

3 dimenziós tér érzékelése kontraszt és frekvencia
alapú képfeldolgozási módszerekkel, pályabecslés
gépi tanulás felhasználásával

Sebők-Tornai Bence
Témavezető: Dr. Várady Géza



Pécsi Tudományegyetem
Műszaki és Informatikai Kar

Biomedical Engineering MSc

2022 novemeber

Tartalomjegyzék

1. Bevezetés	3
2. Problémafelvetés	5
2.1. A technika fejlődése	5
2.2. A fejlődés kihasználása, igények	7
3. Irodalom	9
4. Anyag és módszer - általánosságban	11
5. Eredeti megközelítés hardveres és szoftveres alapjai	13
5.1. Raspberry miniPC	14
5.2. Kamera modul V2	15
5.3. Motorizált fókuszú lencse	16
5.4. Raspberry környezete	17
5.5. Programozási környezet	17
5.6. Eredeti megközelítés megvalósítása	18
5.6.1. A fókusz kalibrálása	18
5.6.2. Felvételkészítés	20
5.6.3. Élkeresés vs. kontrasztszámítás	21
5.6.4. Kontrasztszámító algoritmusunk működése	22
5.6.5. Depth-map, azaz mélységi térkép	24
6. Eredeti megközelítés eredményei	25
7. Eredeti megközelítés értékelése	28
8. Új megközelítés hardveres és szoftveres alapjai	29
8.1. DSLR fényképezőgép	30
8.2. Modell elrendezések	31
8.3. Programozási környezet	33
8.4. Próba a meglévő algoritmus optimalizálására	33

8.5. Frekvenciatartomány-beli képfeldolgozás, szűrés	35
9. Új megközelítés eredményei	38
10.Összegzés	44
11.Köszönetnyilvánítás	46

1. fejezet

Bevezetés

A gyorsan, megbízhatóan és viszonylag magas pontossággal működő 3D-s térérzékelő rendszerekre és eljárásokra manapság nagy igény mutatkozik a digitalizáció valamint az autonomizáció térnyerésének következtében. Mindennek háttérében a hardveres technológia és az algoritmikus alapú képfeldolgozási eljárások fejlődése áll.

A szemrevételezéses minőségvizsgálatot felváltja az optikai rendszerek ellenőrző funkciója, mely sokkal pontosabb és megbízhatóbb, mint az emberi munka. A biztonságtechnológia terén a telefonok alapvető funkciói között megjelent az arcfelismerés valamint az ujjlenyomatfelismerés, magasabb biztonsági szintekhez pedig az írisz alapú azonosítás. Katonai környezetben az emberi veszteségek enyhítése érdekében távolról irányítható robotokat, drónokat alkalmaznak. De, ha a nem túl távoli jövő kihívásait nézzük, az önvezető autók, a rendőr drónok, a csomagszállító drónok, az okos városok, összesítetten az autonóm rendszerek jelentik a fő irányvonalat. Az előző pár sorban felvázolt megoldások mind képfeldolgozási alapokon nyugszanak, minden bizonnyal ez fogja az alapjukat adni a jövőben is. Egy dolgot bizonyossággal állíthatunk: ahogy az emberi érzékelés, úgy a gépi érzékelés fő vonala is a látás, így a gépi látás és képfeldolgozási eljárások fejlesztése és új megközelítések megalkotása kulcsfontosságú feladat a fejlődés és az innováció szempontjából.

Ahogy láthatjuk, az igény adott és ezek egy részét ki is lehet elégíteni speciális céleszközök tervezésével és legyártásával. Ez sokszor nagy anyagi ráfordítást igényel. Elképzelhető, hogy ehhez adottak a források, sok esetben azonban volna igény egy olcsóbb, egyszerűbb, de az adott feladatot megbízhatóan támogató térérzékelő eljárásra, rendszerre. A kutatómunkám is arra irányult, hogy olcsó eszközökkel megvalósítható térérzékelést valósítsak meg. A pro-

jekt célja, hogy elő lehessen állítani egy egyszerű, de mégis robusztus rendszert, mely mindezek mellett költséghatékony is. A munkám során minösszesen egy kamera felhasználásával igyekeztem megoldani a háromdimenziós tér mélységérzékelését.

A rendszer működését a kamerák fókusztávolságának figyelembevételére alapoztam, mely információt szolgáltat térbeli adatok kinyeréséhez. A gondolat szerint a kamera fókuszpontjának állításával több képet készít a rendszer, melyeket egy algoritmus elemez az éles - tehát a fókuszban levő, így magas kontrasztú - területek után kutatva. Ezzel előáll egy becsült kép a fókuszsíkjában lévő térbeli elrendeződésről. Ezeket az adatokat ezután egy mélységi térképbe szervezi az eljárás, ezáltal megbecsülve a tárgyak elhelyezkedését, kiterjedését. Ezen mélységi térképek felhasználásával lehetőség nyílik autonóm döntéshozatalra, például egy drón esetében arra, hogy az kikerüljön egy akadályt, amely felé közeledik, illetve a térben szabad irányokat határozzon meg.

2. fejezet

Problémafelvetés

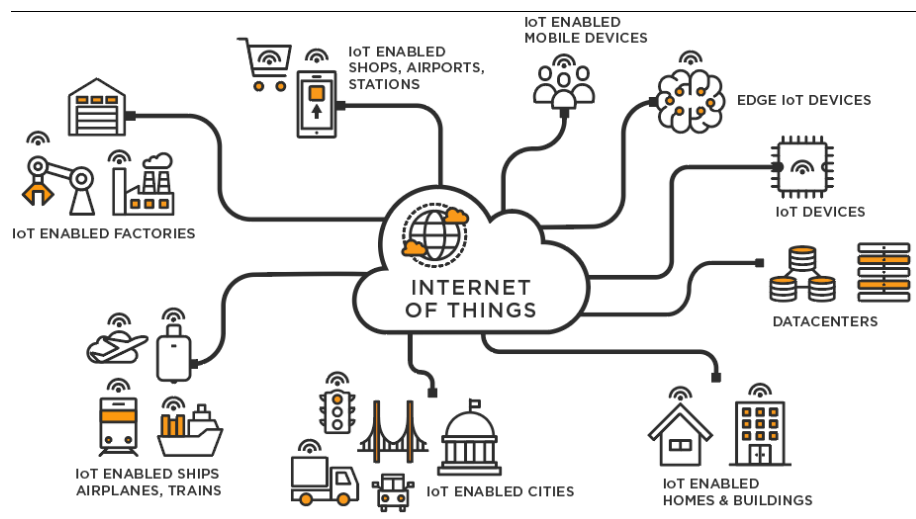
Karunkon számos kutatási projekt foglalkozik gépi látással, képfeldolgozással. Külön kutatócsoport foglalkozik a drónokkal, a bennük rejlő lehetőségek kiaknázásával, fejlesztésével. Ezen kutatások, valamint a dolgozatban leírt projekt nem lenne megvalósítható, ha az utóbbi csaknem egy évszázadban nem történik egy hatalmas, robbanásszerű fejlődés a számítástechnikában. A fejlődés a számítási kapacitások növekedését, a miniaturizálást és csökkenő fajlagos költségeket hozta. Ez azt is jelenti, hogy újabb és újabb felhasználási területek jelennek meg, melyeken gépi látással oldhatunk meg feladatokat.

2.1. A technika fejlődése

A számítástechnika fejlődése elvezetett minket abba a korba, ahol már nagyjából minden használati eszközünkbe számítógépet építünk. Okostelefonokat használunk, okosórát hordunk, okos otthonokban élünk, okos háztartási eszközökkel rendezzük be a lakásainkat, okos járművekkel járunk. Láthatjuk, hogy a személyes életünk és eszközeink nagyban függenek a fejlett és olcsó számítógépektől, hiszen ha nem így lenne, nem venne körül bennünket ennyi "kütyü". A Covid-19 világjárvány meg is mutatta ennek a jelenségnek a hátulütőjét: globális chip hiány alakult ki az ellátási láncok akadozása miatt. Ennek következtében el kellett gondolkodniuk például az autógyártóknak is, hogy mely funkciók azok, melyek nem feltétlenül szükségesek a járművekbe, az emberekben pedig minden bizonnyal elindult egy szemléletváltás az egyszerűbb életre törekvés jegyében. Hiába csábító az érzés, hogy használjuk ki a lehetőségeket, melyeket ez a mérhetetlen ütemű fejlődés hozott az elmúlt csaknem 80 év során, ez nagyon sok hátulütővel, többek között környezetünk és Földünk szennyezésével, kizsákmányolásával, készleteinek kimerülésével jár.

Az említett hatás mellett azonban jótékony következmény, melyet már megfogalmaztam korábban, hogy a számítógépek, valamint önmaga a számítási kapacitás relatíve olcsóvá vált. Az ötödik generációs, azaz a mostani számítógépek fő tulajdonsága a miniaturizálás, mely következtében minél több elem belesűrítése a cél az egyre kisebb chipkebe. Ez a tény vezethetett el minket oda, hogy egy mai okostelefon mérete nagyjából 0,005%-a az első épített számítógép méretének, számítási kapacitása nagyjából 3.500.000-szerese az első számítógépek számítási kapacitásának, ára pedig megközelítőleg 0,0002%-ára csökkent a pénz mai értékében számolva.

Manapság már minden eszköz csatlakozik az internetre, széles körben elterjedt az IoT (Internet of Things) fogalma. A már említett okoseszközök már most kimeríthetetlen, de egyre bővülő tárháza teszi intelligenssé mindennapjainkat. Itt fogalmazódik meg a jelen és a jövő egyik kulcsszereplője: a mesterséges intelligencia. Látó, meghallgató és megértő, beszélő, gondolkodó és tanuló rendszerek, valamint eszközök megalkotása a mérnökök, kutatók célja. Ez egy nagyon ígéretes iránynak bizonyul, tekintettel arra, hogy egy mesterséges intelligencia alapú gépi tanuló eljárás képes bonyolultabbnál-bonyolultabb problémák működésének tanulmányozására, azok megoldására, mindezt egyre kevesebb bemeneti információval. Nem is beszélve arról, hogy ezek az eljárások általában sokkal, de sokkal pontosabban működnek, mint az emberek, egészen pontosan fáradhatatlanul ugyanazzal a teljesítménnyel képesek végig a működésük során.



2.1. ábra. Internet of Things (forrás: <https://www.tibco.com/reference-center/what-is-the-internet-of-things-iot>)

2.2. A fejlődés kihasználása, igények

Az előző részben felvázolt fejlődésből és annak ésszerű és fenntartható felhasználásából a tudományos kutatások, valamint kutatási projektek is profitálhatnak. Alkalmazni tudjuk a fejlődő technológiát új rendszerek kialakításához, akár több innovatív technológiát kombinálva, új innovációt teremtve. Ha csak a drónokat vesszük egy pillanatra a górcső alá, akkor beláthatjuk, hogy mennyire fantasztikus találmányokhoz vezetett a fejlődés kihasználása. Egy mai, nem is feltétlenül csúcskategóriás drón fényképezőgépeket meghazudtoló felvételek készítésére képes, a repülési képességet nyújtó alkatrészeinek és ezáltal repülő funkciójának hála akár olyan területen, mely ember számára bejárhatatlan. Mindezt vezetékes összeköttetés nélkül, meglehetősen kompakt és strapabíró eszközként képes nyújtani egy hasonló drón.

