

GPU-alapú hatékony kép- és információfeldolgozási módszerek

Tézisfüzet a PhD disszertációhoz

Gelencsér-Horváth Anna

Témavezetők:

Dr. Cserey György

Pázmány Péter Katolikus Egyetem

Dr. Karacs Kristóf

Pázmány Péter Katolikus Egyetem



Pázmány Péter Katolikus Egyetem

Információs Technológiai Kar

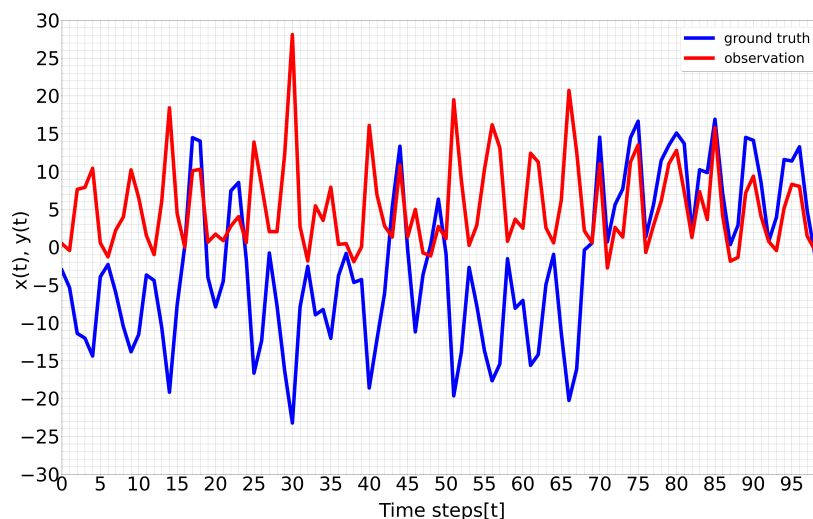
Roska Tamás Műszaki és Természettudományi Doktori Iskola

Budapest, 2023

1. Bevezetés

A vizuális érzékelés jelentős mennyiségű információt nyújt környezetünkről, ezért számtalan kutatás vizsgálja hogyan lehet számítógéppel az emberi érzékeléshez hasonló módon feldolgozni a vizuális információkat. A gépi látás algoritmusainak sebessége kritikus fontosságú számos alkalmazásban, többek között olyankor, amikor valós idejű információfeldolgozásra van szükség, például navigációs rendszerekben.

A Particle Filter (PF) algoritmusok az 1990-es évekre vezethetők vissza [1], és robusztus módszernek tekintjük őket olyan nemlineáris állapotterrel rendelkező modellek állapotainak becslésére, amelyeknek a megfigyelése nem Gauss-eloszlású zajt tartalmaz [2]. A részecskeszűrő (Particle Filter) egyszerre tartozik a szekvenciális Monte-Carlo módszerek (SMCM) algoritmuscsaládba, és tekinthető a Kálmán-szűrő [3] kiterjesztésének is. Gyakran idősorok predikciójára használják, például a pénzügyi matematikában [4, 5] vagy pozíció követésében, ugyanakkor messze túlmutat az egy dimenziós adatokon, még ha a kiterjesztés magasabb dimenzióra kihívást is jelent [6]. Sok megközelítésben alkalmazzák, ahol a bemenet kép vagy képsorozat, ideértve az képrekonstrukciót [7], az objektumdetektálást [8], a navigációt [9], a szegmentálást [10, 11], a kontúrfelismerést [12], és az objektum követést amikor az objektumok gyakran kitakarják egymást [13]. Egy széleskörűen, algoritmusok összevetésére használt modellhez (ún. *benchmark* modellhez) tartozó valós (rejtett) idősor és annak zajos megfigyelése [2] látható az 1. ábrán.



1. ábra. Egy adatsor (valós, rejtett állapotok kékkel) és zajos megfigyelése (pirossal) egy, a részecskeszűrők kiértékeléséhez gyakran használt modellhez [2].

A PF algoritmus számításigényes és hosszú futási idővel jár az újramintavételezés lépése miatt, mivel az a teljes kumulatív eloszlásra épül [14]. Széleskörűen vizsgált terület az, hogy milyen módon lehet párhuzamos architektúrára implementálni úgy, hogy hatékony és gyors futást érjünk el miközben a predikció minősége nem romlik [15–24].

A grafikus kártyák (GPU) vonzó implementációs platformot jelentenek, mivel a számítási teljesítményük magas, az árak viszont relatíve alacsony. Azonban ezt a nagymértékű számítási kapacitást csak az architektúra jellegéhez illeszkedő algoritmusok tudják igazán kihasználni. Ugyanakkor az algoritmus párhuzamosításánál figyelembe kell vennünk, hogy a Particle Filter újramintavételezésének módja nagy hatással van a predikció minőségére. A [25] kutatás összehasonlító táblázata jól bemutatja, hogy a különböző újramintavételezési stratégiák kompromisszumot igényelnek a sebességben vagy a predikció minőségében. Egy olyan PF implementáció, amely az újramintavételezés alatt meg tudja őrizni a lokális összeköttetéseket és az információcserét a részecskék között, lehetővé tehet jelentős gyorsulást anélkül, hogy a hiba mértéke emelkedne. A kutatásom során arra kerestem megoldást, hogy újratervezzem a particle filter algoritmust úgy, hogy az elérhető GPU architektúrákhoz a lehető legjobban illeszkedjen [A1, A3].

A kutatásom célja az volt, hogy a szekvenciális fontossági mintavételezést (SIR) [26] alkalmazó particle filtert újszerű megközelítésbe helyezzem, ami által a GPU implementáció az eredeti algoritmus minőségét megőrizve biztosít predikciót a rejtett állapotokra. A Celluláris Particle Filter (CPF) [27], vagyis celluláris részecskeszűrő ígéretes módszert kínál a mintavételezés jelentette kihívásra azzal, hogy a Celluláris Neurális Hálózatok (CNN) architektúrája alapján a PF részecskéinek logikai reprezentációját egy két dimenziós, lokális összeköttetésekkel rendelkező hálóként definiálja [28]. A szomszédság CNN-jellegű definiálásának köszönhetően a CPF lehetséges megoldást jelent a részecskék közötti lokális információmegosztás csökkenésének problémájára, amelyre a figyelmet a [17] cikk felhívja. Az állapotok becslésének minősége hasonló a szekvenciális implementációéhoz, ugyanakkor ez a reprezentáció nem ideális a GPU architektúra szempontjából. Ezért az volt a célom, hogy újratervezzem az algoritmust olyan módon, hogy illeszkedve a GPU architektúra sajátosságaihoz ki tudjam használni a számítási képességét a gyors futási sebesség érdekében.

A mérnöki feladatok tekintetében az algoritmusok hatékonyságának növelése mellett nem elhanyagolható tényező a végrehajtáshoz a teljes szükséges idő, amelynek része az adatgyűjtés és adat-előkészítés is. Az egyes algoritmikus komponensek (mint például objektumkövetés esetén a részecskeszűrő) optimalizációja mellett számolnunk

kell azzal, hogyan tudunk megfelelő bemeneti adatokat biztosítani az algoritmusok számára. Általában ezt az adatot a bementi kép szegmentációval kapjuk. A szegmentáció, amely során az azonos objektumokhoz tartozó pixeleket azonos címkével látjuk el, a képi értés és vizuális információfeldolgozás egyik stratégiája. Komoly kihívást jelent olyan esetekben megtalálni a kontúrokat és csupán a valóban összetartozó képterületeket összevonni, amikor szomszédos objektumok nagyon hasonló jellegűek. Ezért azt vizsgáltam, hogy Mean Shift alapú szegmentációs eljárással [29] túlszegmentált képen a régiók szín alapú hasonlósága mellett, az objektumok formájából adódó, a régiók geometriai jellegzetességeit hogyan lehet felhasználni a kontúr detekció során [A4, A5].

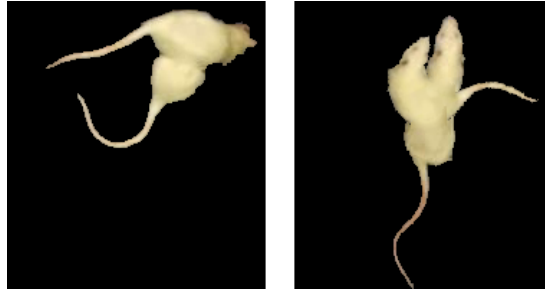
Az elmúlt évtizedben a szegmentáció területén, a képfeldolgozás és gépi tanulás módszerei mellett, a technológiai fejlődésnek köszönhetően a mély tanuláson alapuló megközelítések nagy mértékben elterjedtek [30]. A kutatásomat 2015-ben hat évre felfüggesztettem, és 2021-ben a szomszédos, nagyon hasonló területek szegmentációjának kérdését a mély neurális hálókra építve kezdtem vizsgálni. Ehhez a motivációt egy biológiai kutatásban felmerült nagyon hasonló, jelöletlen patkányok identifikációs címketartó követésének megoldatlan problémája adta [A2]. A mély neuronhálók rendkívül hatékony eszközei a vizuális információ automatikus kinyerésének. Amennyiben elérhető a megfelelő tanítóadat, a végső modell olyan magas szintű reprezentációt is megtanulhat, ami által hasonló feldolgozásra képes, mint a humán agy. A hagyományos algoritmusokhoz képest a mély neuronhálóknak címkézett, ún. annotált tanítóadatra van szükségük, amelynek mennyisége a feladat jellegétől függ. Ezért a mély neuronhálós modellek esetében a tanítóadat előkészítésére és annotálására fordított időre gondolhatunk úgy, mint egy kompromisszum tényezőre a predikció minősége szempontjából. Komplex jelenetek esetén a megfelelő tanító adathalmaz összeállítása (amely alapján egy modell képes megtanulni az információ kinyerését) meghaladhatja a számítógépes vagy emberi erőforrások lehetőségeit, sőt, még amikor az adat rendelkezésre is áll, az annotáció jelentős időt igényelhet. Ezért, azt mondhatjuk, hogy hatékonyság szempontjából nem elég a predikció minőségét és a futási időt figyelembe venni, hanem az ehhez szükséges tanítóadat gyűjtése, előkészítése és annotációja is jelentős tényező egy háló szempontjából. Az ún. *end-to-end* hálók [31, 32] esetében egyetlen, akár komplex struktúra tanítása biztosítja, hogy egy nem előfeldolgozott bemeneti adatra az elvárt kimenetet kapjuk. A feladat és feladathoz tartozó bemeneti képek jellegének függvényében, ilyen hálót megadni komoly kihívás lehet. A *Composite AI* [33, 34] olyan megközelítés, ami az ún. *end-to-end* mély neuronhálók limitációin túl tud lépni. Összetett feladatok esetén a mély neuronhálók, a gépi tanulás, a tradicionális képfeldolgozás, magasabb szintű logika, sőt

