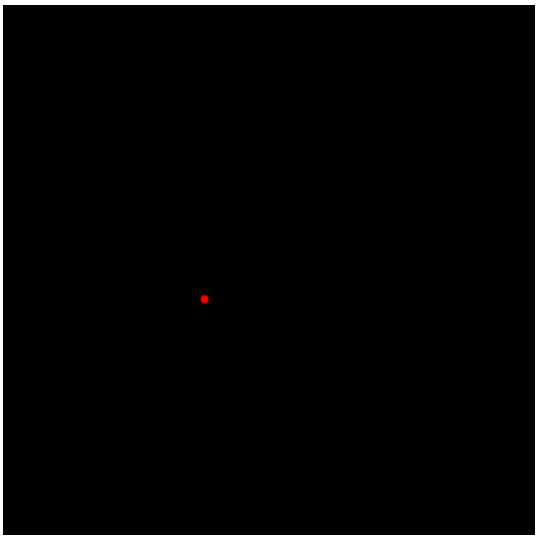
megnyomására a program automatikusan képes felismerni mikor érkezett találat és detektálni annak helyzetét. A képek elkészítése előtt egy platformcsomag ellenőrzi, hogy Linux rendszeren futtatjuk-e a metódust, ha nem, akkor - jelezve a felhasználónak - nem fut le a képkészítő programrész.

A program futásához szükséges inicializálnunk a GPIO port-ot, hogy figyelni tudjuk a trigger-jel beérkezését. A *„Live detection”* kettő különböző folyamatot indít el, ahol mind a két helyen egy-egy ciklusba fut bele a program párhuzamosan. Erre azért volt szükség, hogy egyidőben tudjunk friss nulladik képet készíteni és figyelni a trigger-jel GPIO port-jának állapotát. A mellékszálon 30 másodpercenként készül új kép, kellően aktualizálva a binarizáláshoz használt nulladik fotót. A találat érkezésekor a GPIO17-es port értéke HIGH állapotról LOW-ra változik és kellő várakozással – ezzel elkerülve a futás béli hibákat – átugrik a detektáló programszálra. Mindkettő kamera fotóz, a készült képeket változókba mentjük. A két változóban levő képre alkalmazzuk a torzítások korrigálására szolgáló kameramátrixot, majd .jpg (undistorted1_save.jpg/undistorted2_save.jpg néven) formátumban mentésre kerülnek a szoftver saját mappájába. A képek előkészítésénél a készített vesszőt tartalmazó képek mellett a nulladik fotó előkészítése is megtörténik. A binarizálást olyan függvény végzi, ami paraméterként várja a két kép elérési útját, valamint a függőleges és vízszintes levágásokhoz szükséges adatokat. Az elkészült kettő binarizált fotóhoz tartozik kettő darab nulladik fotó is (_OUT), melyeket a vesszőt tartalmazó képekből (_IN) kivonja az OpenCV absdiff függvénye. A függvény kétszer van meghívva, így először az egyik kamera, majd a másik képét dolgozza fel és adja tovább. A visszatérési értékként továbbadott binarizált képen megkeressük a legalacsonyabban elhelyezkedő fehér pixeleket (az utolsó 3 sort figyelve), amely a vessző becsapódási pontját jelzi. A megkapott $X_{pixelcam}$ koordinátája és a kép szélessége továbbadódik a szükséges matematikai számítások elvégzéshez. A matematikai számításokhoz szükség van az átadott paramétereken kívül a két kamera közötti távolságra, illetve a kamerák betekintési szögeire a levágott képek alapján. A nem átadott adatokat fizikai mérések alapján, manuálisan adtuk meg. A látószöget elosztjuk a képszélességgel, ezzel megkapva egy pixel értékét fokban kifejezve. A kapott $X_{pixelcam}$ koordinátával beszorozva az előzőleg kapott értéket, megkaphatjuk az irányvektor tábla szélével bezárt szögét. A meredekség felírásához szükséges a kapott szög (radiánban kifejezett) értékének a tangensét venni. A kapott meredekség és az ismert kamerapozíciók függvényében meghatározhatjuk az egyenesek egyenletét. Az egyenesekre felírt egyenlőséget megoldva megkapjuk a metszéspont X_{pixel} koordinátáját a jobb felső origónak választott kameraképhez képest. Az Y_{pixel} meghatározásához