Eredeti projektcélunk megfogalmazásakor is törekedtünk a naprakész technológia és az általa nyújtott lehetőségek kihasználására. Karunk és egyetemünk más kutatócsoportjai is nagyban támaszkodnak a drónok adta lehetőségekre. Drónos navigációhoz, pontosabban térbeli navigációhoz manapság egy vezérlő konzolt irányító ember munkája elengedhetetlenül szükséges, habár karunk kutatóinak korábbi eredményei bebizonyították, hogy kontroller nélkül, csupán gesztusokkal is vezérelhetőek ezek az eszközök. Azonban ez a megközelítés is igényli az ember aktív szerepvállalását az irányítási folyamatban. Innen indult ki az egész ötletelés, miszerint jó lenne egy olyan eljárás, mely képes a dinamikusan változó térben biztonságosan, és ami talán ugyanennyire fontos, valós időben irányítani a drónt, mindezt aktív emberi beavatkozás nélkül. Ehhez természetesen elengedhetetlen a megfelelő számítási kapacitás. Ezt eredetileg a fejlődés egy másik áramlatba ejtő eredményével, egy nagyjából bankkártya méretű, csupán egy áramköri lapból álló miniPC-vel kívántuk megoldani. A térérzékelésre csupán egyetlen kamera képét szeretnénk felhasználni, mely merőben más megközelítés az általános - két lencsés - háromdimenziós érzékelésre alkalmas kamerákhoz viszonyítva.

Egy ilyen, vagy egy ehhez hasonló megoldásnak más felhasználási területen is lenne keresnivalója, mely célja a 3D-s környezet feltérképezése, a környezetben lévő tárgyak pozíciójának, orientációjának és kiterjedésének nagyjából meghatározása. A navigációs algoritmus megalkotásához igénybe szeretnénk venni az intelligens rendszerek nyújtotta osztályozási képességeket, mely tovább növelné a megoldás autonómítását, emberfüggetlenségét.

Végső soron, az önállóan működő gépeké a jövő. Összességében elmondható, hogy az a hosszútávú cél, hogy minél jobban terhelhető, fáradhatatlan, és ezáltal végig ugyanolyan megbízhatósággal működő entitásokat hozzunk létre. Ezzel valójában az emberek terhelésének csökkentését érénk el, hozzájárulva egy tartalmasabb, értékesebb élethez.

3. fejezet

Irodalom

Az ötletem már másban is felmerült, az irodalom Depth from Focus (DfF) módszerként ismeri és a lehetőségeit többen vizsgálták már. Az alapötlet vizsgálata közben az eddigi eredményeket és módszereket is vizsgáltam. Általánosságban elmondható, hogy az irodalomban főleg nagy pontosságú és számításigényes módszereket láttam. S. Salokhiddinov és társa (Depth from Focus for 3D Reconstruction by Iteratively Building Uniformly Focused Image Set (SIGGRAPH '18: ACM SIGGRAPH 2018 Posters August 2018 Article No.: 60, <https://dl.acm.org/doi/10.1145/3230744.3230799>) cikkében egy iteratív megoldást mutat be, amelyben változó ablakmérettel több képen újra és újra kell számításokat végezni. A bemutatott eredmények jók, de a futási idők a 100-as nagyságrendű képcsomagokon nem engedik meg a valós idejű felhasználást.

M. Moeller és társai (Variational Depth from Focus Reconstruction, IEEE Transactions on Image Processing, 2015, <https://ieeexplore.ieee.org/document/7271087>) egy módosított Laplace eljárással mérik a kontrasztot. Az általuk kidolgozott módszer meglehetősen bonyolult matematikai alapokon nyugszik. Az eljárásukkal pontos eredményeket és formákat várnak eredményül, mely előállítása meglehetősen számításigényes. A számítások során a GPU kapacitását veszik igénybe. Ez a megközelítés a magas számításigényének köszönhetően vélhetően nem alkalmas valós idejű megoldásokhoz.

S. Suwajanakorn és társai (Depth from Focus with Your Mobile Phone, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3497-3506) egy kézi eszközökre implementált eljárást fejlesztett ki, mely pásztázással (swing) és fix fókusszal készít egy képsorozót, melyet utólagos feldolgozás alá vet. Az eljárásuk megbízhatónak tűnik, azonban egy 25 frame-es sorozatnak a feldolgozási ideje nagyjából 20 percet vesz igénybe, így

valós idejű felhasználásra ez a megközelítés sem alkalmas.

Mindezen áttekintések számos jó gondolatokkal látták el a munkámat, azonban az időigényük miatt mindenképpen más irány figyelembe vétele is szükséges az általunk célul kitűzött eredmények elérése érdekében.

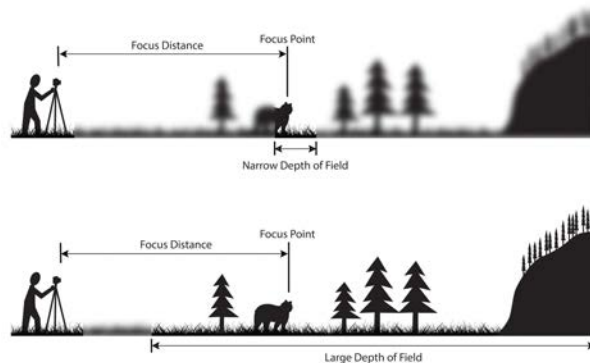
4. fejezet

Anyag és módszer - általánosságban

A következő fejezetekben kívánom bemutatni az eszközöket és a módszereket, melyet a projekt során felhasználunk, alkalmazunk. A projektmunka eddig két fő részre bontható, melyek a megközelítésben bekövetkezett változás miatt jöttek létre. Szeretném bemutatni az eredetileg definiált követelmények szerint alkalmazott eszközöket, eljárásokat, melyek az előzőleg említett egyszerű technológiával oldanák meg a problémát. Emellett szívetném részletesen bemutatni a korlátokat, melyekkel az első megközelítéssel szembesültünk, illetve az új irányt, mely az eljárás alapvető működését gondolja teljesen újra.

Közös azonban mindkét megközelítésben az alap működési elv. Mindkét esetben egy rögzített pozícióból készítünk több képet, melyek mind más, egymást azonos lépésközzel követő fókusztávolsággal (focus distance) rendelkeznek. A fókusztávolság megadja, hogy a kamera lencséjétől milyen távoli dolgok lesznek élesek a képen [9]. Ideális eset, amikor a térben a végtelenségig tudjuk ezt változtatni, azaz tetszőlegesen távoli (és közeli) részek lehetnek élesek a képen, mindezek nagyjából egy nagyon vékony, a tér már-már sík felületén. Ez azonban a fényképezőgépek és kamerák valós működéstől nagyon távol áll. Van ugyanis egy másik tulajdonság, amely befolyásolja az éles terület nagyságát/mélységét: a mélységélesség (depth of field). A mélységélesség valójában egy sáv a fókusztávolság környezetében, melyben benne lévő tárgyak élesen látszanak [8]. A dolgot tovább nehezíti, hogy a mélységélesség sávjának a mérete dinamikusan változhat a fókusztávolság változtatásával együtt. Ezzel eljutottunk oda, hogy az ideális síkszerű fókusztávolság a valóságban egy tartománya a térnek, mely kiterjedése folyamatosan változik a fókusztávolság változtatásával. Ezek a tulajdonságok a kamera

lencsési által definiált tulajdonságok, ezáltal két különböző kamera különböző tulajdonságokkal bír.



4.1. ábra. Fókusz távolság és mélységélesség (forrás: [https://photographylife.com /what-is-depth-of-field](https://photographylife.com/what-is-depth-of-field))

Az elkészített, azonos lépésközű fókusz távolsággal rendelkező képek a tér több síkjának tartalmazzák éles területeit. A következő lépésben ki kell nyernünk ezekből a képekből az éles területeket, azaz azokat, melyek magas kontraszttal rendelkeznek. Ehhez szükség van egy eljárásra, amely képes számszerűsíteni és ezáltal meghatározni a kontrasztot a kép minden egyes részterületén. A magas kontrasztú területek - képpontok - kiválogatásával előáll az adott fókusz távolságban élesen látszó objektumok közelítő körvonala.

Ezen kontrasztképek felhasználásával pedig egy úgynevezett mélységi térkép előállítása a cél, mely térben jeleníti meg az eddig kétdimenziós információkat. A megjelenítés jelen esetben csak az emberi interpretáció szempontjából fontos, a számítógépnek nem feltétlenül szükséges ismernie ilyen módon a teret, erről a későbbiekben részletesebben fogok írni.

5. fejezet

Eredeti megközelítés hardveres és szoftveres alapjai

Ebben az alfejezetben vannak részletezve azok a megközelítések és eszközök, melyek az eredeti terveink szerinti megvalósítást írja le.

A fizikai eszközök tekintetében arra törekedtünk, hogy lehetőleg egyszerű és olcsó eszközöket válasszunk. Azért tartottuk ezt fontosnak, mert a piacon elérhető 3D-s térérzékelésre alkalmas megoldások egyik gyenge pontja az áruk. Egy olcsó, szinte mindenki számára megfizethető eljárás viszont nagy hozzáadott értéket jelentene a tudománynak.

További, és megfotolandó szempont volt még a méret is. Mivel alapvetően drónos megoldásokban szeretnénk a rendszert használni, így kulcsfontosságúvá válik a méret és a súly kérdése is. Igaz, hogy típusfüggően meghatározott súly elhordozására képesek a drónok, azonban a plusz súly nagyban befolyásolja a repüléssel eltölthető időt. Minden drón egy bizonyos kapacitású tápegységgel rendelkezik, melyből minden energiafogyasztást fedeznie kell. Ezáltal minél több súlyt hordoz, annál több energiát használ el az eszköz a repülési műveletekhez a nagyobb ellenállása miatt, ami a repüléssel töltött idő drasztikus csökkenéséhez vezet. Fontos tehát, hogy a lehető legkisebb tömegű kiegészítő rendszert kelljen a drónnak hordoznia, ezáltal a repülési idő is maximalizálható.

Mindezek mellett nem szabadott elmenünk a teljesítmény mellett sem, olyan hardverre volt szükségünk, mely megfelelő számítási kapacitással rendel-

kezik. Képfeldolgozási probléma lévén, számítottunk rá, hogy erőforrásigényes lesz az eljárás.

A kiválasztott eszközök között szerepel egy Raspberry Pi 3 B miniPC, egy kamera modul, amit felszereltünk egy motorizált fókuszú lencsével.

5.1. Raspberry miniPC

A Raspberry Pi valójában egy áramköri lapkára integrált egykártyás számítógép, mely 2012-ben, eredetileg oktatási célra lett kifejlesztve [3]. Az eszközt kifejezetten a fejlődő országokba szánták annak érdekében, hogy széles körben elterjedjen az alap számítástechnikai, programozási oktatás az iskolákban. Mint ahogy az már utólag látszik, az eredeti célkitűzéseknél sokkal népszerűbb lett az eszköz, ez a sikeresség nagyban köszönhető annak, hogy rengeteg szenzoros, aktuátoros bővítési lehetőséggel rendelkezik, mely összességében rettentően sokoldalúan felhasználható eszközzé teszi. 2019-re már több, mint harmincmillió Raspberry Pi eszközt adtak el világszerte, felhasználva őket akár a robotikában, akár gyárakban a gyártósori gépek irányításában. Az elmúlt több, mint tíz év alatt számos verzió, kialakítás látott már napvilágot, jelenleg a Raspberry Pi 4 B verzió a legújabb.

A projekt kezdeti fázisában egy Raspberry Pi 3 B modellt használtunk. Az eszköz egy négymagos, 1,2 GHz-es órajelű processzorral van felszerelve. A számítógép 1 GB operatív memóriát kapott, mely manapság már igen csekélynek tűnik. Képes vezeték nélküli kapcsolat létrehozására Wi-Fi és Bluetooth szabványokon keresztül valamint rendelkezik Ethernet porttal [4].