még az alkalmazási területre vonatkozó tudás is használható ahhoz, hogy egy magas minőséget biztosító lépéssorozatot állítsunk össze a a kihívást jelentő képekhez, ahol a kihívás oka számos jellegű lehet, mint például az ismeretlen objektumok jelenléte, kitakarások, zajok vagy az objektumok hasonlósága. Jelentős mennyiségű időt tudunk megtakarítani, ha olyan adaton tanítjuk a hálót, amelyhez már a létrehozáskor előállítjuk a helyes címkézést/annotációt is. Ugyanakkor ilyen esetben kritikusan fontos és jelentős kihívás, hogy a szintetikusan létrehozott adat a megfelelő metrikák mentén kellően hasonló legyen a ténylegesen várható bemenetekhez, hogy elkerüljük a túlillesztést a létrehozás során létrejött artifaktumokra, zajokra.

Az volt a célom, hogy egy olyan módszert hozzak létre, amely magas minőségű szegmentációt prediktál rendkívül hasonló példányokra humán felügyelet (azaz annotáció) nélküli, tehát teljes mértékben szintetikusan létrehozott tanítóadatokon és automatikusan annotált adatokon, hogy az emberi idő- és figyelemráfordítást lehetőség szerint a legkisebbre csökkenthessük. A kutatásomhoz a motivációt az adta, hogy két egyforma kinézetű patkány automatikus követését biztosítani lehessen, az eredeti ID címkék megőrzése mellett akkor is, ha időszakosan, kisebb vagy nagyobb részben kitakarják egymást. A célom az volt, hogy egy olyan összetett algoritmust hozzak létre, amely egy régió alapú, automatikusan annotált adatokon tanult detekciós módszert [35] és átfedő példányok kontúrjainak detekcióját ötvözi, így olyan magas minőségű szegmentációt biztosít, amelyre az [36] cikkben bemutatott képkockák közti összekapcsolási eljárást alkalmazva megbízható ID-tartó objektumkövetést biztosíthatunk.

Az éldetekciós hálók tanítását szintetikus adatokkal oldottam meg. A tanítóadatokat teljes mértékben automatikusan generáltam (olyan képkockákra építve, amelyeken az állatok nem átfedő pozíciókban vannak). A humán annotáció nélkül létrehozott tanítóadatokon tanítottam az éldetekciós hálót a statikus háttér előtt tartózkodó, átfedő pózban lévő állatok közti kontúr detektálására. A 2. ábrán látható egy valós, illetve egy generált képkocka.

Az éldetekció kimeneteként kapott belső kontúr egyes képeken nem volt folytonos, ezért a magasabb szegmentációs minőség érdekében olyan generatív módszert kerestem, amely a detektált él jellegéhez leginkább illeszkedik, ezáltal ki tudja egészíteni a az elválasztó kontúrt.



2. ábra. Egy valós (balra) és egy szintetikusán létrehozott (jobbra) képkocka két egyforma kinézetű, jelöletlen, átfedő patkányról háttér nélkül. Az egy patkányhoz tartozó és a két patkány által közösen alkotott terület hasonlósága miatt az éldetekciós algoritmusoknak kihívást jelent a valós képkockákon detektálni a két állat közti kontúrt.

2. Vizsgálati módszerek

A részecskeszűrő (PF) GPU adaptációját az elosztott és lokális részecskeszűrők elérhető szakirodalmára [15–24, 27] építve terveztem meg a kutatásom során 2013-ban a következő fő szempontok alapján. Az egyik szempont a részecskék közti információmegosztás jellege volt, hogy a becslés minősége összemérhető legyen az eredeti, szekvenciálisan, a teljes kumulatív eloszlás alapján mintavételező algoritmus becslésének minőségével. A másik szempont a GPU memóriatípusainak jellegzetességei voltak, hogy az időigényes CPU-GPU adatküldést minimalizálni lehessen, és a kernelek futási ideje, a GPU eszközön történő szinkronizáció, valamint a véletlenszámgenerálás módszere a leghatékonyabb legyen.

Habár a CUDA SDK-ban megtalálható NVIDIA Mersenne Twister ígéretes megoldásnak látszik, a véletlenszámok eloszlása alacsony populáció esetén (néhány száz, sőt néhány ezer számra) nem felelt meg az egyenletes eloszlásnak. Ezért a lehetőségeket megvizsgálva két különböző megoldást javasoltam az egyenletes eloszlású véletlenszámok generálásának problémájára.

Két széles körben használt referencia modellt (ún. benchmark modellt) használtam arra, hogy a legkorszerűbb (a kutatás idejében elérhető) módszerekkel összehasonlítsam a módszerem predikciójának minőségét és a futási időt. Az egyik egy nem-autonóm, nemlineáris modell folytonos állapotterrel, amelyet a [2, 37–39] cikkekben találunk meg, a másik egy állapotátmenet modell (BOT modell) [40], amelyet a [22, 23] cikkekben használtak referencia modellként. A mérések egy NVIDIA GeForce GTX 550 Ti GPU, 1 GB GDDR memóriával rendelkező eszközön futottak, 2.1-es számítási kapacitás, a 4.1 CUDA toolkit és 295.49 driver verzió mellett.

A szegmentáció minőségének javítása olyan képkockákon, amelyeken rendkívül

hasonló példányok nagyban átfednek egymással, komoly kihívást jelent, amelynek megoldásához a 1. fejezetben kifejtett "összetett AI" megközelítést alkalmaztam. A különböző AI eszközök, mint modulok integrálásával létrehozott összetett lépéssor predikciója pontosabb lehet, mint egy end-to-end módon dolgozó megközelítésé, sőt, az egyes modulok külön-külön fejleszthetőek és optimalizálhatóak, hogy a kimenetük a következő modul számára a leginkább megfelelő legyen.

Egyedek szegmentációjának javításához a gépi tanulás és a mély neuronhálós módszerek mellett a lépéssor megfelelő pontjain fel tudjuk használni a képi kontextust is, mint magasabb szintű logikai összefüggéseket. Két olyan algoritmus kombinálása, amelyek felfoghatóak egymás duálisaként [41] lehetővé teszi az egymást kiegészítő információk integrációját, ami az egyes algoritmusok korlátjain túlmutató eredményt hozhat létre. Ilyen algoritmuspár például a régió alapú detekció és az éldetekció.

A szakirodalom alapján a mélyháló alapú módszerek ígéretesebbek komplex képek esetén, mint a hagyományos éldetekciós technikák. Ezért a legkorszerűbb (*state-of-the-art*) hálókat [42–45] vizsgáltam meg az éldetekciós eljárásom tervezéséhez. A hálótanításnál azt vettem figyelembe, hogy milyen minőségű predikciót várhatunk, és hogy mennyi időre van szükség az annotált tanítóadat előállításához. Az egyik célom az volt, hogy a monoton és időigényes emberi annotációt kiváltsam, ezért a [36] cikkben bemutatott módszerből kiindulva, a dolgozatban bemutatott transzformációk segítségével állítottam elő a tényleges bemenethez hasonló jellegű tanítóadatokat. Az adathalmaz, amivel dolgoztam az Eötvös Loránd Tudományegyetem Természettudományi Kar Élettani és Neurobiológiai Tanszékén készült 1280x720 felbontású, 25 fps mellett 20 perces felvétel képkockáinak összességéből állt össze. A bemeneti képek azonosak a [36] cikkben használt képkockákkal, amelyeket a videóból képkockánként, a patkányok dobozának területére vágva, azaz 640x420 pixel felbontásban mentettek *png* formátumban. A tanítóadatokat teljesen szintetikus módon készítettem az II.1. és II.2. tézisben leírt módon. Ehhez a videó első 10 percének képkockáit használtam (tehát az első 15 000 képkockát). Az éldetekciós [42] és az élkiegészítő [44] mélyhálók tanításához azokat a képkockákat használtam fel (összesen 9 233 darabot), amelyeken egy előzetes, hisztogram és módusz alapú módszerrel megalkotott háttér-előtér detekció alapján két olyan előtér objektum található, amelyek megfeleltethetőek a példányoknak, nem átfedő pozícióban. A CycleGAN tanításához 2 200 képkockát választottam ki véletlenszerűen a videó fennmaradó, átfedő példányokat tartalmazó képkockái közül. A tanító adathalmaz méretét a CycleGAN cikkben bemutatott tanítóadatokkal azonos nagyságrendi méretben választottam meg, amelyet aztán a tanítás eredményét nézve megfelelőnek ítélt meg.