visszahelyettesítünk az egyik egyenes egyenletébe a kapott X_{pixel} értékkel. A megkapott koordináták a vessző helyét jelölik a tábla síkjában. A valós mértékegységekre való átváltáshoz szükségünk van a képeken 1 pixelnek megfelelő valós centiméter értékre. Ehhez egy előre meghatározott szakasz (a tábla átlója) valós hosszának, a képen látott pixelszámával való leosztása szükséges. [27]

A kiszámolt helyzetet megjelenítjük egy fekete vásznon egy piros ponttal, ami reprezentálja a nyílvessző helyzetét a vesszőfogóban. A program a gombok elhelyezésére a grides módszert használja, melyet a tkinter biztosít. Továbbá a valóságos koordinátákat, cm-ben kiíratjuk.

25. - Végleges kód [melléklet]



*26. ábra - Megjelenített vesszőpozíció [*saját ábra*]*

5. Tesztelés

A tesztelés során a végleges program lefutására, a kapott eredmények pontosságára és a felmerülő problémákra voltunk kíváncsiak. A tesztet két részre bontottuk, az első felében a pontosságot vizsgáltuk, a második felében az egyéb behatások ráhatását az eredményekre.

5.1 Szoftveres lefutás

A szoftveres rész befejeztével lehetőségünk adódott tesztelni a *„Live detection”* funkcióját a programnak. Ez a funkciógomb tekinthető a célhoz legközelebb álló működésnek a projekt mostani állapotában. A teszt során elindítottuk a funkciót, majd megvártuk, hogy sikeresen készítsen legalább kettő nulladik képet, mielőtt lövést adtunk le rá. A képek ellenőrzése után, a táblára leadott lövés aktiválta a detektálás lefutását és sikeresen megkaptuk a vessző pozícióját.

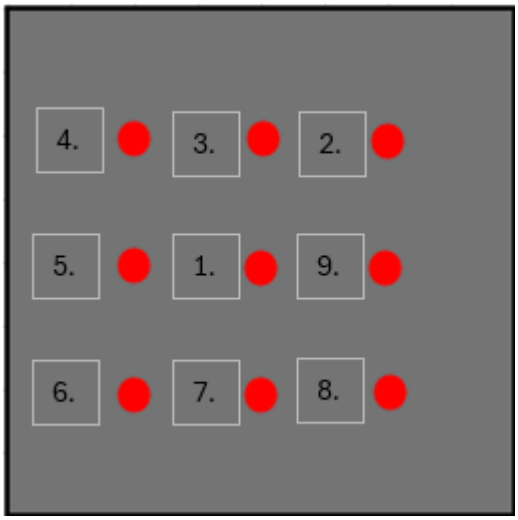
A tesztet ezután még háromszor megismételtük, különböző szögekből, a tábla más-más részeire leadott lövésekkel. Minden esetben sikeresen, a kívánt módon futott le a program.

A reprodukálhatóságot a végén a tábla középpontjába kézileg beállított vesszővel végeztük. A pontosan ugyanott levő vesszőre, háromszor ismételtük meg a detektációt, egy hamis triggerelés segítségével. A kapott koordináták a mindössze pár pixeles eltérést mutattak, így ez a teszt is sikeresnek tekinthető.