Raspberry Pi 3 B	
Processzor	Broadcom BCM2537
Magok száma, órajel	4*1,2 Ghz
GPU, órajel	Broadcom VideoCore IV 400 MHz
SDRAM	1 GB
Ethernet	10/100 Mbit/s
Wi-Fi	802.11.b/g/n, 2,4 GHz
Bluetooth	4.1 BLE
Méretek, tömeg	85,6 mm * 56,5 mm * 17 mm, 45 g

5.1. táblázat. A Raspberry Pi 3 B miniPC specifikációjának fontosabb elemei (forrás: https://en.wikipedia.org/wiki/Raspberry_Pi)



5.1. ábra. Raspberry Pi 3 B (saját kép)

5.2. Kamera modul V2

A Raspberryhez időközben számos külső szenzort fejlesztettek ki, melyek között kamera modulok is helyet kaptak. A projekt során már a második generációs modult használtuk, mely a Sony IMX219 8 megapixeles szenzorát tartalmazza [5]. A kamera modul nagyon jól használható nagy felbontású videófelvételek készítéséhez, valamint fényképezéshez. Képmínősége rossz megvilágítási viszonyok között is megfelelően jó. Számos harmadik féltől származó funkciókönyvtár látott már napvilágot a modulhoz, ezek közül az egyik legjelentősebb és legki-forrottabb a Picamera Python könyvtár [5]. Ez a tény nagyban befolyásolta, hogy milyen irányba kell elindulnunk a szoftveres környezet terén. Probléma viszont, hogy a szenzor önmagában alap esetben fix fókuszú, tehát önállóan a kamera modul sajnos nem elégíti ki a fókusz állítására vonatkozó igényünket, melyet a megvalósítás során megfogalmaztunk.



5.2. ábra. Kamera modul V2 (forrás: <https://www.raspberrypi.com/products/camera-module-v2/>)

5.3. Motorizált fókuszú lencse

Amint az előző alfejezetből kiderül, a V2-es kamera modul sajnos nem alkalmas alap esetben a fókusz távolság állítására, ezzel szemben az 1 métertől a végtelenig terjedő tér minden egyes pontja élesen látszik a képen, tehát fix fókuszú. Viszont a kamera modul tervezésekor a mérnöki megfontoltság helyét adott akár a külső fejlesztési lehetőségekre is, így a kamera modul cserélhető a kamera board-on. Elérhető ugyanis az Arducam IMX219 Auto Focus Camera Module, mely segítségével felszerelhetjük a kameránkat egy állítható fókuszú lencsével. Ennek következtében alkalmassá válik az eszköz a projektekben megfogalmazott igények kielégítésére. Ez a szenzor is a Sony 8 megapixeles eszközét használja, kifejezetten a V2-es kamerához lett kifejlesztve.



5.3. ábra. Arducam kamera modul (forrás: <https://www.arducam.com/product/arducam-imx219-auto-focus-camera-module-drop-in-replacement-for-raspberry-pi-v2-and-nvidia-jetson-nano-camera/>)



5.4. ábra. Camera modul V2 + motorizált fókuszú lencse (saját kép)

5.4. Raspberry környezete

A szoftveres környezet egy része determinált, habár a Raspberryken valójában bármilyen operációs rendszert el lehet futtatni, melynek kielégíti a minimális rendszerkövetelményeit. Mégis úgy gondoltuk, hogy a kifejezetten Raspberryhez kifejlesztett és ajánlott operációs rendszert, a Debian alapú Raspberry Pi OS-t (korábban Raspbian) [6] fogjuk használni a lehető legoptimálisabb működés elérése érdekében. Az operációs rendszer olyan verzióját használtuk, mely rendelkezik asztali környezettel, csupán a fejlesztőmunka megkönnyítése érdekében.

5.5. Programozási környezet

A kamera modul bemutatásánál már említettem, hogy az egyik legjobb függvény-csomag, mely elérhető a vezérléshez, Python nyelven íródott. Szerettük volna implementálni az abban megtalálható függvényeket, így a kamera vezérlésére egy Python környezetet konfiguráltunk.

A további, képfeldolgozási munkálatokhoz is a Python adta lehetőségeket szerettük volna kihasználni. Tudvalevő, hogy a Python kiválóan alkalmas matematikai problémák megoldására, tudományos számolások elvégzésére. Tulajdonképpen a képfeldolgozás is felfogható matematikai műveletek összességéként, így kézenfekvő a választás.

A Raspberry-n az eszközre legfrissebben elérhető, Python 3.7-es verziója futott, azonban ez nem alkalmas háromdimenziós plotok készítésére, így a számítógépen telepítettem a 3.9-es verziót az ábrázolásra. Valójban az ábrázolás nem feltétlenül kell, hogy az algoritmus feladata legyen, így ez a megoldás nem vesz el az eljárás értékéből.

5.6. Eredeti megközelítés megvalósítása

Ebben a fejezetben szeretném ismertetni az eredeti megvalósítás eljárását, a megvalósítás munkafolyamatait.

A megvalósítás során legelőször ki kellett tapasztalnunk az Arducam modul működését. Sajnos, leírások hiányában csak néhány példa programkód állt rendelkezésünkre az eszközzel kapcsolatban. Fontos, és elhanyagolhatatlan feladat volt, hogy kiderítsük, a tér melyik tartományában működik a fókuszállítás, ezt milyen paraméterekkel tudjuk beállítani. Ez volt a kalibrációs folyamat.

A következő lépés a képkészítés eljárásának megalkotása, mely különböző fókusz-távolságok alkalmazásával készít 5 képet a környezetről.

A felvételkedzítést követte a képek utófeldolgozása, mely magába foglalta a méretcsökkentést a feldolgozási idő csökkentése érdekében, valamint a kontrasztos részek megkeresését az élkereső eljárások helyett egy kontrasztszámító algoritmus felhasználásával.

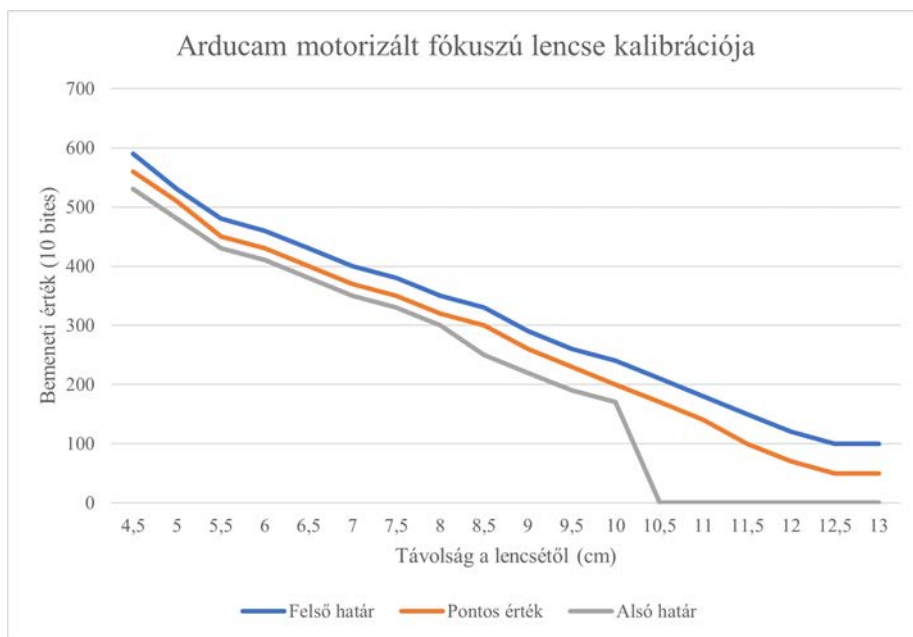
Az elemzést pedig a mélységi térképen való ábrázolás követte, mely ahogy már említettem, az asztali számítógépen történt meg egy másik Python verzió felhasználásával.

5.6.1. A fókusz kalibrálása

A manuális fókuszt implementáló mintakód segítségével megállapítottuk, hogy a kamera bemenete 10 bites, azonban nem kezeli mind az 1024 értéket. Irányadó, hogy minél nagyobb a bemenet, annál közelebbre fókuszál a kamera. A kamera kalibrálására két mérést végeztünk, melyek során az alábbi eljárást követtük:

- Rögzítettük a kamerát egy adott ponthoz a környezetben.
- Mérőszalagot helyeztünk a kamera mellé, 0 cm-nél maga a lencse feküdt.
- Egy szemvizsgálatra alkalmazott tábla segítségével fél centiméterenként megvizsgáltuk a látott képet.
- Minden egyes állásban feljegyeztük az ideális fókusz-távolsághoz tartozó bemeneti értéket, valamint a mélységélesség határaihoz tartozó bemeneti értékeket.

A kalibrálások során mindkét esetben szinte teljesen azonos eredményt kaptunk. Fontos megjegyezni, hogy ez egy empirikus módszer, egyénfüggő eredményt kapunk. Konkrét leírás és adatok hiányában azonban ennél jobb és kézenfekvőbb megoldást nem találtunk a probléma áthidalására.



5.5. ábra. Arducam motorizált fókuszú lencse kalibrációja

Távolság (cm)	Felső határ	Pontos érték	Alsó határ
4,5	590	560	530
5	530	510	480
5,5	480	450	430
6	460	430	410
6,5	430	400	380
7	400	370	350
7,5	380	350	330
8	350	320	300
8,5	330	300	250
9	290	260	220
9,5	260	230	190
10	240	200	170
10,5	210	170	0
11	180	140	0
11,5	150	100	0
12	120	70	0
12,5	100	50	0
13	100	50	0

5.2. táblázat. Arducam motorizált fókuszú lencse kalibrációjának számszerűsített bemeneti értékei

Az eredmények alapján látható, hogy a legkisebb elérhető fókusz távolság 4,5 cm. A mélységélesség tartománya a kezdeti elég kicsi tartományhoz viszonyítva fokozatosan szélesedik, olyannyira, hogy 10 cm-től már végtelenbe nyílik a tartomány. Mindezek következtében meg kell jegyeznünk, hogy a kamera modul sajnos felállította az első korlátot az eredeti elképzelés kapcsán, ugyanis ez a rendszer, ebben a formában csak nagyon közeli és viszonylag kis méretű objektumok esetében képes ellátni a feladatát. Mindezek ellenére az eljárás ilyen kis tartományban is kidolgozható, sőt működőképes kell, hogy legyen, emiatt nem változtattunk az eredeti elrendezésen.

5.6.2. Felvételkészítés

A kamera modul működését tanulmányozva megállapítottuk, hogy maximálisan mennyi időre van szüksége a hardvernek ahhoz, hogy átállítsa a fókusz az igények szerint, valamint mennyi időre van szüksége a képkészítési feladat és a mentés elvégzésére. Ezek alapján egy viszonylag gyors - maximum 0.5 s alatt lefutó - algoritmus elkészíti a képeket. Ebben az algoritmusban végezzük el a

képek átméretezését is, a tapasztalataink alapján 500 pixel széles képek elegendőek a kontrasztszámításhoz.

5.6.3. Élkeresés vs. kontrasztszámítás

Sokat gondolkodtunk, hogy milyen információt nyerjünk ki a képekből, amely segítségével létre tudjuk hozni a mélyégi térképet. Mivel Pythont használtunk, kézenfekvő lett volna egy függvénykönyvtár egyik élkereső eljárását felhasználni, és az élek alapján felépíteni a depth-mapet.