A dolgozatomban bemutatott módszerrel kapott szegmentáció minőségének kiértékeléséhez 18 egyenként 200 képkockából álló képsorozatot használtam. Ezek olyan képkockák, amelyek megfigyelés alapján lettek úgy kiválasztva, hogy kihívást jelentő átfedő pozíciókat tartalmazzanak. Összesen 1 669 olyan képet tartalmazott az adathalmaz, amelyben az előtérsegmentáció alapján nem található két különálló objektum. A tanításhoz és teszteléshez használt képek száma a 1. táblázatban látható a különböző hálókhöz. Az volt a célom, hogy a humán annotáció és figyelem idejét csökkentsem, ezért az időre vonatkozó mérések jelzés értékűek. A tanítási idők a következők voltak: az éldetekció háló esetében (DexiNed) 6 óra (15 epoch), a CycleGAN esetében 1 óra (10 epoch), az élkiegészítő háló (EdgeConnect) 16 óra (20 epoch)¹.

1. táblázat. A különböző mély hálók tanító és teszt képeinek száma. A tanítóhalmaz képeinek 90%-át tanításra, 10%-át validációra használtam.

	Képek száma			Teszt
	Tanítás			
	RGB/aRGB	aGT(élkép)	Nem zárt kontúr	
DexiNed	92 330	92 330		3 600
CycleGAN		2 200	2 200	200
EdgeConnect	92 330	92 330	92 3300	3 600

Az annotációhoz az adatbázist négyfelé osztottuk, minden részt egy fő annotált, egy másik validálta az annotációt.

A dolgozatban bemutatott módszeremet a [35] cikk objektumkövető eljárásával kombináltam, majd a 3 600 annotált képkockából álló videón kiértékeltem és összevettem a legkorszerűbb ID követő módszerekkel [46–49]. Habár az elsődleges cél az volt, hogy az ID címkék felcserélését elkerüljük, a címketartó követés mellett a szegmentációs pontosságot is összehasonlítottam (amelyik módszernél erre vonatkozó adat elérhető volt). A mérésekhez és a futtatásokhoz két szervert használtam². Az egyik szerver specifikációi: két NVIDIA TITAN RTX @ 24 GB videó memória, és AMD Ryzen TR2920X CPU @ 64 GB RAM. A másik szerver specifikációi: két Nvidia RTX 3090 GPU @ 24 GB videó memória, és AMD Ryzen TR1920X CPU @ 64 GB RAM.

Mindkét szerveren Ubuntu 18.04 operációs rendszer fut, a kódok Apptainer [50] virtuális környezetben futottak Pytorch 1.4 illetve 1.9 verzió mellett az éldetekció illetve

¹Méréshez használt architektúra: Nvidia TITAN RTX GPUs @ 24 GB videó memóriával, AMD Ryzen TR2920X CPU @ 64 GB RAM

²Eötvös Loránd Tudományegyetem, Informatikai Kar, Mesterséges Intelligencia Tanszék, Neurális Információfeldolgozási Csoport *nipg8* és *nipg10* szerverei

az élkiegészítés hálóihoz, OpenCV 4.1.1 verzió mellett (a pontos verzió- és csomagkövetelmények, valamint a környezetet létrehozó fájl megtalálható a dolgozathoz csatolt github projektkönyvtárban).

3. Új tudományos eredmények

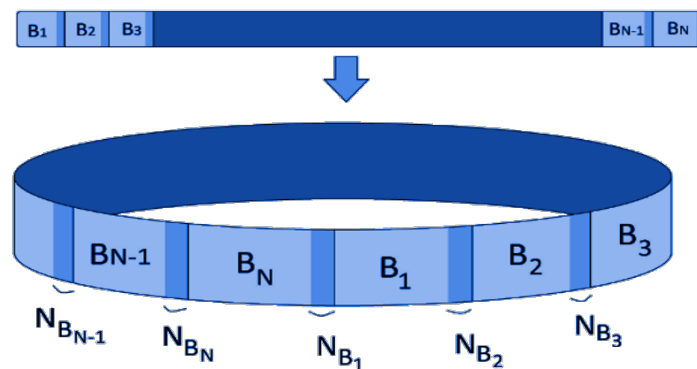
I. TÉZISCSOPORT: Részecskeszűrő (Particle Filter) algoritmus leképezése a GPU architektúrára a lokális összeköttetéseket megtartó módon az információveszteség megelőzésére.

A részecskeszűrők a Kálmán-szűrők kiterjesztéseként is tekinthetők, ezért számos területen, mint például a képfeldolgozás, robotika vagy tőzsdei előrejelzés területén a rejtett Markov dinamikát követő modellek esetében alkalmas megoldást jelenthetnek. A GPU architektúrák hatékony implementációs felületet kínálnak a nagy számítási igényű alkalmazások számára. Ugyanakkor a részecskeszűrők eredeti algoritmusát a párhuzamosításra alkalmatlannak tartották jelentős mennyiségű információveszteség nélkül, ami jelentős korlátot jelentett a gyorsulás szempontjából. A [27] cikkben bemutatott módszer a programozható logikai kapukat tartalmazó hálózatok (Field Programmable Gate Array, FPGA) vagy Celluláris Neurális Hálózatok (CNN) esetében biztosított lokális összeköttetések alapján javasol megoldást a párhuzamosítás problémájára. Míg más módszerek esetében a párhuzamos vagy elosztott szűrők a csökkentett információmegosztás és így a predikció minőségének csökkenése árán biztosították a nagy futási sebességet, a [27] cikkben bemutatott módszer alapján lehetséges az eredeti szűrőnél akár jobb pontosságot is elérni, míg a futási idő drasztikusan lecsökken.

A GPU architektúrák nagymértékben adatpárhuzamosak, hatékony számításokat tesznek lehetővé, és széles körben elterjedtek, ellentétben a CNN és az FPGA architektúrákkal. A korábbi GPU implementációknál az újramintavételezés lépésével együttjáró információveszteség limitációt jelentett az elérhető sebességnövekedéssel szemben.

I.1. tézis: Algoritmust terveztem, amely lehetővé teszi a Celluláris Particle Filter (CPF) hatékony adaptációját a GPU architektúrára. A logikai reprezentációt oly módon képeztem le a GPU memória architektúrájára, hogy a sebességet maximalizáljam, de a lokális összeköttetések az eredeti CPF topológiának megfelelő módon megmaradjanak.

Olyan módszert alkottam a celluláris részecskeszűrő struktúra alapján, amely lehetővé teszi az információmegosztást a GPU architektúrán. A megosztott memórián (*shared memory*) belül összefűzött (*coalesced*) szálkezeléssel és a globális memórián keresztül történő szinkronizációval gyorsabb futáshoz jutunk, mint a *surface* vagy *texture* memóriatípusok használatával. Ezért olyan módszert terveztem, amivel a lokális leképezések a kétdimenziós CNN architektúrában hatékonyan megfeleltethetőek az egy-dimenziós, írható/olvasható, gyors elérésű, a chip-en található memóriának a GPU-n (a koncepciót a 3. ábra mutatja be).



3. ábra. Az N blokk egy gyűrű jellegű topológia segítségével történő lineáris reprezentációjának illusztrációját mutatom be ezen az ábrán. B_i jelöli az i . blokkot, és N_{B_i} a hozzátartozó szomszédságot az előző blokkból, $i \in 1, \dots, N$.

Mérésekkel megmutattam, hogy a hiba hasonló vagy jobb, mint a meglévő GPU implementációk hibája, míg a sebesség jobb, habár a közvetlen összehasonlítás lehetőségét korlátozza meglévő módszerek során mérésekhez használt GPU architektúrák különbözősége. Ráadásul a legkorszerűbb munkáknál gyakori, hogy csak a kernelek futási idejét mérik és mutatják be. Azok a megközelítések, amelyek hasonló eszközön mérik az algoritmusokat vagy amelyek nem adják meg a mérésre használt architektúra jellemzőit, ott a futási idő azonos nagyságrendben volt a mi teljes, 77 ms futási időnkkel³, ami nem csupán a kernelek, hanem az I/O műveletek idejét is tartalmazza. A GPU-ra implementált elosztott részecskeszűrőkhöz viszonyítva az

³A kutatás idejében (2013-ban) elérhető NVIDIA GeForce GTX 550 Ti GPU-n mérve, CUDA toolkit 4.1 használatával.

algoritmusunk megőrzi a részecskék helyi kapcsolódását, így eléri az eredeti szűrő pontosságát, azonban a teljes futási idő kevesebb, mint 12 milliszekundum 16 ezer részecskével állapotonként, ami a CPU implementációhoz képest 164-szeres sebességfokozást jelent⁴.

Ezzel egyidőben a processzorok 2D elrendezésének leképezési módszere és a helyi kapcsolatok megőrzése az 1D-s memóriarchitektúrára lehetővé teszi más, kétdimenziós kapcsolatokra épülő algoritmusok hatékony leképezését is a GPU-ra.

A szerző téziscsoporthoz tartozó publikációja: [A1, A3]

II. TÉZISCSOPORT: Éldetekcióra épülő mélyhálós módszer a nagyon hasonló, átfedő pozícióban lévő patkányok felügyeletlen szegmentációjára.