5.2 Pontosság

A pontosság tökéletes meghatározása érdekében, rengeteg leadott találatot kéne kielemezni a táblának a teljes felületén. Az idő hiánya miatt, egy egyszerűsített tesztet használtunk, miszerint a táblára 9 darab szimmetrikus alakzatban elhelyezett mérési pontokat rögzítettünk. Sorra minden mérési ponton detektáltuk a vessző valós koordinátáit, majd összevetettük a fizikai mérések eredményével. Az eredményeket Excel-es környezetben rögzítettük a jobb kiértékelhetőség miatt.

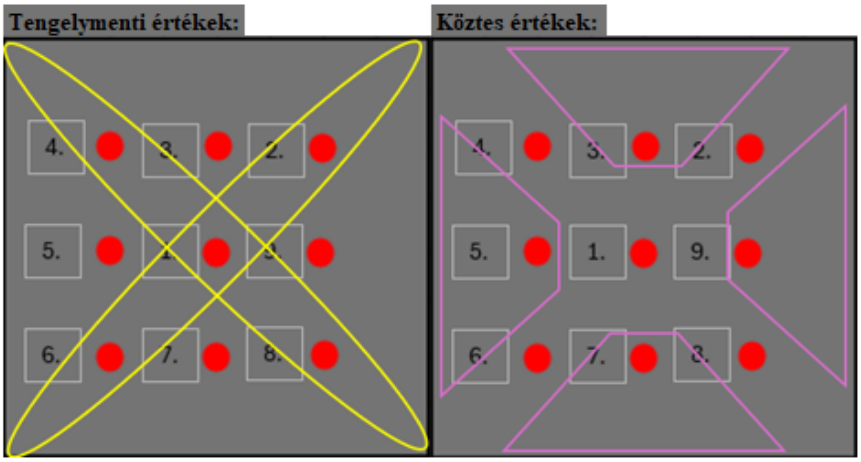
27 - Teszt esetek adatai - Excel fájl [melléklet]



*28. ábra – Teszt - mérési pontok [*saját ábra*]*

Az eredmények kiértékelése után azt a következtetést lehetett levonni, hogy a projekt kezdetén megszabott célpontosságot nem sikerült elérni. A kamerák látószögének felezőtengelyén elhelyezkedő találatok kifejezetten pontosak, itt 0,5-2 cm-es eltérések fordulnak elő, ritkábban egy-egy kiugró (max. 3 cm-es) értékkel. Az ilyen mértékű pontatlanság, bár kívül esik némelyik esetben a meghatározott céltól (1,2 cm), elfogadhatónak mondhatók egy hobbi sportolói felhasználásra. A tengelyeken kívül eső találatok viszont a fent említett eltérések többszörösét mutatják. Itt az átlag hibaérték 5-10 cm-es határok között mozog.

A mérés során a tengelymenti adatok eltérésének átlaga 1,64 cm, míg a köztes részeké 7,43 cm. A köztes részek pontatlansága már amatőr szintű felhasználás során sem megfelelő.



*29. ábra - Teszt - pontossági zónák [*saját ábra*]*

A pontosságromlás területekre való levetítéséből egyértelműen arra következtethetünk, hogy a kamerák torzítása nem megfelelően lett eltávolítva. A kamerák látószögének szélei felé a torzulások miatt nem megfelelő X_{pixel} értékeket kapunk. A hibás értékkel való tovább számítások, illetve a másik kamerának a hibáival a végérték többszörösen is ferdül a helyes értékhez képest. A hiba orvoslására, a kamerák torzítási értékeinek pontosabb meghatározása adna megoldást. A megfelelő torzítási értékekkel és korrekciójával a köztes területeken is feltételezhetően a tengelymentihez hasonló eltérési értékek lennének. A kamerák megfelelő korrigálásához sakktáblás paramétermeghatározás, majd aszerint gyártott új kameramátrix lenne szükséges.

6. Fejlesztési lehetőségek, potenciál

A fejlesztési lehetőségek bemutatásánál sorra vettük a végleges – nem prototípus szintű – megvalósításnál szükséges javításokat, átalakításokat, illetve a további funkcióbővítési és fejlesztési lehetőségeket. Továbbá szerettük volna bemutatni egy véglegesített eszköz lehetséges integrációját más edzéssegítő rendszerek hardveres és szoftveres környezetében.