Az élkereső eljárások alapja a konvolúció, mely értelmezhető egydimenziós adatsorokon, valamint kétdimenziós adathalmazokon is. Utóbbi esetben kétváltozós függvény teremti meg a kapcsolatot a bemeneti adathalmaz és a kimeneti adathalmaz között. Jelen esetben a 2D-s adathalmazok maguk a képek, a függvény pedig az élkeresési eljárás implementálása. „... egy $k[n,m]$ kép impulzusokra való felbontásáról beszélünk, akkor az egyes impulzusok értéke a kép n -edik sorában és m -edik oszlopában álló pixel fényességértéke (az egyszerűség kedvéért szürkeárnyaltos 256 színű képekre gondoljunk, ahol minden képpont fényességét egy-egy byte tartalmazza).” (Dr. Sári Zoltán: *Bevezetés a számítógépes jelfeldolgozásba*, 16. oldal)

Konvolúciós eljárásokat alkalmazunk például egyebek mellett a zajszűrési eljárások alkalmazása során is [2].

$$y[n, m] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} x[i, j] * p[n - i, m - j]$$

5.6. ábra. A kétdimenziós konvolúció matematikai háttere

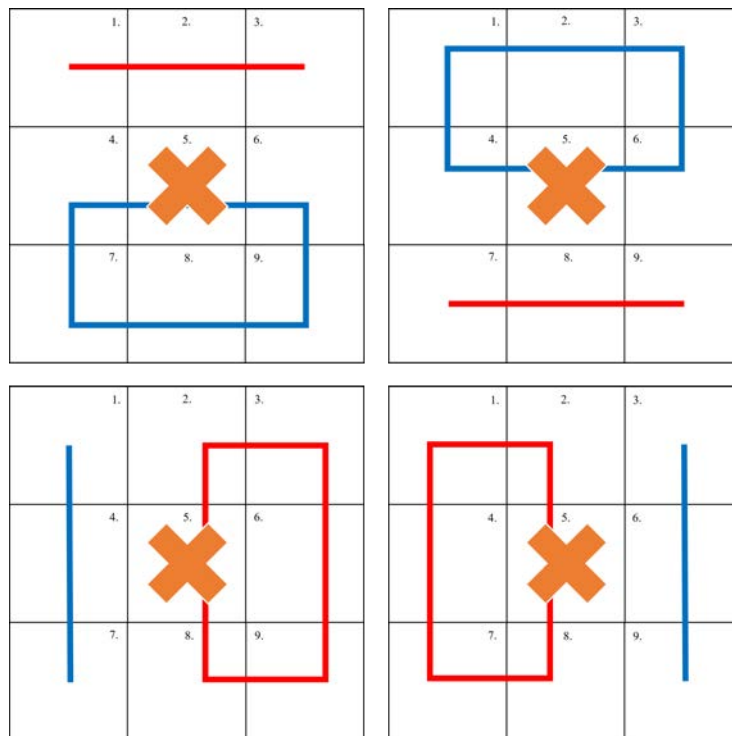
Az élkeresési eljárások lényege tulajdonképpen az, hogy meghatározza az objektumok kontúrjait. Olyan pontok megkeresése a cél, mely környezetében a fényerősség, azaz az intenzitás ugrásszerűen megváltozik [2]. Nekünk első sorban csak az éles területeken szükséges a kontúrmeghatározás, amely valójában a kontrasztszámítással meg is oldható. Az éles területeken az ugrásszerű intenzitásváltozások lesznek az élek. Valójában ez is egy élkeresési eljárás, csak kifejezetten a problémára paraméterezve, megalkotva (az élkeresési eljárások általánosságban véve az egész képet elemzik, érzékenyebben a nem fókuszban lévő területekre is, ezáltal olyan éleket is megtalálnának a mi esetünkben, melyeknek nem lenne relevanciája a probléma szempontjából - nem a fókusz síkjában lévő körvonalak).

5.6.4. Kontrasztszámító algoritmusunk működése

A kontrasztvizsgálathoz szürkeárnyaltos képeket kell felhasználnunk. Egy szürkeárnyaltos kép minden egyes képpontja az adott képpont színe helyett az intenzitását reprezentálja, mindegyik 2^8 , azaz 256 különböző értéket vehet fel.

Kontrasztvizsgálat során megkeressük azokat a részeit a képnek, melyek mentén nagy intenzitásváltozás figyelhető meg. Ilyenkor nem két képpont intenzitását hasonlítjuk össze, hanem egy kis részét elemezzük a képnek, néhány képpontját. Hasonlóan a konvolúcióhoz, egy kernel képpontjait vizsgáljuk, azok egy részét hasonlítjuk a többi részéhez, ezáltal meghatározva, hogy azon belül megfigyelhető-e éles intenzitásváltozás.

Az algoritmusunk egy 3×3 -as kernelt futtat végig az összes belső képponton (szélső sorok és oszlopok a kernelvizsgálat tulajdonsága miatt érintetlenek). A kernelen belüli képpontok intenzitásarányát vízszintesen és függőlegesen is vizsgáljuk.



5.7. ábra. Kernel, vizsgált átalgok

Ahogy azt a 5.7. ábra mutatja, négy vizsgálat történik: a felső három pixel intenzitásátlagát hasonlítjuk az alsó hat átlagához, a felső hat átlagát az alsó három átlagához, a bal oldali három pixel intenzitásátlagát a jobb oldali hat átlagához, valamint a bal oldali hat átlagát a jobb oldali három átlagához. A számolások során egyszerű matematikai átlagot használtunk. A képpontok intenzitásértékei normalizáltak, azaz 255-tel el lettek osztva a vizsgálat előtt. Ha két terület között szeretnénk eldönteni, hogy nagy-e az intenzitáskülönbség, azaz kontrasztos-e az adott rész, a következő műveletet kell elvégeznünk:

$$k = \frac{p_1 + 0,05}{p_2 + 0,05}$$

5.8. ábra. Kontrasztszámítás matematikai képlete

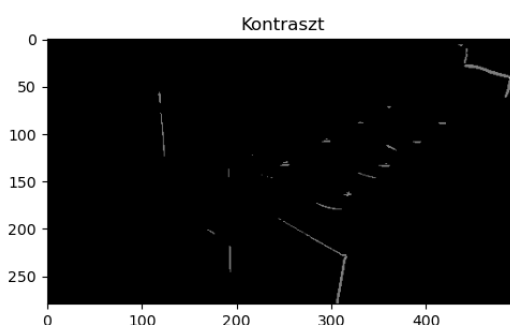
, ahol p_1 és p_2 az előbb kiszámolt átlag-párok.

Az így kapott arányszám egyértelműen jellemzi a két terület között lévő kontrasztot. Amennyiben ez az arány a lehető legnagyobb értékű - +21, mely akkor adódik, ha $p_1 = 1$ és $p_2 = 0$ -, vagy a lehető legkisebb értékű - $+\frac{1}{21}$, mely akkor adódik, ha $p_1 = 0$ és $p_2 = 1$ -, akkor a két terület között nagy a kontrasztkülönbség. Amennyiben viszont ez az arány egyenlő 1-gyel - $p_1 = p_2$ -, akkor a két terület között nincs kontrasztkülönbség.

Ezeket a számolt értékeket nem lehet közvetlenül kimenetként elmenteni, megjeleníthető és dekódolható értékekké kell őket alakítani. Ennek érdekében külön-böző intervallumokban - összesen 5 - lévő arányértékekhez hozzárendeltünk egy előre definiált kimeneti értéket, mellyel a kimeneti képen az adott kontraszt-pixelt helyettesíteni lehet. Ezzel egy olyan kimeneti kép áll elő, mely összesen maximum 5 különböző intenzitásértéket tartalmaz, a nagyobb értékkel jelölve a kontrasztosabb területek képpontjait.



5.9. ábra. Bemeneti kép



5.10. ábra. Kimeneti kontrasztkép

Látható, hogy az algoritmus azonosítja azokat a részeit a képnek, melyek a fókuszban élesen látszanak, azaz kontrasztosak. Természetesen, a paraméterezés finomításával hangolható az algoritmus, azonban minél részletesebb eredményt szeretnénk kapni, annál nagyobb a veszélye annak, hogy irreleváns területek is megjelennek a képen, mely élek valójában nem is a fókusz síkjában vannak. A megvilágítási viszonyok is okozhatnak ilyen jelenséget, ennek hatását a megfelelően szigorú paraméterezés csökkenteni tudja.

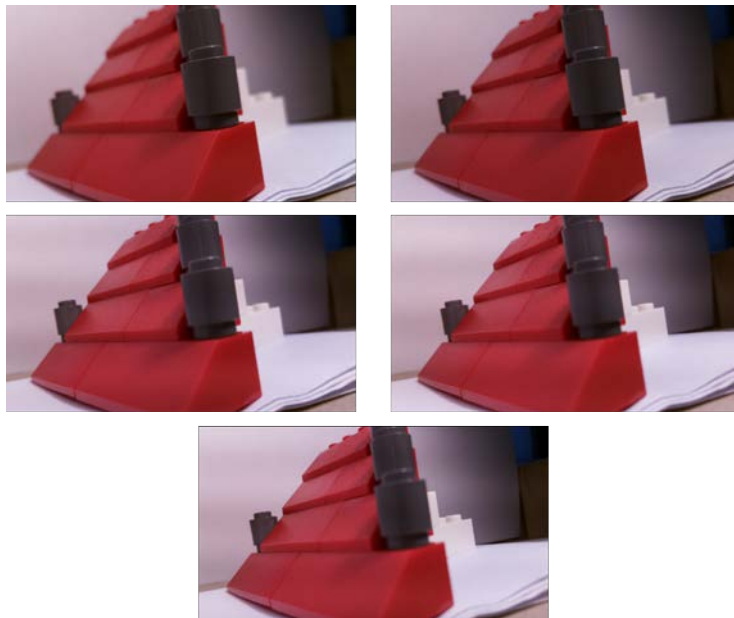
5.6.5. Depth-map, azaz mélységi térkép

A kimeneti képek elemzésével meg tudjuk keresni azokat a pontokat, melyek intenzitásértéke különböző 0-tól, azaz kontraszt információt tartalmaznak. Mivel a képek térben meghatározott távolságra készültek egymástól, a képből kiolvasható x és y koordinátákon kívül egy harmadik, z koordinátát is hozzá tudunk rendelni a pontokhoz, mely a térben való, a kamerától mért mélység szerinti elhelyezkedésüket mutatja. Így, valójában minden képnél előáll egy olyan pontthalmaz, amely megadja a kép éles síkjában lévő kontrasztos pontok pontos térbeli elhelyezkedését.