Hatékonyság tekintetében az egyes algoritmusok hatékony implementációja mellett figyelembe kell venni a szükséges emberi figyelem- és időszükségletet. Az identitásmegetartó követés és példányszegmentáció számtalan biológiai kutatásban kritikus jelentőséggel bír. Többek közt mezőgazdasági, gyógyszerészeti, illetve viselkedésökológiai tanulmányok során gyakori, hogy a rendkívül hasonló, egy csoporton belüli állatok egyedi viselkedésének elemzésére van szükség, amelyhez jelentős mennyiségű emberi annotációra van szükség. Az ilyen, sokórányi videó automatikus annotációja olyan biológiai kutatásokat tehet lehetővé, amelyek csupán kézi annotációt alkalmazva nem lennének megvalósíthatóak. A gépi tanulásra épülő algoritmusok általában jól teljesítenek szegmentáció és követés terén, viszont rendkívül hasonló, jelöletlen példányok esetén (pl. fehér egerek vagy patkányok esetén) még a legkorszerűbb megközelítések is hibáznak, ahogy azt a 2. táblázat második oszlopa (TS-k, azaz felcserélések száma) is mutatja. Az egymással határos, rendkívül hasonló területek szegmentációja nagy kihívást jelent és hagyományos módszerekkel [A4, A5] is megközelíthető, de ígéretesebb mélyhálók tanulására épülő módszereket használni.

A kutatásom során olyan videókra fókuszáltam, amelyeken akár különböző kezelést kapott, jelöletlen egerek/patkányok szerepelnek, amelyek egyéni viselkedését kell elemezni. Tipikus esetben a kamera fix és az előtér szegmentációja lehetséges, ami összességében előnyt jelent. Ugyanakkor az állatok formájának változásai és a nagymértékű kitakarások jelentős kihívást jelentenek.

Az volt a célom, hogy a *Composite AI* [33, 34] technikára építve olyan eljárást hozzak

⁴A következő NVCC fordító beállításokat használtuk a GPU bináris kódgeneráláshoz, ahogy azt a CUDA SDK útmutató javasolja: `-arch=sm_20; -use_fast_math`. Néhány mérésünk volt `-arch=sm_13` beállítással is. A CPU-n futó C kódot GCC 4.5-tel fordítottuk; a fordító optimalizálási opciója `-O2` volt. Ezek a kapcsolók hatékonyabb, a számítási kapacitás lehetőségeihez illeszkedő fordítást tesznek lehetővé a GPU architektúrára.

létre, amely megbízható szegmentációt biztosít identitástartó példánykövetéshez. A Composite AI különböző tanulási módszereket épít egybe a magasabb szintű humán tudással ötvözve, hogy a felhasznált módszerek egymás limitációit kipótolják a predikció minőségének érdekében.

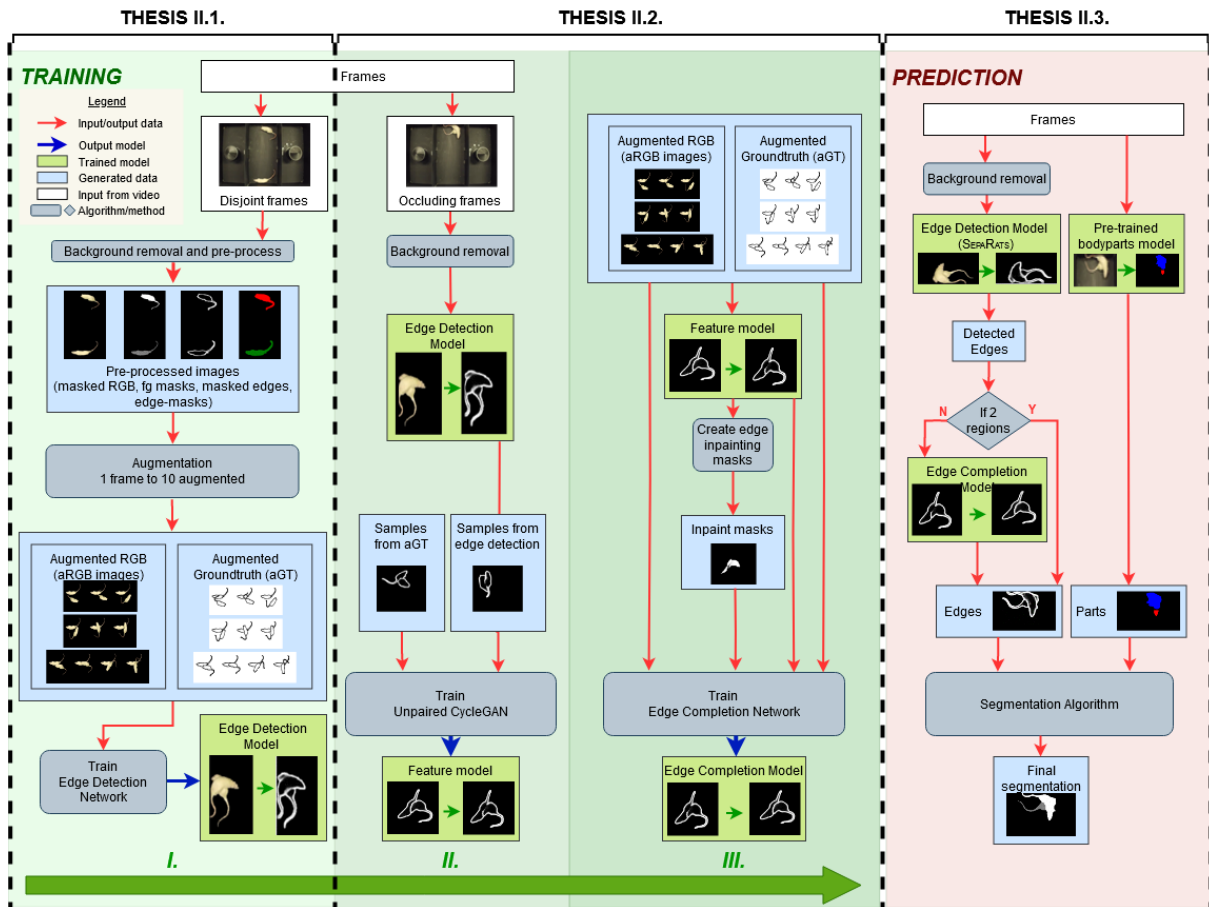
A hasonló példányok képkockáról-képkockára történő követése csupán emberi megfigyelés módszerével rendkívül időigényes, amelyhez képest már egy részlegesen automatizált megoldás is jelentős gyorsulást jelenthet. Ugyanakkor a felügyelt tanításhoz szükséges előzetes annotáció egy-egy kísérlet esetén jelentős időt igényelhet.

Olyan eljárást készítettem, amely automatikus módszerekkel csökkenti a folyamat humán igényét. A módszeremet a 4. ábra illusztrálja.

A mérési eredmények megmutatták, hogy a legkorszerűbb módszereknél jobb eredményt értem el a betanított modellek kimenetéből kapott szegmentációra alkalmazva egy objektumkövető algoritmust.

II.1. tézis: Olyan módszert terveztem, amely teljesen automatikusan, humán annotáció nélkül generál átfedő példányokat tartalmazó képeket kizárólag olyan, automatikusan választott képkockákat használva, amelyeken a példányok nem átfedő helyzetben vannak. A szintetikusan létrehozott képek és élképek tanítóhalmazán tanított éldetekciós modell képes statikus háttér esetén, valós képkockákon, rendkívül hasonló, átfedő patkányok közti szeparáló élt detektálni.

Egy 3 600 képkockából álló videó annotációja amelyen két, rendkívül hasonló patkány látható, nagyjából dupla annyi idő egy humán annotálónak mint a videó hossza, emellett egyhangú és szakértelmet nem igénylő feladat.



4. ábra. A tanító és predikációs lépések illusztrációja. A tanítás során három fő blokkra bontható a folyamat: I. szintetikus adatgenerálás és tanítás az éldetekcióhoz; II. a *Feature Model*, azaz él jelleget biztosító modell tanítása; III. a tanítóadatok kiegészítése és az élkiegészítő háló tanítása. A tanítás kizárólag szintetikusan létrehozott képeken alapul. A nem átfedő példányokat tartalmazó képkockákat használom a szintetikusan létrehozott átfedő példányokat tartalmazó képek és hozzájuk tartozó élképek (GT) generálásához. A betanított éldetekciós hálóval generálom az élképeket azokra a képkockákra, amelyeken az állatok átfednek. Ezeket használom egy *unpaired* (párba állított tanítóképeket nem igénylő) CycleGAN [51] tanítására, hogy generálni tudjak hiányos élképekhez is elvárt kimeneti éleket az élkiegészítő háló tanításához teljesen automatikusan. A predikció során az előtér objektumokra alkalmazom az éldetekciót, majd amennyiben a példányok közti elválasztó kontúr nem folytonos, hanem egyetlen összefüggő régiót alkot a két állat, az élkiegészítő háló „kipótolja” a hiányzó kontúr szakaszt. A szegmentációs algoritmus minden képkockára predikciót ad a példányok maszkjaira. Ehhez a [35] cikkben bemutatott modellel kapott testrész régiókat és az élek által meghatározott területeket használom fel, hogy a létrejött szegmentáció megbízható alapot adjon a hasonló és jelöletlen állatok követésére nagymértékű átfedések esetén is. A módszer teljesen automatikus, nincs szükség egyetlen ponton sem humán annotációra.

Többórányi videó esetében ez olyan mennyiségű adatfeldolgozást jelent, amelyet

egy szakember nem tud megspórolni mielőtt a szaktudást igénylő elemzést és analízist elvégezhetné. Ráadásul, ez az annotációs sebesség akkor érhető el, ha csupán minden hatodik képkockán jelölik a példányokat, ami az időbeli felbontásra kompromisszumot jelent, és összességében rontja az elemzés minőségét a képkockánkénti annotációval szemben.

