

RGBD képsorozat relokalizációja inverz neurális radiancia-mezők segítségével

szervezők: Csehi Ágoston
Dr. Józsa Csaba
külsős konzulens: Dr. Józsa Csaba - Nokia Bell Labs
tanszéki konzulens: Dr. Lengyel László - BME AUT

Absztrakt

A számítógépes grafika, a gépi látás, valamint a robotika számos, mostanra már hagyományosnak mondható térreprezentáló módszert alkalmaz. Ilyenek például a háromszöghálók, pontfelhők vagy az előjeles távolság-mezők. A közelmúltban megjelent egy diszruptív irányzat, neurális radiancia-mezők (NeRF, Mildenhall et al. 2020) alkalmazása térrekonstrukcióra, mely különböző problémák megoldását teszi lehetővé. Segítségével rekonstruálhatóak statikus vagy dinamikus terek mindössze néhány kép alapján, szerkeszthető videók tartalma, és javítható rossz fényviszonyok között készült képek minősége. Neurális hálókat tanítanak egy olyan implicit leképezés reprezentálására, ami a tér minden pontjára és nézeti irányára megadja az áthaladó RGB sugársűrűséget, illetve a pont volumetrikus sűrűségét. A tanításhoz mindössze néhány képre van szükség, ismert merev transzformációkkal (továbbiakban transzformáció. lásd Függelék). A számítógépes grafikában közismert volumetrikus képalkotás algoritmusán keresztül egy ilyen leképezéssel tetszőleges pozíciójú és orientációjú új kép szintetizálható. A volumetrikus képalkotás differenciálhatóságának köszönhetően ez a leképezés megfordítható. A betanított modell segítségével egy optimalizálási probléma eredményeképp egy kép transzformációja is becsülhető.

Kutatásunkban bemutatjuk a NeRF-k alkalmazhatóságát a robotikában ismert relokalizáció problémájára, mely során statikus teret reprezentáló NeRF-ek segítségével keressük monokuláris kamerákkal készített képek pozícióját és orientációját. Az ötlet aktív kutatás témája, de ismeretünk szerint mi vagyunk az elsők, akik ismert relatív transzformációkkal ellátott RGB-D kép szekvenciákat használunk NeRF-kel való relokalizációra.

Ennek hátterében az áll, hogy a mélységtérképek eltéréseiből származtatott veszteségfüggvények gradiensei jóval simábbak, mint a csak egyszerű RGB pixeleket használó veszteségfüggvények gradiensei, melyek tipikusan elég zajosak és mintaigényesek. A NeRF-k alapján történő képalkotás során valós mélység-adatokat is tudunk rekonstruálni intenzív többlet számítások nélkül, így RGB-D kamera alkalmazásával sebességcsökkenés- mentesen növelhető a konvergenciatartomány.

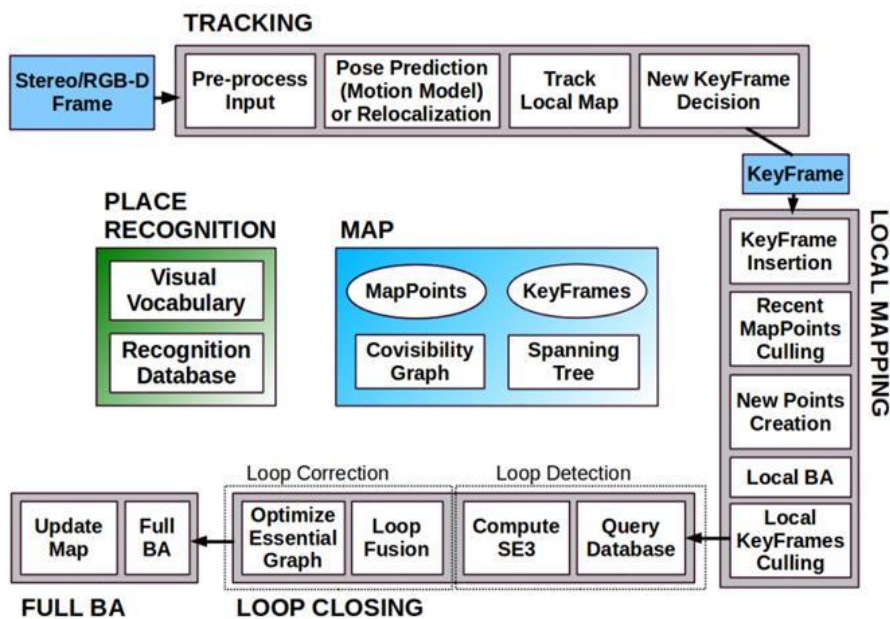
Kitérünk arra is, hogy az konvergenciatartomány jelentősen kiterjeszthető, amennyiben az optimalizációra felhasznált modellt alacsonyabb frekvenciás térreprezentációra tanítjuk be.