6. fejezet

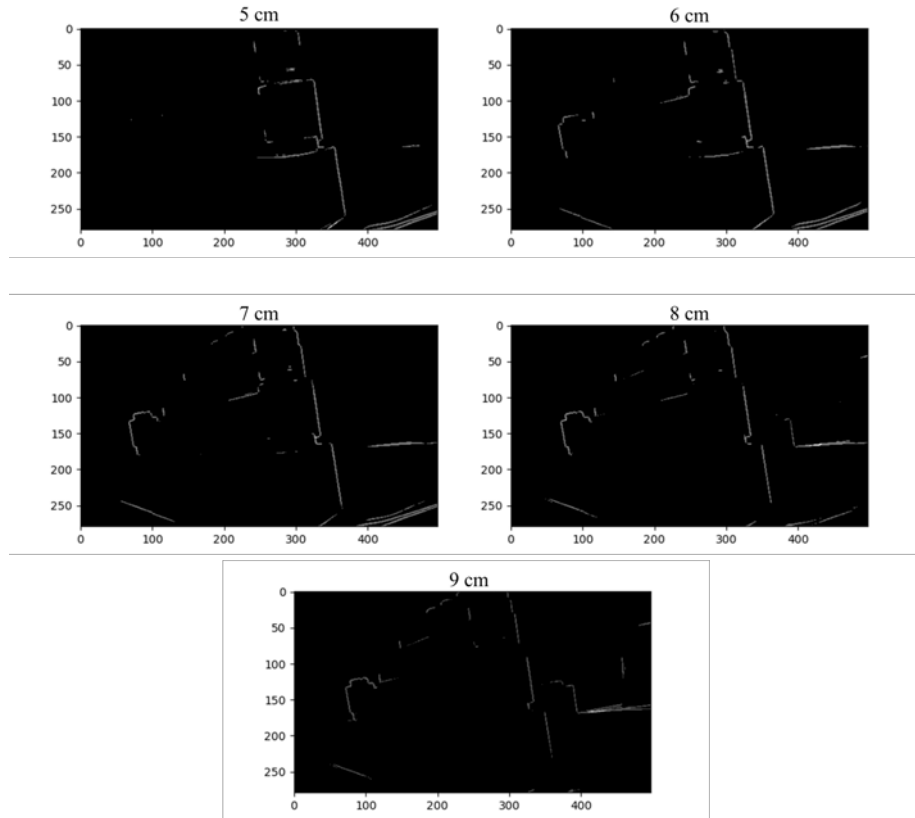
Eredeti megközelítés eredményei

A kísérletek során LEGO építményekről készültek a felvételek. Jó választásnak tűnt, ugyanis a kamera adta szigorú méretbeli korlátozások nem engedték nagy kiterjedésű tárgyak felhasználását. Olyan építmények vizsgálatát végeztük el, melyeknek a térben a kamerára merőleges tengelyen távolodva, tehát a fókusztávolságot növelve változik a szerkezetük, elrendezésük. A fényképeket 5 cm-es, 6 cm-es, 7 cm-es, 8 cm-es és 9 cm-es fókusztávolságokkal készítettük el.



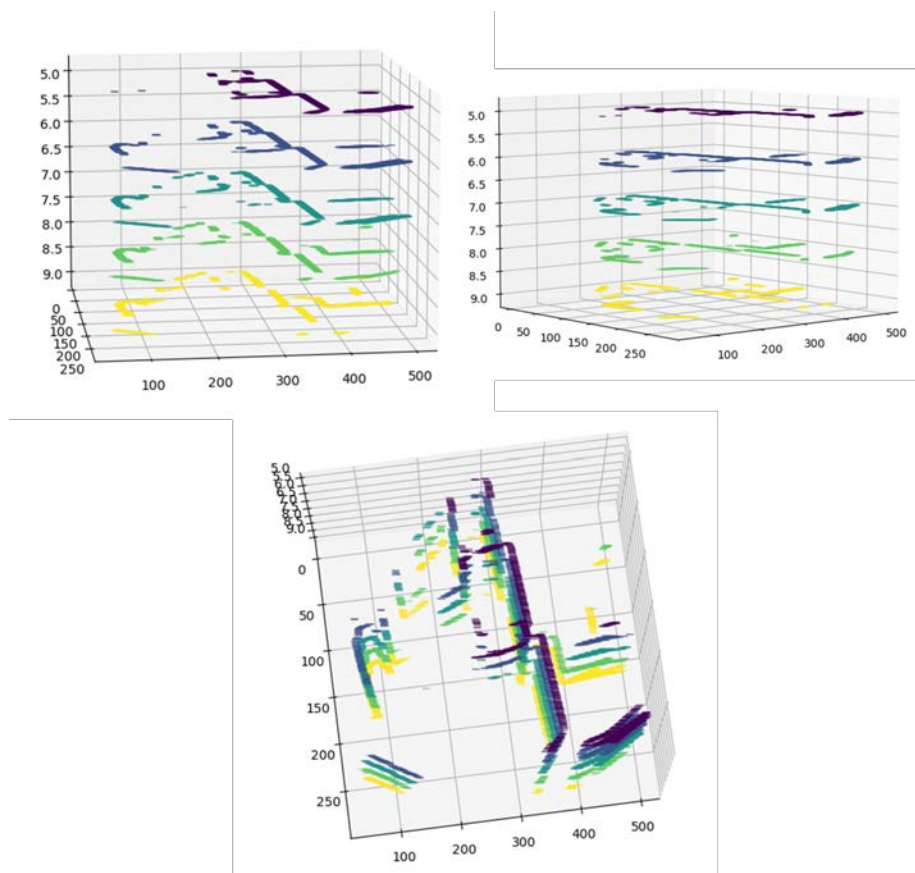
6.1. ábra. Első elrendezés képei

Az első elrendezés képeihez tartozó kontrasztképek:



6.2. ábra. Első elrendezés kontrasztképei

Nagyon jól megfigyelhető, hogy amikor alacsony a fókusztávolság, akkor a kép közepétől kicsit jobbra lévő rész az, amely a kontrasztképeken megjelenik - a várakozásainknak megfelelően, ugyanis az a része volt az adott képnek éles. Ahogy növeljük a fókusztávolságot, fokozatosan távolodunk a kis építmény eleje felől hátrafelé, ezáltal megjelenik a tető egyre hátrébb fekvő része, egészen a hátsó kis hengerig a bal oldalon. Mindezzel párhuzamosan a képek közepétől kicsit jobbra fekvő, az első kontrasztképen erősen látszó terület kezd eltűnni a hátulsóbb elhelyezkedő kontrasztképeken. A kapott eredmények mélységi térképbe rendezése után egy becsült elrendezését láthatjuk a térnek, egy síkokra lebontott becslésről van szó. A depth-map szempontjából amúgy lehet, hogy előnyösebb lenne a bináris képek használata, az öt értékű kontrasztintenzitás megjelenítés helyett, ugyanis ebben az esetben nincs súlyozva a mélységi térképen a kontraszt erőssége, aszerint ahogy a kontrasztképeken látható.



6.3. ábra. Első elrendezéshez tartozó mélységi térkép

Az előbb említett súlyozatlanság miatt látszik úgy, mintha minden szinten megjelenne az építmény elején lévő kis hengertorony. Ezt kiküszöbölendő fel kell használni az adott síkot megelőző és követő kontrasztképeket is, amelyek alapján pontosabban meg lehet mondani az összetartozó pontok intenzitásváltozásai alapján, hogy melyik síkban helyezkedik el az adott objektum a legnagyobb valószínűséggel.

7. fejezet

Eredeti megközelítés értékelése

Összességében elmondható, hogy az eredeti elképzeléseink szerint és az elgondolt elrendezés segítségével és eszközökkel megoldás implementálható a problémára.

Igaz, hogy mind a Raspberry Pi 3 B-nek, mind a kamera modulnak a motorizált fókuszú lencsével vannak nagyon erős korlátai. A miniPC-nek a számítási kapacitása a tesztek során kevésnek bizonyult. Ezt kiegészítheti az a tény is, hogy az algoritmus nincs a végletekig optimalizálva. Ezen hatások összességéként sajnos a megoldás nagyon messze áll a valós idejűség feltételeitől. Az öt 500 pixel széles kép teljes feldolgozása csaknem 50 másodpercet vesz átlagosan igénybe, mely több tízszerese az elvárt futásidőnek. A kamera által támasztott méretbeli korlátok sajnos nem teszik a rendszert alkalmassá "nagyban" történő felhasználásra.

Mindezek arra vezettek bennünket, hogy gondoljuk végig, valóban a miniPC-s vonalon szeretnénk tovább folytatni a munkát? Ha igen, akkor találunk olyan optikát, amely alkalmas nagyobb távolságokon is elvégezni ugyanezt a feladatot, amit a felhasznált eszköz a 10 cm-es környezetében? Ha igen, akkor tudjuk az algoritmust úgy optimalizálni, hogy megállja a helyét egy közel valós idejű rendszerben?

Ezen kérdések kapcsán el kellett gondolkodnunk új irányok keresésén mind eszközök terén, mind algoritmikus téren. Azt továbbra is fenntartva, hogy az alap elképzelés működik és érdemes tovább finomítani.

8. fejezet

Új megközelítés hardveres és szoftveres alapjai

Az előző fejezetekben bemutatott megközelítés az egész folyamatra szánt új eszköz megalkotását tűzte ki célul. Az volt az irányelv, hogy lehetőleg minden, ami részt vesz a folyamatban legyen fizikailag is egy helyen, lehetőleg a drónon. Ez határozta meg a számítógép kiválasztását, ami valamilyen szinten determinálta az optika kiválasztását is. Ez a megközelítés alapvetően nem rossz, ugyanis az általunk felhasznált Raspberry Pi 3-as verziónál már elérhető újabb verzió, mely nagy előrelépést jelentene a számítási kapacitás miatt megjelent szűk keresztmetszet tekintetében. Példának okáért, a 4-es verzió 1,5 GHz-es órajelű processzorral szerelt az általunk használt előd 1,2 GHz-es processzorával szemben, valamint már elérhető akár 8 GB RAM memória az általunk használt 1 GB helyett. A számítási kapacitás problémája abban az esetben megoldható lenne az újabb eszközzel, ha megfelelően optimalizált programot tudnánk írni a feladat elvégzésére.

A nagyobb problémát az optika jelenti: ha csak a kifejezetten Raspberry eszközökhöz gyártott kamera modulok és optikák kínálatát vesszük, akkor nem találunk az igényeinknek teljes mértékben megfelelő eszközt. Értem ezalatt azt, hogy a rendelkezésre álló optikák nagy része nem alkalmas fókuszállításra, továbbá ami alkalmas is lenne, az nem működik optimálisan az emberi léptékben mérhető környezetben. Ugyan a módszer jónak tűnt, a további vizsgálódáshoz egyelőre más optikai rendszereket kellett keresnünk.

Abban az esetben, ha más rendszerből származó kamerát - akár a drón saját kameráját - alkalmazunk, akkor viszont feleslegessé válhat a Raspberryt fel-

használni a számítási feladatok megoldására. Ennek következtében egy erősebb, külső eszköz is meg tudná oldani a program futtatását, de természetesen csak abban az esetben, ha megoldott a folyamatos és valós idejű kommunikáció a felvevőegység és a feldolgozó egység között. Az is lehetséges, hogy egy hibrid rendszert fogunk tudni majd létrehozni: a képkészítés megfelelő optikával egy helyi kis számítógéppel történik, míg a feldolgozást egy távoli eszköz végzi az igényelt pontosság és gyorsaság függvényében. Minden az algoritmikus téren és optika terén fennálló lehetőségektől függ, ezért ezt az ajtót egyelőre nyitva szeretnénk hagyni, nem rögtön determinálni egy irányt, ami elképzelhető, hogy nem is lesz célravezető.

Mindezen kérdések megválaszolása előtt először a szoftveres megoldáson szerettünk volna előrelépést elérni, ugyanis nem voltunk megelégedve az eredménnyel. Az eljárás lassú, valós időben még egy nagy teljesítményű számítógépen sem futna el.

A fényképek elkészítéséhez egy DSLR gépet használtunk, mely kiküszöböli azt a hibát, hogy az optika nem alkalmas az emberi léptékben vett környezet fókusz szempontjából megfelelő észlelésére.

A szoftveres feldolgozást pedig egy viszonylag nagy teljesítményű laptop végzi, függetlenül a képkészítéstől. Ennek oka az, hogy szeretnénk ismereteket szeretni minden lehetséges megoldásról, hogy a végén mindent figyelembe véve meg tudjuk határozni a végső rendszer pontos felépítését, mely lehetőleg az optimalizált feladatmegoldást végzi.

8.1. DSLR fényképezőgép

A második szakasz egyik kulcsfontosságú célja az, hogy olyan képekkel dolgozzunk, mely hasonló környezetben készül, mint amiben a valós felhasználás során működni fog a rendszerünk.