Egy megbízható szegmentációt előállító mélyháló jelentősen tudja gyorsítani a folyamatot. A tanításához szükséges annotáció azonban szintén hosszú, monoton és általában szakértelmet nem igénylő feladat. Ráadásul mérsékelten rugalmas az annotációs protokoll változásaira, azaz más jellegű feladat vagy a mérési környezet kis változása esetén is újra el kell végezni.

Azoknak a képkockáknak a segítségével generálhatunk szintetikus tanítóadatot, amelyeken, például előtérsegmentáció alapján könnyen szegmentálhatjuk egyenként a példányokat. Ugyanakkor az kritikusan fontos, hogy a szintetikus adat, amit ilyen módon augmentációval kapunk, a mélyháló számára kellően hasonló legyen a valódi képkockákhoz.

Hasonló, átfedő példányok RGB térben történő generálása komoly kihívást jelent, mivel nehéz a nem átfedő példányok maszkjának oly módon történő meghatározása, amely kellően pontos, és épp a megfelelő mennyiségben tartalmazza a részleteket, miközben a háttérből és árnyékokból adódó zaj minimális benne. Mivel a példányok száma ismert és rögzített a teljes videóra nézve, az előtérsegmentáció alapján automatikusan kiválogatjuk azokat a képkockákat, amelyeken nem átfedő helyzetben vannak a példányok. Az élképeket is ez alapján maszkoltam, hogy az éldetekciós hálót [42] szintetikus adatokon lehessen tanítani. Minden képkockához, amelyen nem átfedő példányok vannak, 10 különböző, véletlenszerűen átfedő pozíciót generáltam az RGB színtérben (aRGB) a hozzájuk tartozó élképpel (aGT) együtt. A két példány közti kontúr mentén egy *inpainting* (kitöltés) alapú elmosást alkalmazva biztosítottam az aRGB képeken a realiztikus jelleget. A háló tanulásának javítása érdekében a háttérrel eltávolítottam a generált képkockákról.

A betanított modell segítségével az átfedő pozícióban szereplő példányokat tartalmazó képkockákon belül azoknak a száma, amelyeken a felső példány kontúrja nem zárt 81.53%-ról 17.83%-ra csökkent. Az előtanított, korszerű éldetekciós háló predikciójához képest (ami jobb kimenetet ad, mint a hagyományos Canny [52] vagy Sobel [53] módszerek), ez 4.57-szeres javulásként tekinthető.

II.2. tézis: Olyan mély generatív hálózatokra épített módszert terveztem, amely humán annotáció nélküli tanítóhalmazon tanulva javítja a kontúrok zártságát átfedő, hasonló és jelöletlen példányok élképeire.

A II.1. tézisben bemutatott eljárással kapott élképeken a felső példány kontúrjának zártságát szeretném tovább javítani, mivel a hiányzó szakasz bináris morfológiai operátorokkal nem pótolható. Egy mély generatív hálót választottam el *inpainting*-re. A szükséges tanítóadatot humán annotáció nélkül, automatikusan generáltam. A háromdimenziós térben az egymást átfedő patkányok esetében az egyik a másikon van, ezért a részben sem kitakart példányt, amely pozíciójából adódóan a példányok fölött elhelyezett kamerához közelebb van, felső példánynak fogom nevezni.

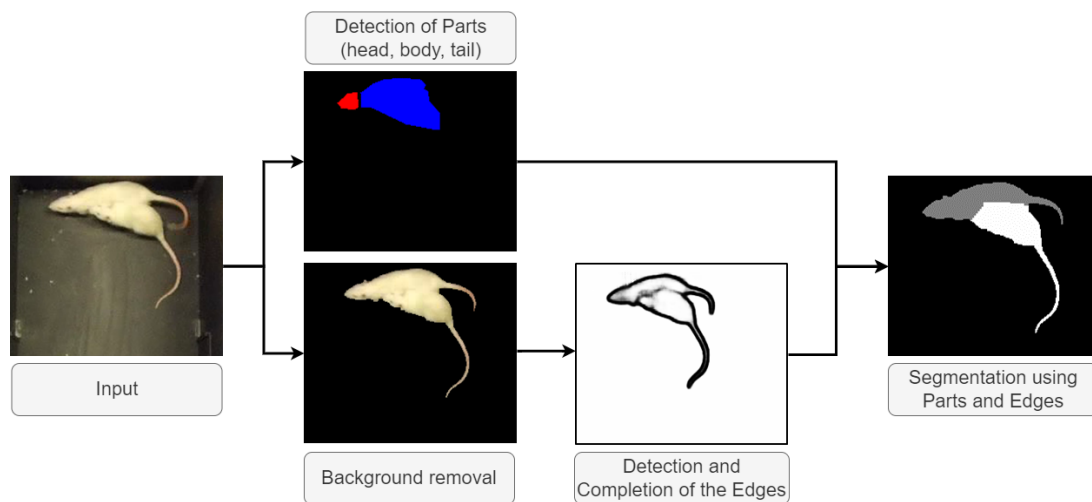
Egy generatív ellenséges élkiegészítő hálóból [44] indultam ki, hogy kiegyesítssem a nem zárt kontúrt, amelyet az éldetekciós háló prediktált a felső állatra. Az élkiegészítő háló tanításához olyan nem zárt kontúrú képekre volt szükség, amelyekhez biztosítva volt a hiányzó élek területét definiáló maszk. A nem zárt kontúrt tartalmazó élképeket egy *unpaired*, azaz nem egymásnak megfelelő tanítóképeken tanuló CycleGAN [51] segítségével generáltam, amely képes arra, hogy képek jellegzetességeit tanulja meg, tipikusan az RGB színtérben. Az ötletem arra épül, hogy ez a háló képes megtanulni adott típusú objektumok élképeinek jellegzetességeit is. Tanítóhalmazként az aGT élképeket használtam és az éldetekciós modell olyan átfedő példányokat tartalmazó valós képkockákra adott élképeit, ahol a felső patkány kontúrja nem zárt. Így a generált képkockák hasonlóak voltak, mint az éldetekció kimenete a felső patkányra, amikor annak a kontúrja nem zárt. Ugyanakkor az ezekhez tartozó elvárt élkép (aGT) a szintetikus generáláson (augmentáció) biztosítja. A hiányzó élek területét definiáló maszkot az előtér maszk és az élkép alapján állítom elő.

A háló eredeti alkalmazásával szemben [44] az RGB inputon nem alkalmazok maszkolást. Mivel a tanítás alatt a hiányzó kontúrszakasz elérhető az aGT szintetikus élképen, ezért módosítottam a diszkriminátor háló loss függvényét, így az előtér és a hiányzó élszakasz területe egyforma arányban befolyásolja diszkriminátor háló loss értékét. Az élkiegészítő háló a leírt tanítást követően tovább tudja csökkenteni azoknak a képkockáknak a számát, amelyeknél a felső példány kontúrja nem zárt.

II.3. tézis: Olyan szegmentáló algoritmust terveztem és implementáltam, amely a detektált élek és testrész régiók kombinálásával megbízható szegmentációt biztosít az egyes példányokra.

Az éldetekáló és élkiegészítő hálókat kombinálva használtam a becsült előtér maszkon. Ha egyetlen régió alakult ki az éldetekció alapján, akkor alkalmaztam

élkiegészítő hálót a kapott élképre. Ha az élek alapján egyetlen régió alakult ki, akkor csak a *testrészt detekció*s módszert [35] alkalmazza az algoritmus a szegmentációhoz. Ha több területre válik szét az előtér maszk az élek alapján, akkor a *testrészt detekció* kimenetével kombinálva hozza létre a szegmentációt. A kezdeti címkézés az élek alapján számítható egy zajcsökkentést követően. Az élek és a testrészt detekció alapján kialakuló területeket az átfedések mértéke alapján rendezem össze. Ez a címkézés a kiindulása egy több címkéjű watershed [54] algoritmusnak, amelyet az előtér maszkon belül futtatok, és amely által létrejön a képkocka szintű szegmentáció. A fő lépéseket az 5. ábra mutatja be.



5. ábra. A predikciós lépéssorozat illusztrációja látható ezen az ábrán. Az a cél, hogy egy képkockán belül maximális legyen a szegmentációs pontosság, amely így megbízható kiindulópontot jelent egy objektumkövető algoritmusnak. Az előtér szegmentáció megadja az objektumok maszkját, aminek a segítségével a háttér nélküli képre meghívom a testrészek és az élek predikcióját, illetve szükség szerint az élkiegészítést. Az élek és régiók információi alapján az egyes példányok szegmentációját egy kombinált algoritmussal számolom ki. Az utolsó ábrarészen megjelenő hiba, ahol az egyik állat feje a másik állat címkéjét kapja, összességében nem befolyásolja a címkék megtartását, a követés hiba nélkül tud emellett futni, mivel a címkék nem cserélődnek fel így sem a két állatra.

A képkockák szegmentációt a [36] cikkben bemutatott objektumkövető eljárással kötöttem össze, hogy mérhető legyen a címketartó követés pontossága. Az eredményeket három, a legkorszerűbb módszerek közé tartozó, a feladat jellegét tekintve hasonló célú objektumkövető eljárással hasonlítottam össze. Míg a módszeremmel nem történt címkecsere a másik három módszer esetében több hiba is előfordult, így nagyon hasonló patkányok valós, laboratóriumi felvételein történő identitáskövetésben és szegmentációban jobbnak bizonyult a módszerem ezeknél. A mérési eredményeket a 2. táblázat foglalja össze.