6.1 Hardveres fejlesztések

Ahogy a dolgozat elején is kifejtettük, a projektötlet komplex, a dolgozat csak az egyik szegmensének a bemutatásával foglalkozik. A továbbfejlesztési lehetőségek az alapötlet befejezésével kezdődnek, amelyek magában foglalják 5-6 nyílvessző helyzetmeghatározását, helyzetének tárolását és a hasznos statisztikai adatok kiszámítását. A végleges hardveres

megvalósításkor számos helyen lehet korrigálni a pontosság növelése érdekében. Ilyen például a teljesen megegyező kameraállások (elhelyezkedés, szögek stb.), a jobb anyaghasználat az elmozdulások elkerülése végett és a műszerek megfelelő pozícionálása. A kamerák esetleges fejlesztésével szintén nagyobb pontosság érhető el. Elhelyezhető kezelői panel (start/stop gomb), állapot jelző LED is a tábla keretén, ezekkel is felhasználóbarátabbá téve az eszközt. Új szenzorok alkalmazásával (például szélerősség mérés, fényerősség mérés, erőmérés) új funkciókat, adatokat lehetne a vesszőfogóhoz adni.

6.2 Szoftveres fejlesztések

A szoftveres fejlesztési lehetőségekhez sorolható a kamerák kalibrálásának automatizálása, a fényviszony változások kezelésének automatizálása (hardveres változtatások is szükségesek), továbbá a különböző lőlapok feltűzési lehetőségének biztosítása a táblára. A felhasználói felület kidolgozásával, weblap vagy mobilapp fejlesztésével kényelmesebbé és edzés közben is ellenőrizhetővé válnának a feldolgozott adatok. A felhasználói profilhoz kötött használattal visszamenőleg is követhető lenne az edzések intenzitása és hatékonysága. A hosszútávú statisztikákból fejlődési utat, segédletet lehetne készíteni, de az esetleges íjászezközök cserélésének, fejlesztésének hatása is látható lenne.

6.3 Fejlesztési potenciál

Amennyiben más külső eszközök összekapcsolása is megoldható lenne a használt felülettel (pl.: mobilapp), a találati adatokhoz más hasznos információk is köthetőek lennének. Dinamikus íjászat során például - az okosóránk adataival – hasznos külső adatokat lehetne bevonni az edzés monitorozásába, ilyen lehet a pulzusszám, sebesség, giroszkópos mozgásérzékelés. Az esetleges speciális elektronikai íjász kiegészítőkkel, szenzorokkal tovább növelhető lenne az edzések effektivitása. Ilyenek lehetnek például az íjra helyezhető segédeszközök melyekkel íjpozíciót, elmozdulást vagy épp céltáblától való távolságot mérhetünk. Szintén ide sorolhatóak a testtartást, kézmozdulatokat figyelő eszközök is.

Összegzés

A projekt sikerességének megítélése vitatható. Bár a megcélzott pontosságot nem sikerült elérni, azonban a hiba forrásának megtalálásával a későbbiekben orvosolhatóvá vált. Amennyiben a nagyobb eltérésekért okolható nem megfelelő kamera torzulási korrekciót nem vesszük figyelembe, a metódus pontosnak tűnik. A számítási és logikai műveletek helyesnek tűnnek, bizonyítja ezt a kamerák tengelyén való találatok meghatározásának pontossága. Arra következtethetünk, hogy a megfelelő módosításokkal elérhető a célpontosság is. Nyilvánvalóan ezt biztosan csak a módosítások után elvégzett megfelelő mérésekkel, tesztekkel igazolhatjuk.