Ennekokáért a felvételek elkészítését egy teljesen általánosnak mondható rendszerrel, egy képstabilizátor nélküli CANON EF-S 18-55 mm III-as objektívvel szerelt CANON 2000D kamerával végeztük. Az objektív legkevesebb 0.25 m-es fókusz távolságra képes, manuális fókusz természetesen rendelkezésre áll, igaz nem szoftveresen vezérelve, hanem manuálisan által beállítva. Ennek természetesen van egy olyan hátulütője, hogy nem lehet minden egyes alkalommal ugyanolyan fókusz távolságokat alkalmazni, viszont cserébe dinamikusan lehet a fókuszálást elvégezni a kísérletekre szabottan, illetve az EXIF adatokban

tárolt beállítások rendelkezésre állnak. A felvételek felbontása nem is igazából lényeges, hiszen a képeket az eljárás úgymint meghatározott szélességűre méretezi át.



8.1. ábra. ECANON 2000D gépház EF-S 18-55 mm objektívvel

8.2. Modell elrendezések

A kísérleti elrendezésekhez unokatestvéremtől kaptam ötletet: egyszer egy kenyérpapírt fotóztam le folyamatosan távolodó fókusszal. Az eredmény lenyűgöző volt, kíváncsi voltam, hogy vajon erre a példára is működne-e az eljárás. Ennek elméleti akadályát nem láttam, maximum az algoritmusunk robusztussága nem megfelelő egy ilyen aprólékos feladat elvégzéséhez.

A próbeképek egy népszerű társasjáték, a CATAN telepesei figuráiról, valamint a társasjáték dobozairól készültek kültéri körülmények között. A képkészítés során három felvételt készítettem, melyeknél az automatizálás hiányában saját megítélésem alapján állítottam be a fókusz távolságokat. Amint ezt már előzőleg is hangsúlyoztam ez a rész akkor lenne tökéletes, ha szoftveresen, hasonlóan az Arducam kamera modulhoz itt is be lehetne állítani fix fókuszértékeket és minden esetben azzal lehetne fotózni. Az algoritmikus elemzés tesztelése érdekében azonban ezzel a megoldással is megelégedtünk, ugyanis ahogy azt már az eredeti megközelítésnél is leírtam, ha az eljárás működik kisebb léptékekben, akkor azt már csak ki kell terjeszteni nagyobb léptékekre. Az optika tulajdonképpen

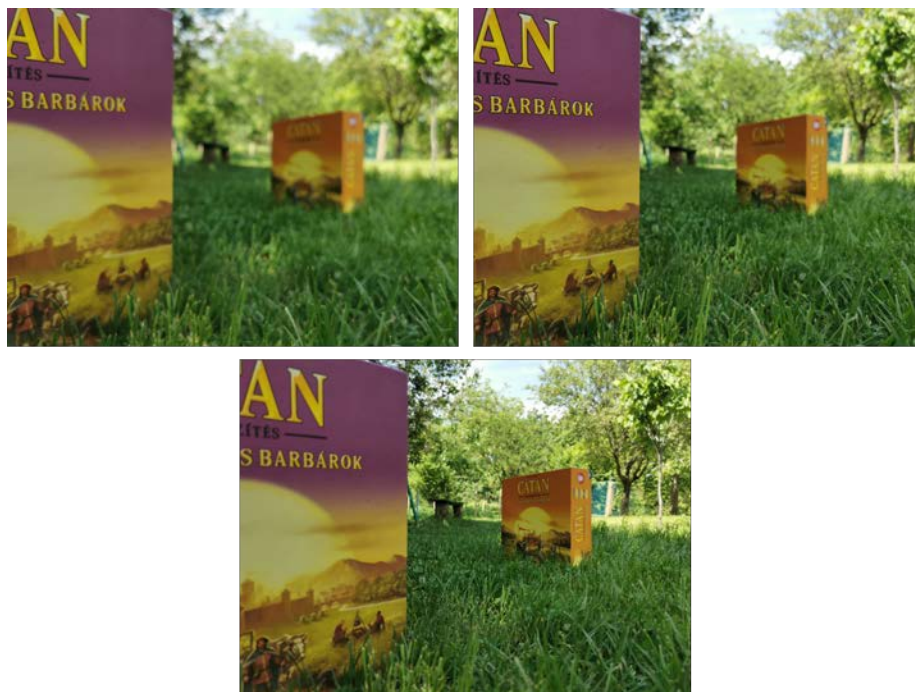
szabadon változtatható, ezért is vágunk bele egy olyan eszközzel történő képkészítésbe, mely esélytelen, hogy drónos felhasználás során szerepet kap.



8.2. ábra. Kenyérpapír távolodó fókusszal (saját kép, nagyon kis távolság)



8.3. ábra. CATAN telepesei bábuk távolodó fókusszal (saját kép, kis távolság)



8.4. ábra. CATAN telepesei dobozok távolódó fókusszal (saját kép, nagy távolság)

8.3. Programozási környezet

Az eddig elkészített algoritmikus produktum további felhasználása érdekében nem szerettük volna alapjaiban megváltoztatni a programozási környezetet, így maradtunk a Pythonnál, egészen pontosan a 3.9-es verziónál. A programozás már csak laptopos környezeten történt, Raspberry ebben a fázisban nem volt bevonva a munkába. Szövegszerkesztőnek a JetBrains PyCharm editorját használtam felhasználóbarátsága és logikus felépítése miatt.

8.4. Próba a meglévő algoritmus optimalizálására

Tény, hogy amikor elkészült az eredeti kontrasztszámító algoritmus, akkor az optimalizáltság rovására ment az, hogy egy mindenképpen jól működő eljárást hoztunk létre. A működőképesség sikerült, azonban ez sok esetben azt jelentette viszont, hogy egy bizonyos képszélesség felett már nagyon nagy lett a futási idő, valójában már 250 pixel széles képeken is a Raspberry-n futtatva több tíz másodperc volt.

Különböző programozási praktikákat alkalmazva sikerült elég nagy léptékben lerövidítenünk és egyszerűsíteni a kódokat. Több feltételvizsgálatot összevontunk, így a számuk és az általuk előidézett műveletek száma is nagyban lecsökkent. Az így előállt programpéldány felhasználásával a futási idő nagyjából 50%-kal csökkent. A tesztelést egy Dell G3 laptopon végeztük, mely a feladat szempontjából releváns specifikációs adatok közül 16 GB RAM memóriával és egy 8. generációs Intel Core i7-es, maximálisan 4,2 GHz-en működő processzorral szerelt eszköz.

Az eredeti, javítatlan algoritmus futási ideje 5 db 500 pixel széles képet használva ezen az eszközön átlagosan 5,96 s volt, míg a javított változat átlagosan 2.82 s-os futási idővel rendelkezett. Ez nagyjából 50-52%-os javulást jelent, azonban még egy meglehetősen erős számítógépnek is csaknem 3 másodpercbe telt elvégezni a feladatot. Ez az eredmény, valamint a tény, hogy az algoritmuson sokat már nem tudnánk finomítani, elvezetett minket arra a megállapításra, hogy valamilyen új megközelítést kell alkalmaznunk a kontrasztszámítás megvalósítására, esetlegesen egy jobb módszerrel a kiváltására.

Átfogó irodalomkutatásba kezdtünk, mely során eljutottunk az úgynevezett frekvenciatartomány-beli képfeldolgozási eszközökhöz. S. Dewangan és társa (Image smoothening and sharpening using Frequency Domain Filtering Technique, International Journal of Innovative Research and Growth, 2017) munkájából sok ötletet tudtam meríteni a szűrésekkel kapcsolatos frekvenciatartomány-beli feldolgozási eljárások tekintetében.

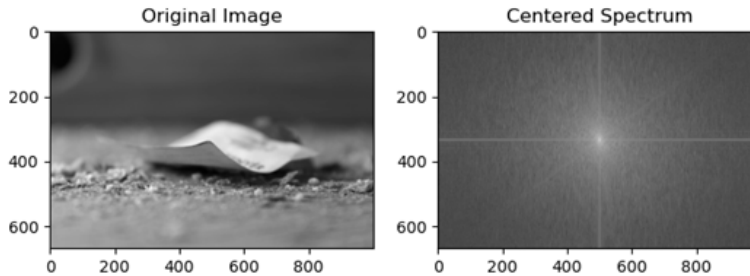
Emellett Kanjar De és társa munkája (Image Sharpness Measure for Blurred Images in Frequency Domain, Procedia Engineering, 2013) is a szűrés lehetőségei mélyebb megértéséhez segített hozzá, az elmosódottság és az élesség mértékének vizsgálata lehetséges összetevő lehet a még kifinomultabb eljárás megalkotásában.

8.5. Frekvenciatartomány-beli képfeldolgozás, szűrés

Általánosságban elmondható, hogy a Fourier-transzformáció a jelfeldolgozás egyik, ha nem a legfontosabb módszere, mely megadja a jeleknek a frekvenciatartomány-beli reprezentációját [10]. Ez a képfeldolgozás világában sincs másként, a Fourier-transzformáció segítségével felbonthatjuk a képeinket szinuszos és koszinuszos komponensekre (azok összegére). A transzformáció bemenetén egy kép van, az úgynevezett normál képtérben, a kimenetén pedig frekvenciatartomány-beli kép reprezentáció jelenik meg. A frekvenciatartomány-beli képnek minden egyes pontja egy bizonyos frekvenciájú összetevőjét ábrázolja a normál képtérben lévő képnek [1]. Mivel a probléma során digitális képekkel foglalkozunk, ezért a megfelelő eljárás a frekvenciatartomány-beli kép létrehozására a DFT, azaz a diszkrét Fourier-transzformáció a mi esetünkben.

$$F(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m, n] e^{-j2\pi(\frac{um}{M} + \frac{vn}{N})}$$

8.5. ábra. A diszkrét Fourier-transzformáció matematikai képlete



8.6. ábra. Kép és annak frekvenciatartomány-beli megfelelője

Ahogy az a 8.6. ábrán is látható, a spektrum ember számára értelmezhetetlennek tűnik, ugyanis komplex-exponenciális formában kapjuk meg az eredményt, ami nem ábrázolható értelmezhető módon. De ezzel a lépéssel, vagyis azzal, hogy áttértünk a frekvenciatartományba nagyon jó lehetőségek nyíltak meg előttünk a képfeldolgozás szempontjából. Ennek tulajdonképpen az oka az, hogy míg a valós képtérben a konvolúció nyújtotta lehetőségekkel tudjuk módosítani a rendszerünk bemenetére adott képet, a frekvenciatartományban ez a számításigényes művelet eltűnik, cserébe egyszerű szorzás - akár és, illetve vagy - művelettel a

feldolgozás sokkal egyszerűbbé és kevésbé számításigényessé, azaz gyorsabbá válik. Igaz, hogy a Fourier-transzformáció egy bonyolult és számításigényes lépés, azonban az a kapacitás, amit ezzel nyerni tudunk bőven túlmutat azon az időn, melyet elvesztünk az áttérésekkel.

A 8.6. ábra jobb oldali képe, tehát a spektrum a következőképpen épül fel: az alacsonyfrekvenciás komponensek a spektrum közepén helyezkednek el, míg a magas frekvenciás összetevők a spektrum szélénél [7]. Minél több gyors intenzitásváltozást tartalmaz a kép, annál több lesz a magasfrekvenciás összetevő a spektrumban. Ez érthető, hiszen a gyors intenzitásváltozások következtében a jel frekvenciája magas lesz. Ezek a gyors intenzitásváltozások pedig nem mások, mint az élek a képen.

Itt ér össze a probléma a frekvenciatartomány-beli képfeldolgozással: különböző szűrők felhasználásával ki lehet szűrni a képből a magas frekvenciás összetevőket, vagy fordítva. Ezáltal eltüntethetjük a képünkről az éleket, vagy akár pont fordítva: csak azokat hagyhatjuk meg és minden más részletet eltávolítunk a képből. Ez a módszer nagyon gyors, a másodperc töredéke alatt elvégezhető a szűrés a frekvenciatartományban.