A szerző téziscsoporthoz tartozó publikációja: [A2]

A tézisekhez tartozó kódok a <https://github.com/g-h-anna/phd-diss-code> címen elérhetőek.

2. táblázat. Az ábra képkockákat mutat szegmentáció alapú, előzetes humán annotációigény nélküli objektumkövető módszerek összehasonlításáról a mérési 3600 képkockából. *TS* a címkecsere-t jelöli, amely metrikát a [36] cikk vezette be. Azt is elvesztett ID-nek jelöljük, ha kevesebb mint kettő előtér-címke-t detektál a módszer. A ToxTrax és idtracker esetében nem elérhetőek a futás során a szegmentált maszkok. Az 1-es felsőindex a hivatkozott módszerek grafikus felhasználó felülete alapján számolt átlagos értékeket jelöli, mert az összehasonlítás csak így volt lehetséges. A BIPED-TL modell az eredeti BIPED előtanított modell [42] súlyaival inicializált, az augmentált adatokkal tovább tanított modellt jelöli. Szürke háttér jelöli az átlagos *Intersection over Union* (IoU) értékeket a képkockákra. A referencia mérésekhez azonos metrikákat használtunk, mint a [36]-es cikkben.

ID Tracking Eredmények							
Módszer	TS-k száma	Képek száma elvesztett ID-vel	IoU átlag	IoU & F átlag	IoU Recall	F átlag	F Recall
ToxTrac [48]	9	267 ¹	N/A	N/A	N/A	N/A	N/A
idtracker [46]	8	1055 ¹	N/A	N/A	N/A	N/A	N/A
idtracker.ai [47]	10	1485	0.5556	0.604	0.59	0.652	0.746
BIPED & Parts	4	2	0.833	0.871	0.978	0.908	0.985
SEPARATS & Parts	0	0	0.846	0.883	0.994	0.921	1.000
BIPED-TL & Parts	0	0	0.845	0.883	0.994	0.921	0.999

4. Az eredmények alkalmazási területei

Az általam javasolt mélyhálós feldolgozási algoritmus, amely képes erősen hasonló példányok megkülönböztetésére akár átfedések esetén is, sokkal hatékonyabb szegmentációs és jobb minőségű egyed követést tesz lehetővé a 3 600 annotált képen, mint a jelenleg ismert megközelítések. Az algoritmus arra is képes volt, hogy egy annotáció nélküli szegmentációt tartalmazó adathalmazon, mely 4 videóból, azaz összesen 224 577 képkockából áll, a példányok mozgását összecserélés nélkül lekövesse, miután a tanító adathalmazt kiegészítettem egy fényesség-alapú szín augmentációval. Jóllehet ez a megközelítés megfeleltethető egy általános éldetekciós modellnek, amely nem igényel emberi annotációt ahhoz, hogy megbízható szegmentációt biztosítson a példányok követéséhez, elképzelhetőek olyan esetek (ahol egy másik típusú háttérrel vagy különböző kinézetű példányokat használunk), amikor egy másik megközelítés jobb megoldáshoz vezethet. A mélyhálók olyan bemenetre adják a legjobb minőségű becslést, amely eloszlása a lehető legközelebb esik ahhoz az eloszláshoz, amely a tanítóadatokon megfigyelhető [55]. Egy olyan videóból kinyert új adathalmaz, amelyben egymást követő jeleneteket alkotnak a képkockák, egyenletes eloszlású adatnak tekinthető, ezért ebben az esetben számíthatunk a legjobb becslésre, mivel a tanítóképek a lehető legjobban hasonlítanak a becslendő képkockákhoz és az eloszlásuk is nagyon hasonló lesz. Ha rendelkezésre állnak képkockák átfedő, valamint nem átfedő pózban is a példányokról, teljesen automatikusan generálhatjuk a tanítóadatokat, ezáltal minimális emberi idő és figyelem szükséges ahhoz, hogy az aktuális bemenethez igazodó tanító adathalmazt biztosítsunk. A szintetikus adatgenerálással és automatikus címkézéssel szerzett tanítóadatok és a várható bementi képek nagy hasonlóak, így ezeken tanítva él-detekciós modellt a példánykövetéshez megbízható szegmentációt érhetünk el. Ezért várhatóan a dolgozatban bemutatott módszerem jól alkalmazható olyan feladatokra, ahol példány szegmentációt kell biztosítani több, jelöletlen, egymást néha eltakaró, rendkívül hasonló példányhoz, állandó háttérű és rögzített távolságú felső kamerából, ha a fényviszonyok lehetővé teszik a vizuálisan megfigyelhető tulajdonságokat (pl. árnyékokat) az állatok közötti átfedések esetén, minimális emberi figyelemmel és erőfeszítéssel.

A részecskeszűrőt alkalmazzák több területen is, ahol a Kálmán-szűrő nem optimális az állapotbecsléshez az állapotdinamika nemlinearitása és nem-Gauss zaj miatt. A GPU-ra adaptált Celluláris Részecskeszűrő megőrzi az eredeti részecskeszűrő azon tulajdonságát, hogy nem korlátozódik egy meghatározott alkalmazási területre. Amely feladatokra a részecskeszűrő megoldást kínál, azokra a GPU CPF az eredeti algoritmushoz hasonló mértékű hiba mellett hatékonyabb, párhuzamos módszert jelent (többek közt képfeldolgozás, önvezető járművek, robotika területén). A javasolt GPU-ra adaptált szűrő könnyen alkalmazható egy- és kétdimenziós bemenetekre (ahogy azt a két benchmark modell kiértékelésében is láthatjuk), és módosítható magasabb dimenziók kezelésére. A bemutatott módszeremben az információmegosztás aránya paraméterezzhető (nem csupán egy, előre meghatározott információmegosztási értéket használhatunk) és a feladatnak megfelelően adaptívan szabályozható, ezáltal lehetőséget ad arra, hogy a részecskék közötti optimális megosztási arányt megtaláljuk, amely a legkisebb hibát és a legnagyobb sebességet biztosítja. Azoknál a feladatoknál, ahol a GPU-n generált mélyhálós detekciót használnak és részecskeszűrésre van szükség a zaj vagy más jellemzők miatt, úgy alkalmazható a GPU-adaptált Celluláris Részecskeszűrő (hasonló becslési minőséget nyújtva, mint a CPU-n egy szálon futó PF), hogy megspórolható az időigényes adatátvitel a GPU és a CPU között. Az 2020-ban bemutatott Ampere GPU architektúra, amelynek számítási képessége 8.x, jelentősen nagyobb hatékonyságot és sebességet kínál, mint a kutatás idején elérhető Fermi architektúra, amelynek számítási képessége 2.x volt. Miközben az algoritmus fő koncepcióját megtartjuk, az architektúra fejlődése miatt a Celluláris Részecskeszűrő teljesítménye a GPU-n jelentősen növekedhet több okból, például a megosztott memóriával kapcsolatos változásoknak köszönhetően. A gyorsabb memóriáhozáférés csökkenti az adatátvitel és a számítások idejét a megosztott memóriában, valamint a megosztott memória mérete háromszor nagyobb az Ampere eszközön [56], mint a Fermi eszközön [57], azaz nagyobb részecskeszám mellett csökkenhet a globális memória-hozzáférések száma a részecskék szinkronizálásához, ami összességében kisebb futási időt eredményez. Ezen felül az Ampere architektúrán a memóriák közti adatátviteli sávszélesség is jelentősen megnövekedett, nemcsak az eszközön belül, hanem a PCIe-v4 buszok alkalmazása a CPU és a GPU között is kétszeres sebességnövekedést hozott. A memóriaelérés sávszélessége 1,6 TB/s (A100 40GB esetén) – 3 TB/s (H100 esetén), szemben a Fermi 192 GB/s sávszélességével. Mivel a módszerem a korszerű GPU-kon történő párhuzamos végrehajtást ugyanúgy lehetővé teszi, mint a 10 évvel ezelőtti architektúrákon, az architektúrák fejlődése miatti számszerű gyorsulás mellett, a párhuzamosításnak köszönhetően várhatóan

megmaradna a szekvenciális CPU algoritmus futási ideje és a párhuzamos GPU adaptáció futási ideje közti nagyságrendbeli különbség.

Jelenlegi és jövőbeli kutatásaim során azt a vizsgálom, hogyan csökkenthető az emberi időráfordítás több állat annotációja és ID-megtartó követése során. A megoldáshoz felhasználok a szintetikus adatokon alapuló mélyhálós tanítást és finomhangolást, valamint a mélyhálós modellek és hagyományos számítógépes látási algoritmusok kombinálását, hogy biztosítsam a Max Planck Biológiai Intelligencia Intézetével közös munkámban több pajzsos cankó követését, amely által lehetséges az egyéni viselkedés hatékony automatikus elemzése oldalnézeti felvételeken, a kitakarások és képelhagyások mellett.

Köszönetnyilvánítás

Először is szeretnék köszönetet mondani témavezetőimnek. Cserey Györgynek szeretném megköszönni, hogy támogatott abban, hogy megtegyem az első lépéseimet a kutatás irányába és mindig számíthattam pozitív és támogató személyiségére, ami segített szembenézni a kihívásokkal.

Rendkívül hálás vagyok Karacs Kristófnak az útmutatásért és támogatásért, amivel kezdettől segített engem. A meglátásai, bátorítása és a heti konzultációnk felbecsülhetetlenek a számomra. Mind a kutatásban, mind az oktatásban az évek során mentorként támogatott. Azzal, ahogy hitt bennem segített, hogy megtaláljam az utam és fejlődni tudjak, míg közvetlen hozzáállása és az, hogy kollégaként kezelt nagyon sokat jelent a számomra, és komoly megtartóerő volt. Nagyon várom, hogy a jövőben is együtt dolgozhassunk, és tovább tanulhassak tőle.