A szoftveres pontosításokat és továbbfejlesztéseket a torzítási tényezők megfelelő korrekciója után lehet elvégezni. A pontossági hibahatárt nem számítva a projekt hardveresen megfelelőnek tűnik, a rögzítések és a választott anyagminőségek eleget tesznek a prototípus szintű megvalósításnak. A projekt továbbfejlesztése lehetséges és célszerű lehet. A jelenlegi eredmények szerint a fejlesztésekkel elérhető a projekt végleges célja is, hogy egy hobbi szinten használható és hiánypótló koncepcióvá váljon.

Irodalomjegyzék

[1] „lovasijaszat,” Kassai Kft., [Online]. Available: https://lovasijaszat.hu/termekek.

[2] C. Wells, „Automatic camera arrow spotting makes Olympic debut in Tokyo,” *World Archery,* Szeptember 2021.

[3] „dimensions,” [Online]. Available: https://www.dimensions.com/element/fita-recurve-target-olympic.

[4] RyngDyng, „ARCHERY ANALYTICS,” RyngDyng, [Online]. Available: https://shop.archery-electronics.com/detail/8aa96ad144114884a666aa14ee9dbbd5.

[5] RyngDyng, „ARCHERY ANALYTICS,” RyngDyng, [Online]. Available: https://www.archery-electronics.com/en/public/ryngdyng/products.

[6] J. B. Bowman, „infoarchery,” Október 2023. [Online]. Available: https://infoarchery.com/what-are-archery-targets-made-of/.

[7] K. H. L. D. H. C. J. T. S. P. Basil Andy Lease, „Two-Stage Computer Vision Assisted Automatic Archery Scoreboard Scoring System,” *ieeexprole,* 2023.

[8] Y. Jeong, H. Lee, J. Yu és S. Kim , „Measurement and Calibration System of Arrow’s Impact Point,” 2013.

[9] K. Sadekar, „Understanding Lens Distortion,” 2020.

[10] A. f. Beginners, „The Complete Guide To Arrow Speed,” 2021. [Online]. Available: https://archeryforbeginners.com/blog/arrow-speed/.

[11] „Íjászbarát - Jelmagyarázat,” [Online]. Available: https://www.ijaszbarat.hu/jelmagyarazat.

[12] R. P. (. Ltd., *Raspberry Pi 4 Model B DATASHEET,* 2024.

[13] J. R. T. A. M. N. A. W. M. M. J. A. H. Kahlil Muchtar, „A Low-cost Raspberry Pi and Deep Learning System for Customer Analysis,” 2023.

[14] S. R. H. S. B. A. T. S. Azfar Isa Mahmood, „Real-time Vehicle monitoring for traffic surveillance and adaptive change detection using Raspberry Pi camera module,” 2017.

[15] „TeamViewer,” [Online]. Available: https://www.teamviewer.com/hu/products/remote/solutions/remote-work/.

[16] *FIT0892 Datasheet,* 2023.

[17] „tudasbazis.sulinet,” [Online]. Available: https://tudasbazis.sulinet.hu/hu/0d0cc85d-f7b5-41fb-aec0-d1b8362c7ebf_e90c4562-46d5-4b3a-a5ed-d640f67b512a_102e5c57-b50f-4053-b4a3-c1a7c5aa564f_1fa287d4-8ce1-4afa-8f49-8df586214075_d39de7f3-eddc-4727-913c-3d5dda4be5fc_d6dbdc94-6a12-46da-843f-d1c2edceb6b4.

[18] „skrelectronicslab,” [Online]. Available: https://skrelectronicslab.com/how-to-make-clap-switch-using-transistor.

[19] „NumPy,” [Online]. Available: https://numpy.org/doc/stable/.

[20] D. A. T. P. H. D. M. C. Ivan Culjak, „A brief introduction to OpenCV,” 2012.

[21] „cocodatabase,” 2020. [Online]. Available: https://cocodataset.org/#detection-2020.