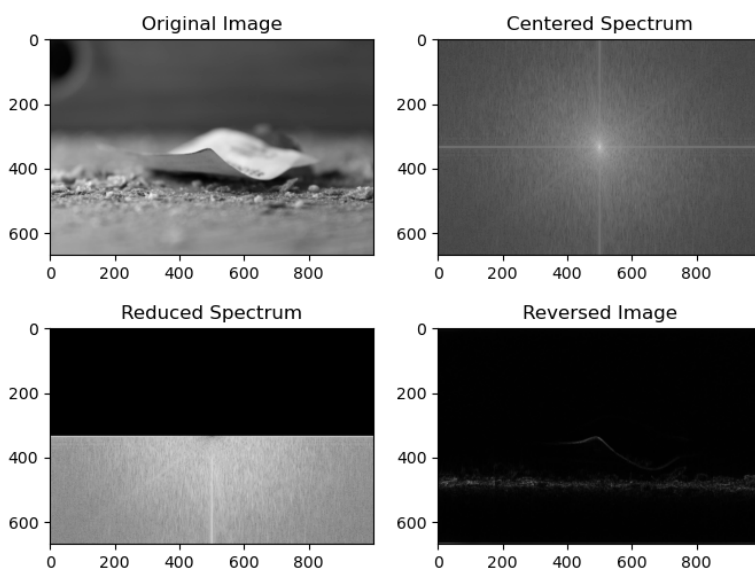
Érdekes tulajdonsága a spektrumnak, hogy vízszintesen és függőlegesen is szimmetrikus. Ennek következtében a képi adatok valójában négyszeresen el vannak tárolva a spektrumban, melynek csupán egy negyedének felhasználásával elméletileg kódolható minden változtatás a képeken.

A munkamenet tehát a következő: a képeinket a diszkrét Fourier-transzformációval átranzformáljuk a frekvenciatérbe, elvégezzük a megfelelő szűréseket, majd az inverz DFT-vel visszatranszformáljuk a spektrumból a valós, képtérbeni képet. A visszatranszformáláshoz elméletileg elegendő a spektrum egynegyedét felhasználni amellet, hogy ugyanazt az eredményt kapjuk, mintha az egészet használnánk, a gyakorlatban ez azonban nem teljesen így működik.

$$f(m, n) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F[u, v] e^{+j2\pi(\frac{um}{M} + \frac{vn}{N})}$$

8.7. ábra. Az inverz diszkrét Fourier-transzformáció matematikai képlete

Azzal, ha a spektrum közepét úgymond "kivágjuk", tehát egy olyan szűrőt alkalmazunk, amely levágja az alacsonyfrekvenciás komponenseket, eljutunk oda, hogy csak a magas intenzitáskülönbséggel rendelkező részei maradnak meg az eredeti képünknek, tehát az élek. Ez úgy végezhető el, hogy levágjuk a spektrum közepét egy r sugarú kör mentén. A szűrés paraméterezésével lehet változtatni, hogy milyen érzékeny lesz az eredményünk: minél erőteljesebb a szűrés, annál kevesebb komponens marad, így csak a legnagyobb frekvenciás összetevőket hagyjuk meg, ami csak a leghangsúlyosabb éleket fogja az eredeti képünkben megtartani. Kétélű fegyver ez: egyszerre kell a lehető legkevesebb zavaró részletet meghagynunk a képünkben, ezzel párhuzamosan pedig a legkevesebb releváns információt elveszítenünk. Az alábbi képen látható a felezett spektrum, majd egy szűrt változatából visszaállított, csak éleket tartalmazó képtér-beli eredménye a bemenetnek.



8.8. ábra. Bemeneti kép, annak frekvenciatartomány-béli megfelelője, felezett spektrum, valamint szűrt spektrumból való visszaállított kép

A kimeneti képek felhasználásával ebben az esetben is létrehoztuk az előző verzióhoz hasonlóan a mélységi térképeket.

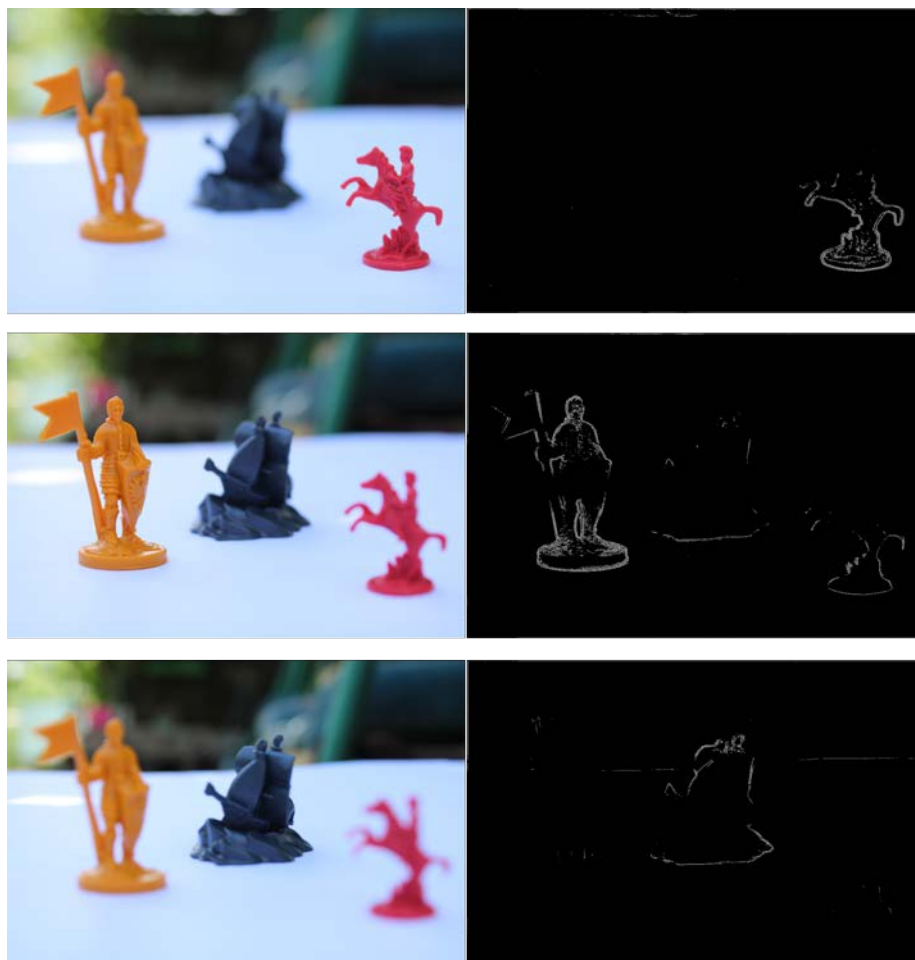
9. fejezet

Új megközelítés eredményei

Az új megközelítéssel, azaz a frekvenciatartomány-beli manipulációval számos kísérletet hajtottunk végre. Több algoritmus is elkészült, ezek közül a három legjobban működő és ezáltal legjobb eredményekkel kecsegtetőt szeretném bemutatni.

Az első esetben az egész spektrum felhasználásával végezzük el a szűrést és a visszaállítást, a második esetben a spektrum függőleges felezéséből előállt részspektrumot szűrjük, majd abból állítjuk vissza a képtér-beli képet, míg a harmadik esetben a spektrum vízszintes felezésével végezzük el ugyanazt, mint a második esetben.

A bemutatott eredményeket vegyesen 500 és 1000 pixel széles képek feldolgozásával értük el, minden egyes alkalommal 3-3 képet elemeztünk. Az első esetben, tehát a kis bábukát bemutató sorozatnál az előbb felsorolt első programverziót használtuk, a második, tehát a dobozokat bemutató sorozatnál a második említett programverziót, a harmadik, tehát a kenyérpapíros sorozatnál pedig a harmadik programverziót használtuk.



9.1. ábra. Első összeállítás és a hozzá tartozó kimeneti képek

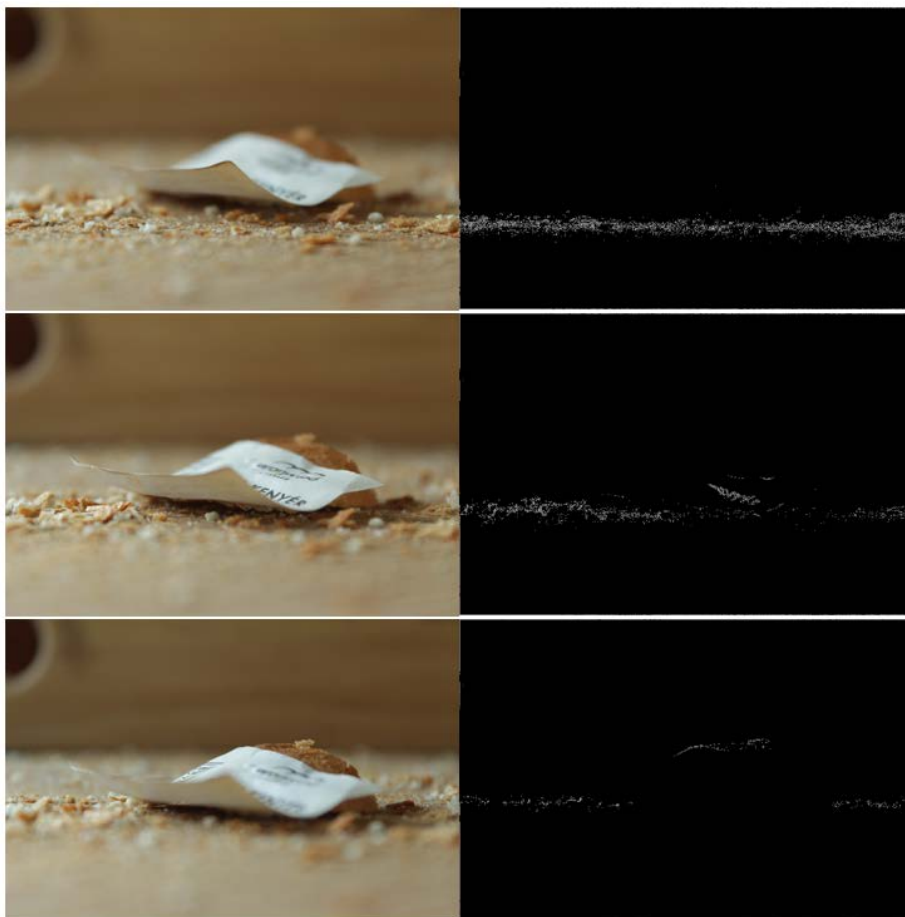
Ezen az összeállításon nagyon jól látszik, hogy a fókusz távolodásával a soron következő bábuk körvonala fokozatosan kirajzolódik, miközben az előbbi síkokban lévő figurák körvonalai gyengülnek, illetve el is tűnnek. Az új megközelítés kitűnően működik ennek a példának a tekintetében, a paraméterezés és a manuális fókuszbeállítás is nagyon jól sikerült.