Szeretnék köszönetet mondani Lőrincz András professzor úrnak, aki lehetőséget adott számomra, hogy csatlakozzak a *Neural Information Processing Group* (NIPG) csoporthoz, ahol rendkívül inspiráló munka folyik. A tanácsai, szakmai útmutatása és a rám fordított ideje nélkül a második téziscsoportomban bemutatott eredmények nem jöhettek volna létre. A kutatási kérdésekben megadta nekem a szabadságot, de közben nem hagyta, hogy a minőséget és a célt szem elől tévesszem, és hálás vagyok érte, hogy nem csak a kutatási kérdésekben, de a hosszú távú céljaim tekintetében is segített engem. Bízom benne, hogy továbbra is együtt dolgozhatunk.

Köszönöm Nyékyné Gaizler Juditnak, Szolgay Péternek és Iván Kristófnak, hogy dékánként támogattak és biztosították a lehetőséget, hogy az egyetemen végezhessem a doktori kutatásomat. Rendkívül hálás vagyok Tornai Gábornak a CPF kutatása során nyújtott szakmai támogatásáért és tanácsaiért, és Horváth Andrásnak, akire mindig számíthattam a particle filter részleteinek megvitatásában. Nagyszerű lehetőségnek tartom, hogy az NIPG csoport tagjaival dolgozhattam, különösen: Kopácsi László, Fóthi Áron, Fülöp Zsombor, Varga Viktor, Faragó Kinga Bettina, Ungvári Gergő, Téglás Ervin, Hartyáni Kevin, Szász Milán, Kovács Bálint. Rendkívül sokat adott számomra az a háttér, amit a közös munka, a tapasztalatok megosztása, és a technikai támogatás biztosított.

Szeretném megköszönni az ELTE Informatikai Kar Mesterséges Intelligencia Tanszékének támogatását, és azt is, hogy a kutatásomhoz használhattam az NIPG szervereit. Köszönettel tartozom az ELTE Természettudományi Kar támogatásáért is. Hálás vagyok, hogy lehetőséget kaptam, hogy a kutatásomat a *Max Planck Institute for Biological Intelligence*-szel közösen folytassam a jövőben is.

Külön köszönet illeti Dr. Vida Tivadarnét és Babiczné Rácz Máriát, akikre az ITK-n adminisztratív feladatokban és kérdésekben számíthattam.

Végül, de nem utolsósorban, aligha tudom megköszönni a családom Andris, Julcsi és Marci támogatását, kitartását és türelmét. Szeretném megköszönni a szüleimnek, hogy mindig motiválnak és lelkesítenek, azt, hogy mindig hisznek bennem, és a gyerekeimmel eltöltött sok közös órát, ami segítette, hogy a kutatásra fordíthassam az időmet és a figyelmemet.

Független hivatkozások listája

A szerző [A1] publikációjának független hivatkozásai: *

1. Cole, C. B., Machine Learning Methods for next Generation Sequencing Data: Applications to MLL-AF4 Leukemia and Demographic Inference, *University of Oxford, PhD thesis*, 2021.
2. Spyridon Patmanidis, Alexandros Charalampidis, George P. Papavassilopoulos, Tumor Growth Modeling: State Estimation with Maximum Likelihood and Particle Filtering, *28th Mediterranean Conference on Control and Automation (MED), IEEE*, 2020, 144-149. 10.1109/MED48518.2020.9183193.
3. Havard Heitlo Holm, Martin Sætra, Peter Jan Van Leeuwen Massively parallel implicit equal-weights particle filter for ocean drift trajectory forecasting, *Journal of Computational Physics: X*. 6. 100053., 2020, 10.1016/j.jcpx.2020.100053.
4. Dan Crisan, Joaquin Miguez, Gonzalo Ricardo Ríos Muñoz, On the performance of parallelisation schemes for particle filtering, *EURASIP Journal on Advances in Signal Processing*, 2018.05, doi:10.1186/s13634-018-0552-x
5. David Jáuregui, Patrick Horain, Real-time 3D motion capture by monocular vision and virtual rendering., *Machine Vision and Applications*, 28., 2017., 10.1007/s00138-017-0861-3.
6. Sile Hu, Qiaosheng Zhang, Jing Wang, Zhe Chen, Real-time particle filtering and smoothing algorithms for detecting abrupt changes in neural ensemble spike activity, *Journal of Neurophysiology*, 119(4), 2017, 10.1152/jn.00684.2017.
7. Grigorios Mingas, Algorithms and architectures for MCMC acceleration in FPGAs, *Electrical and Electronic Engineering PhD theses, Imperial College London*, 2015, DOI:10.25560/31572
8. Achutegui, Katrin et al., *A simple scheme for the parallelization of particle filters and its application to the tracking of complex stochastic systems*, 2014, DOI:10.48550/arXiv.1407.8071

*Listakészítés időpontja: 2023. április 3.

Publikációs lista

Folyóirat publikációk

- [A1] **Anna Gelencsér-Horváth** és tsai. “Fast, parallel implementation of particle filtering on the GPU architecture”. *EURASIP J. Adv. Signal Process.* 2013 (2013), 148. old. DOI: 10 . 1186 / 1687 - 6180 - 2013 - 148. URL: <https://doi.org/10.1186/1687-6180-2013-148>.
- [A2] **Anna Gelencsér-Horváth** és tsai. “Tracking Highly Similar Rat Instances under Heavy Occlusions: An Unsupervised Deep Generative Pipeline”. *Journal of Imaging* 8.4 (2022). ISSN: 2313-433X. DOI: 10 . 3390 / jimaging8040109. URL: <https://www.mdpi.com/2313-433X/8/4/109>.

Konferencia közlemények

- [A3] **Anna Horváth**. “Cellular Particle Filter on GPU”. 2011-2012 Academic year köt. 2012, 145–148. old. URL: https://itk.ppke.hu/uploads/articles/159/file/phd_proceedings_2012.pdf.
- [A4] **Anna Horváth**. “Region-merging based on contour-structure of clusters in over-segmented image”. 2012-2013 Academic year köt. 2013, 79–81. old. URL: https://itk.ppke.hu/uploads/articles/159/file/Mini_simposium_2013.pdf.
- [A5] **Anna Gelencsér-Horváth**. “Using contour geometry as a merging cue in oversegmented images”. 2014, 91–94. old. URL: https://itk.ppke.hu/uploads/articles/159/file/Mini_simposium_2014.pdf.

Irodalomjegyzék

- [1] P. del Moral. “Nonlinear Filtering Using Random Particles”. *Theory of Probability & Its Applications* 40.4 (1996), 690–701. old. DOI: 10.1137/1140078. eprint: <https://doi.org/10.1137/1140078>. URL: <https://doi.org/10.1137/1140078>.
- [2] M.S. Arulampalam és tsai. “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking”. *Signal Processing, IEEE Transactions on* 50.2 (2002), 174–188. old. ISSN: 1053-587X.
- [3] Rudolph Emil Kalman. “A New Approach to Linear Filtering and Prediction Problems”. *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), 35–45. old.
- [4] H.F. Lopes és R.S. Tsay. “Particle filters and Bayesian inference in financial econometrics”. *Journal of Forecasting* 30.1 (2011), 168–209. old. ISSN: 1099-131X.
- [5] S. Chib, F. Nardari és N. Shephard. “Markov chain Monte Carlo methods for stochastic volatility models”. *Journal of Econometrics* 108.2 (2002), 281–316. old. ISSN: 0304-4076.

- [6] Petar Djuric és Mónica Bugallo. "Particle filtering for high-dimensional systems". 2013. dec., 352–355. old. ISBN: 978-1-4673-3144-9. DOI: 10.1109/CAMSAP.2013.6714080.
- [7] N. Azzabou, N. Paragios és F. Guichard. "Image reconstruction using particle filters and multiple hypotheses testing". *Image Processing, IEEE Transactions on* 19.5 (2010), 1181–1190. old. ISSN: 1057-7149.
- [8] J. Czyz, B. Ristic és B. Macq. "A particle filter for joint detection and tracking of color objects". *Image and Vision Computing* 25.8 (2007), 1271–1281. old. ISSN: 0262-8856.
- [9] F. Gustafsson és tsai. "Particle filters for positioning, navigation, and tracking". *Signal Processing, IEEE Transactions on* 50.2 (2002), 425–437. old. ISSN: 1053-587X.
- [10] Adrian Dalca. "Segmentation of Nerve Bundles and Ganglia in Spine MRI using Particle Filters". Dissz. 2012. márc.
- [11] M. de Bruijne és M. Nielsen. "Image segmentation by shape particle filtering". *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* 3. köt. 2004, 722–725 Vol.3. DOI: 10.1109/ICPR.2004.1334630.
- [12] Donka Angelova és Lyudmila Mihaylova. "Contour segmentation in 2D ultrasound medical images with particle filtering". *Mach. Vis. Appl.* 22 (2011. máj.), 551–561. old. DOI: 10.1007/s00138-010-0261-4.
- [13] Hamd Abdelali és tsai. "Visual Vehicle Tracking via Deep Learning and Particle Filter". 2020. okt., 517–526. old. ISBN: 978-981-15-6047-7. DOI: 10.1007/978-981-15-6048-4_45.
- [14] A. Doucet és A. M. Johansen. "A tutorial on particle filtering and smoothing: fifteen years later". *Oxford Handbook of Nonlinear Filtering* (2008), 4–6. old.
- [15] Gustaf Hendeby, Rickard Karlsson és Fredrik Gustafsson. "Particle Filtering: The Need for Speed". *EURASIP J. Adv. Signal Process* 2010 (2010. febr.). ISSN: 1110-8657. DOI: 10.1155/2010/181403. URL: <https://doi.org/10.1155/2010/181403>.
- [16] A.S. Bashi és tsai. "Distributed implementations of particle filters". *Sixth International Conference of Information Fusion, 2003. Proceedings of the.* 2. köt. 2003, 1164–1171. old. DOI: 10.1109/ICIF.2003.177369.
- [17] M. Bolic, P.M. Djuric és Sangjin Hong. "Resampling algorithms and architectures for distributed particle filters". *Signal Processing, IEEE Transactions on* 53.7 (2005), 2442–2450. old. ISSN: 1053-587X. DOI: 10.1109/TSP.2005.849185.
- [18] C.Y. Chu és tsai. "Multi-prediction particle filter for efficient parallelized implementation". *EURASIP Journal on Advances in Signal Processing* 2011.1 (2011), 1–13. old.
- [19] Anthony Lee és tsai. "On the Utility of Graphics Cards to Perform Massively Parallel Simulation of Advanced Monte Carlo Methods". *Journal of Computational and Graphical Statistics* 19.4 (2010). PMID: 22003276, 769–789. old. DOI: 10.1198/jcgs.2010.10039. eprint: <https://doi.org/10.1198/jcgs.2010.10039>. URL: <https://doi.org/10.1198/jcgs.2010.10039>.