[22] M. M. S. B. L. B. R. G. J. H. P. P. D. R. C. L. Z. P. D. Tsung-Yi Lin, „Microsoft COCO: Common Objects in Context,” 2015.

[23] C. Benedek és T. Szirányi, Szerzők, *Képfeldolgozás - Konvolúció, élkereső eljárások.* [Performance]. 2008.

[24] H. L. Juan Yan, „Window frame obstacle edge detection based on improved Canny operator,” 2019.

[25] H. W. S. A. T. S. I. O. T. T. Z. Hiroshi Sasano, „ieeexplore,” Beijing, China, 2013.

[26] E. W. Weisstein, „MathWorld - Convolution,” [Online]. Available: https://mathworld.wolfram.com/Convolution.html.

[27] „KOORDINÁTAGEOMETRIA,” [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://ptlklszl.web.elte.hu/misc/material/koordgeoD.pdf.

[28] „Hestore,” [Online]. Available: https://www.hestore.hu/prod_10041994.html?gross_price_view=1&source=gads&lang=hu&gad_source=1&gclid=Cj0KCQiAoae5BhCNARIsADVLzZd-kpFLXMurHxopa67RN69pQpx1ZUimdMtzjtAhPkrYcapNL3iNUAwaAhuREALw_wcB.

[29] TME. [Online]. Available: https://www.tme.eu/en/details/df-fit0892/camera-modules/dfrobot/fit0892/.

Mellékletek

1. ábra - Képfeldolgozással megvalósított vessződetektálás [7] *8*
2. ábra - Lézeres módszertannal megvalósított vessződetektálás [8] *9*
3. ábra - Saját módszertan logikai folyamatábrája [saját ábra] *10*
4. ábra - Kezdeti elrendezési terv [saját ábra] *5. ábra - Végleges kamera elrendezés [saját ábra]* *12*
6. ábra - Felhasznált vesszőfogó [1] *14*
7. ábra – Talapzat [saját ábra] *14*
8. ábra - Kamera rögzítő modul [saját ábra] *15*
9. ábra - Pozícionált kamera látképe [saját ábra] *16*
10. ábra - Hardveres elemek összképe [saját ábra] *16*
11. ábra - Raspberry Pi 4B [12] *17*
12. ábra - FIT0892 DFROBOT kameramodul [29] *18*
13. ábra - 20 mm piezo mikrofon [28] *18*
14. ábra - piezo max. kimeneti feszültsége - oszcilloszkóp ábra [saját ábra] *19*
15. ábra - piezo kimeneti feszültség értéke találat esetén - oszcilloszkóp ábra [saját ábra] *19*
16. ábra - piezo diódával levágott max. kimeneti feszültsége [saját ábra] *20*
17. ábra - Áramkör kimeneti feszültség értéke találat esetén [saját ábra] *20*
18. ábra - Végleges kapcsolás [saját ábra] *21*
19. ábra - Vesszőfogó hátoldalán rögzített eszközök [saját ábra] *21*
20. ábra - Képkorrekció (előtte-utána) [saját ábra] *24*
21. ábra – Előkészített kép [saját ábra] *24*
22. ábra - Canny éldetektálás alkalmazása [saját ábra] *26*
23. ábra - Binarizált kép [saját ábra] *27*
24. ábra - Zajmentesített (kivont) binarizált kép [saját ábra] *28*
25. - Végleges kód [melléklet] *30*
https://drive.google.com/drive/folders/1G5qWNTjpxecarEp1ltyNCgJLQSLuiNC3?usp=sharing
26. ábra - Megjelenített vesszőpozíció [saját ábra] *30*
27 - Teszt esetek adatai - Excel fájl [melléklet] *31*
https://drive.google.com/drive/folders/1G5qWNTjpxecarEp1ltyNCgJLQSLuiNC3?usp=sharing
28. ábra – Teszt - mérési pontok [saját ábra] *31*
29. ábra - Teszt - pontossági zónák [saját ábra] *32*