9.2. ábra. Második összeállítás és a hozzá tartozó kimeneti képek

A második összeállítás tekintetében kettős az eredmény. Amiért mégis be szeretném mutatni az az, hogy megmutassam, mennyire fontos a megfelelő szűrési paraméterezés kiválasztása. Amellett, hogy a harmadik kép fókusza egy kicsit távoli lett, ami mellett a mélységélesség tartománya is nagyon kinyílt, a paraméterezése is túl finom lett a szűrésnek. Ez azt jelenti, hogy túl sok él maradt meg a kimeneti képen amiatt, mert nem volt eléggé szigorú a szűrés, és a jó megoldáshoz képest sok alacsonyabb frekvenciájú összetevő maradt a spektrumban. Mindemellett nagyon fontos dolgora hívja fel a legelső kép a figyelmet: a sima felületek érzékelésére. A textúra nélküli felületeken nincsenek kontrasztos elemek, így nehezebb az éles vagy homályos állapotokat megkülönböztetni. A homogén felületek vizsgálata egy sarkalatos pontja kell, hogy legyen a követ-

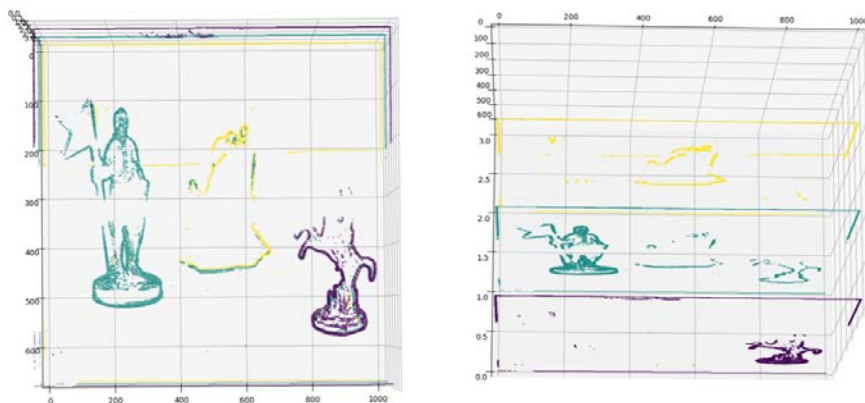
kező fejlesztési időszaknak. Erre a problémára az aktív rendszerek általában strukturált fényt alkalmaznak, amivel textúrát vetítenek az adott felületekre. A mi passzív megoldási elképzelésünkben más megoldást kell találnunk.



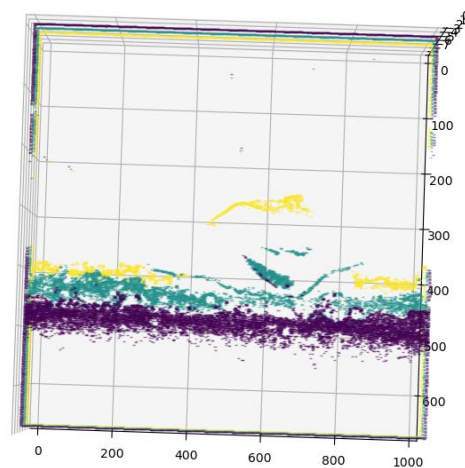
9.3. ábra. Harmadik összeállítás és a hozzá tartozó kimeneti képek

Ebben az esetben, amikor a sima felületen olyan elemek vannak, amiket az éldetektáló algoritmus felismer, akkor nincs olyan probléma, mint az előző esetben. Ennek hiányában azonban ebben az esetben is problémát okozna az asztal detektálása. Ami viszont abból a szempontból jelentene problémát, hogy a további algoritmikus feldolgozás során akár rossz irányváltoztatási döntést is hozhatna az eljárás.

A mélységi térképeken is nagyon jól látszik az eljárás működése, valamint működőképessége. Nagyon jól le lehet követni a tér különböző síkjaiban dektálható hozzávetőleges kiterjedéseket és pozíciókat.

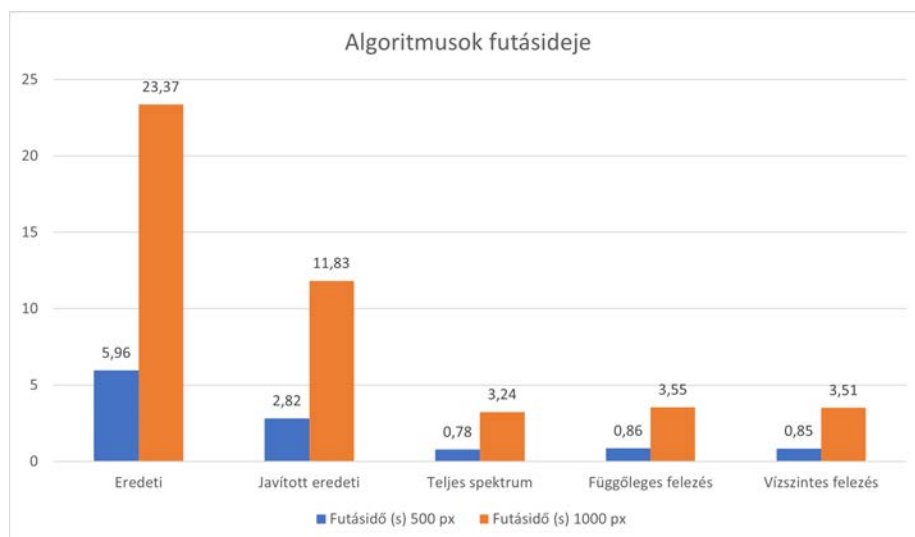


9.4. ábra. Mélységi térkép a figurákhoz



9.5. ábra. A kenyérpapír mélységi térképe szemből - a színek a különböző mélységeket kódolják

Futási időket tekintve is nagyon nagy előrelépést tettünk. Az alábbiakban egy grafikont fogok mutatni, melyeken az 500 pixel és az 1000 pixel széles bemeneti képek feldolgozásához szükséges futásidőt lehet megfigyelni az eredeti megközelítés, annak javított változata, valamint a három új programverzió tekintetében.



9.6. ábra. Futásidő 500 px és 1000 px széles bemeneti képek esetében a programverziók szerint

Általánosságban megfigyelhető az a jelenség, hogy a kétszer olyan széles és magas kép feldolgozásához négyszer annyi időre van szükség, de ez ugye a képek oldal és terület négyzetes összefüggéséből adódik, ez a jelenség minden egyes program esetében felismerhető. Látható, hogy mindkét bemeneti méret esetében a sima kontrasztszámító algoritmusok meglehetősen lassúak, hiába esetek át több iteráció javításon. Megfigyelhető továbbá az, hogy azzal tulajdonképpen nem lehet érdemben javítani az algoritmus futásidején, ha nem a teljes spektrumot használjuk fel, hanem csak a feléből állítjuk vissza a kimenetre szánt képet. A maszkolás több számítási kapacitást foglal le, mint a teljes spektrum használata, ami összességében nagyobb hatással van a futásidőre, mintha nem is történne meg. Minden esetre egy érdekes próbát megért a megközelítés megvizsgálása.

10. fejezet

Összegzés

Összességében elmondható, hogy az általunk alkalmazott új megközelítés eredményesnek mondható, a feladatot ugyanolyan, ha nem jobb minőségben megtudja oldani, mint az eredeti megközelítés szerinti kontrasztszámító algoritmus. Ami viszont a két eljárás között a legnagyobb különbség, ami miatt fontos volt az új megközelítés, az a futásidő. Az új eljárásokkal csaknem 87%-os futásidő csökkenést érhattünk el az első megközelítéshez képest, annak optimalizált változatához képest is csaknem 73%-os csökkenést tapasztalhatunk az átlagos futásidőkben. Ezekkel az értékekkel, azaz a három 500 pixeles kép feldolgozásához szükséges időértékekkel megvalósíthatjuk a valósídejűség, vagy legalább a közel-valósídejűség igényét.

Az optika szempontjából átfogó kutatást kell végeznünk a különböző dinamikus fókuszálási lehetőségekkel rendelkező eszközök körében. Elképzelhető, hogy találunk olyan eszközt, mely hibátlanul el tudja végezni ezt a feladatot, lehetőleg szoftveresen vezérelve. Abban az esetben, ha nem találunk megfelelő optikát, akkor a drón kameráját is fel lehet használni, viszont akkor az elérhető repülési sebesség nagyban csökkenne. Eszerint a koncepció szerint a drón egy állandó, de lassú sebességgel repülne, mely imitálná a fókuszállítást. A képfeldolgozás folyamatosan pipeline-szerűen működne. Itt a látószög torzulásokat lehet szükséges kezelni, de ez, mivel nem pontos szkennelés a cél, vélhetőleg kevésbé zavaró hatás lesz.

Számítási kapacitás tekintetében a valós idejű rendszer megalkotásához vagy pontosabb térbeli adatok kinyeréséhez szükséges lehet egy erős számítógép, az eredeti megközelítésben felvázolt miniPC-s irány valószínűleg még az új módszerrel is a valósídejűség határán billeg. A komolyabb számítási kapacitás bevonása megoldható úgy is, hogy a drón folyamatosan streameli a felvételeket

egy távoli feldolgozóegységhez, amely elvégzi a folyamatot, majd továbbítja a drón felé az irányításához szükséges paramétereket. Ez a mai, kiforrott hálózati technológia (5G) felhasználásával nem jelentene nagy akadályt, a rendszer gyorsasága nem szenvedne csorbát. A külső egység felhasználása új lehetőségeket is felvillant. Ez a megközelítés azonban egy másodlagos, párhuzamos irány lenne, azonban elsődlegesen egy gyors és jól becslő megoldás fejlesztése a cél, lehetőleg ragaszkodva az eredetileg felvázolt egyszerű megvalósításhoz.

Jövőbeni terveink között szerepel egy olyan, mesterséges intelligencia alapú gépi tanuló eljárás megalkotása, mely a kimeneti kontrasztképek elemzésével egy iránybecslést ad a drónnak, hogy az előtte álló térben milyen pályán lenne érdemes mozognia annak érdekében, hogy ne ütközzön bele semmilyen tárgyba.

11. fejezet

Köszönetnyilvánítás

Mindenek előtt szeretném megköszönni témavezetőmnek és mentoromnak, Dr. Várady Gézának mind emberi, mind szakmai segítségét, mely nélkül a dolgozat nem készülhetett volna el. Támogatásával és iránymutatásával segítette a munkát, mely számomra nagy szakmai fejlődést jelent.

Szeretném megköszönni továbbá a Pécsi Tudományegyetem Műszaki és Informatikai Karának, azon belül is az Informatika és Villamos Intézetnek, valamint a Műszaki Informatika Tanszéknek, hogy minden tárgyi, lehetőségi támogatást megadtak a projektben való munkálkodásom során.

A projekt a Pécsi Tudományegyetem Kriszbacher Ildikó Tehetséggondozó Ösztöndíj támogatásával valósul meg.

Irodalom

- [1] *Image Transforms - Fourier Transform*. URL: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm>.
- [2] *Képfeldolgozás 4. előadás Konvolúció, élkereső eljárások. Fotometriai alapok*. URL: <https://docplayer.hu/108962957-Kepfeldolgozas-4-eloadas-konvolucio-elkereso-eljarasok-fotometriai-alapok-benedek-csaba-es-sziranyi-tamas.html>.
- [3] *Raspberry Pi*. URL: https://en.wikipedia.org/wiki/Raspberry_Pi#Model_B.
- [4] *Raspberry Pi 3 Model B*. URL: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>.
- [5] *Raspberry Pi Camera Module 2*. URL: <https://www.raspberrypi.com/products/camera-module-v2/>.
- [6] *Raspberry Pi OS*. URL: https://en.wikipedia.org/wiki/Raspberry_Pi_OS.
- [7] *Spectrum of images*. URL: <https://vincmazet.github.io/bip/filtering/fourier.html>.
- [8] *Understanding Depth of Field*. URL: <https://photographylife.com/what-is-depth-of-field>.
- [9] *What is sharpness? - Outdoor Photographer*. URL: <https://www.outdoorphotographer.com/tips-techniques/nature-landscapes/what-is-sharpness/>.
- [10] Sári Zoltán. “Bevezetés a számítógépes jelfeldolgozásba II.” (2015).