- [20] Raúl Cabido és tsai. “Multiscale and local search methods for real time region tracking with particle filters: local search driven by adaptive scale estimation on GPUs”. *Machine Vision and Applications* 21 (2008), 43–58. old.
- [21] Kazuhiro Otsuka és Junji Yamato. “Fast and Robust Face Tracking for Analyzing Multiparty Face-to-Face Meetings”. *Machine Learning for Multimodal Interaction*. Szerk. Andrei Popescu-Belis és Rainer Stiefelhagen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, 14–25. old. ISBN: 978-3-540-85853-9.
- [22] Min-An Chao és tsai. “Efficient parallelized particle filter design on CUDA”. *2010 IEEE Workshop On Signal Processing Systems* (2010), 299–304. old.
- [23] Mehdi Chitchian és tsai. “Distributed Computation Particle Filters on GPU Architectures for Real-Time Control Applications”. *IEEE Transactions on Control Systems Technology* 21.6 (2013), 2224–2238. old. DOI: 10.1109/TCST.2012.2234749.
- [24] Olivier Brun, Vincent Teulière és Jean-Marie Garcia. “Parallel Particle Filtering”. *J. Parallel Distributed Comput.* 62 (2002), 1186–1202. old.
- [25] Vasileios Belagiannis és tsai. “Segmentation Based Particle Filtering for Real-Time 2D Object Tracking”. 7575. köt. 2012. okt., 842–855. old. ISBN: 978-3-642-33764-2. DOI: 10.1007/978-3-642-33765-9_60.
- [26] J.V. Candy. “Bootstrap Particle Filtering”. *Signal Processing Magazine, IEEE* 24.4 (2007. júl.), 73–85. old. ISSN: 1053-5888. DOI: 10.1109/MSP.2007.4286566.
- [27] A. Horvath és M. Rasonyi. “Topographic implementation of particle filters on cellular processor arrays”. *Elsevier Signal Processing* (2013). DOI: 10.1016/j.sigpro.2012.11.025.
- [28] L. Chua és L. Yang. “Cellular Neural Networks: Theory”. *IEEE Trans. on Circuits and Systems* 35(10) (1988), 1257–1272. old.
- [29] Dorin Comaniciu és Peter Meer. “Meer, P.: Mean shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603-619”. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24 (2002. jún.), 603–619. old. DOI: 10.1109/34.1000236.
- [30] Li Liu és tsai. “Deep Learning for Generic Object Detection: A Survey”. *International Journal of Computer Vision* 128.2 (2020. febr.), 261–318. old. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01247-4. URL: <https://doi.org/10.1007/s11263-019-01247-4>.
- [31] Tobias Glasmachers. “Limits of End-to-End Learning”. *CoRR* abs/1704.08305 (2017). arXiv: 1704.08305. URL: <http://arxiv.org/abs/1704.08305>.
- [32] Aston Zhang és tsai. “Dive into deep learning”. *arXiv preprint arXiv:2106.11342* (2021).
- [33] Gartner. *The 4 Trends That Prevail on the Gartner Hype Cycle for AI*. <https://www.gartner.com/en/articles/the-4-trends-that-prevail-on-the-gartner-hype-cycle-for-ai-2021>. [Online; accessed 7-Dec-2021]. 2021.

- [34] Debmalya Biswas. *Compositional AI: Fusion of AI/ML Services*. 2021. márc. URL: https://www.researchgate.net/publication/351037326_Compositional_AI_Fusion_of_AIML_Services.
- [35] László Kopácsi, Áron Fóthi és András Lőrincz. “A Self-Supervised Method for Body Part Segmentation and Keypoint Detection of Rat Images”. *Annales Univ. Sci. Budapest., Sect. Comp.* 53. köt. 2021.
- [36] László Kopácsi és tsai. “RATS: Robust Automated Tracking and Segmentation of Similar Instances”. *Artificial Neural Networks and Machine Learning – ICANN 2021*. Springer, 2021, 507–518. old. ISBN: 978-3-030-86365-4. DOI: 10.1007/978-3-030-86365-4_41.
- [37] Bradley P. Carlin, Nicholas G. Polson, and David S. Stoffer. “A Monte Carlo Approach to Nonnormal and Nonlinear State-Space Modeling”. *Journal of the American Statistical Association* 87.418 (1992), 493–500. old.
- [38] N.J. Gordon, D.J. Salmond és A.F.M. Smith. “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”. *IEEE Proceedings F, Radar and Signal Processing* 140.2 (1993), 107–113. old. DOI: 10.1049/ip-f-2.1993.0015.
- [39] G. Kitagawa. “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models”. *Journal of computational and graphical statistics* 5.1 (1996), 1–25. old.
- [40] D. Salmond Gordon N. és A.F.M. Smith. “Novel approach to nonlinear/nonGaussian Bayesian state estimation”. *IEE Proceedings F-140* (1993), 107–113. old.
- [41] O. Morris, M. Lee és A. Constantinides. “A unified method for segmentation and edge detection using graph theory”. *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*. 11. köt. 1986, 2051–2054. old. DOI: 10.1109/ICASSP.1986.1168866.
- [42] Xavier Soria Poma, Edgar Riba és Angel Sappa. “Dense Extreme Inception Network: Towards a Robust CNN Model for Edge Detection”. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2020. márc.
- [43] Xavier Soria Poma és tsai. “Dense Extreme Inception Network for Edge Detection”. *CoRR* abs/2112.02250 (2021). arXiv: 2112 . 02250. URL: <https://arxiv.org/abs/2112.02250>.
- [44] Kamyar Nazeri és tsai. “EdgeConnect: Structure Guided Image Inpainting using Edge Prediction”. *The IEEE International Conference on Computer Vision (ICCV) Workshops*. 2019. okt.
- [45] Saining Xie és Zhuowen Tu. “Holistically-Nested Edge Detection”. *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, 1395–1403. old. DOI: 10.1109/ICCV.2015.164.
- [46] Alfonso Pérez-Escudero és tsai. “IdTracker: Tracking individuals in a group by automatic identification of unmarked animals”. *Nature methods* 11 (2014. jún.). DOI: 10.1038/nmeth.2994.

- [47] Francisco Romero-Ferrero és tsai. "Idtracker. ai: tracking all individuals in small or large collectives of unmarked animals". *Nature methods* 16.2 (2019), 179–182. old.
- [48] Alvaro Rodriguez és tsai. "ToxId: an efficient algorithm to solve occlusions when tracking multiple animals". *Scientific reports* 7.1 (2017), 1–8. old.
- [49] Alvaro Rodriguez és tsai. "ToxTrac: a fast and robust software for tracking organisms". *Methods in Ecology and Evolution* 9.3 (2018), 460–464. old.
- [50] *Apptainer: Application containers*. [Last accessed 16-Jan-2023]. URL: <https://apptainer.org/>.
- [51] Jun-Yan Zhu és tsai. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss". *Computer Vision (ICCV), 2017 IEEE International Conference*. 2017.
- [52] J. Canny. "A Computational Approach to Edge Detection". *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Szerk. M. A. Fischler és O. Firschein. Los Altos, CA.: Kaufmann, 1987, 184–203. old.
- [53] Nick Kanopoulos, Nagesh Vasanthavada és Robert L Baker. "Design of an image edge detection filter using the Sobel operator". *IEEE Journal of solid-state circuits* 23.2 (1988), 358–367. old.
- [54] Anton S. Kornilov és Ilia V. Safonov. "An Overview of Watershed Algorithm Implementations in Open Source Libraries". *Journal of Imaging* 4.10 (2018). ISSN: 2313-433X. DOI: 10.3390/jimaging4100123. URL: <https://www.mdpi.com/2313-433X/4/10/123>.
- [55] Ian Goodfellow, Yoshua Bengio és Aaron Courville. *Deep learning*. MIT press, 2016.
- [56] NVIDIA Ampere architecture paper. "Accessed 2023-02-08". URL: <https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf>.
- [57] *NVIDIA Fermi architecture paper*. "Accessed 2023-02-08". URL: https://www.nvidia.com/content/PDF/fermi%5C_white%5C_papers/NVIDIA%5C_Fermi%5C_Compute%5C_Architecture%5C_Whitepaper.pdf.