Vizsgáljuk továbbá, hogy több kép alapján történő pozíció illesztésével (bundle adjustment) javítható-e a becslés robusztussága, hibatűrése. A megközelítés alkalmazhatósága abból adódik, hogy vizuális odometriában képek sorozata állnak rendelkezésünkre, amelyek közötti relatív elmozdulások becslésére többféle szenzorfüziós megoldást használhatunk. Ezen relatív transzformációk apró hibái idővel felgyülemlenek. Az így keletkező jelentős hibákat relokalizációval kell eliminálni.

Végezetül leírjuk, hogy a javasolt rendszerben hogyan kezelhetőek dinamikus objektumok, amelyek az eredeti NeRF tanítását félrevihetik, illetve rontják a relokalizáció pontosságát, megbízhatóságát.

Bevezető

Napjainkban számos alkalmazásban van szükség centiméteres nagyságrendű pozíciómeghatározásra. Talán a legfontosabb érintett területek, automatizálásukra tett jelentős beruházásoknak köszönhetően, a gyártás-ipar és a közlekedés. Gyártás-iparban például előkerül az önműködő robotok mozgásának koordinálásának feladata, ahol elengedhetetlen az ágensek pozíciójának és sebességének követése. Ágensek alatt nem feltétlenül földön guruló robotokat értünk, drónok (UAV - Unmanned Aerial Vehicle) alkalmazása is kezd elterjedni automatizált gyártósorokon, rakodó üzemekben. Levegőben való útvonaltervezés, útvonalkövetés és ütközés elkerülés során már nem elegendő földre festett jelzéseket követni. A hatékony beltéri légi közlekedés megvalósításához szükséges a tér háromdimenziós feltérképezése és az ágensek állapotainak ebben a térben való rögzítése. Ezt a feladatot oldják meg a SLAM algoritmusok (Simultaneous localization and mapping). A SLAM a robotika egyik fő problémája, melyre bár több, valós ipari környezetben jelenleg is alkalmazott, megoldás is ismert, továbbfejlesztése mégis aktív kutatás témája.



1. ábra ORB-SLAM algoritmus felépítése

SLAM során a tér bejárásával párhuzamosan egy statikus térképet is készítünk annak struktúrájáról. Az 1. ábraán látható az egyik közkedvelt SLAM megoldás, az ORB-SLAM felépítése [1]. Az algoritmus jól demonstrálja a SLAM feladatokban megoldandó problémákat. Az ORB-SLAM első lépése a mélység becslése képek alapján. Erre használhatóak speciális szenzorok, de akár az SGM algoritmus is [11]. SLAM során fontos még az ágens pozíciójának és orientációjának követése. Ezzel a feladattal foglalkozik az odometria. Ismert pozíciók birtokában megfelelően kiválasztott megfigyelésekből már felépíthető egy térreprezentáció a SLAM algoritmusokban. Az ORB-SLAM erre key-frame-eket használ. Key-frame-nek nevezünk azokat a referencia képeket, amik a bejárt tér egy részét jól jellemzik,

valamint ismert a kép transzformációja. Ahhoz, hogy egy látott képhez hozzá tudjuk rendelni a hozzá legközelebb eső key-frame-t, bag-of-words [9] algoritmus alkalmazható.

A mélység-információk becslése és az odometria során minden lépésben új hibát viszünk be a rendszerbe. Ezen hibák idővel összeadódnak melyet az irodalom drift-nek nevez. A drift eliminálására ORB-SLAM-ben bundle-adjustment-et és loop-closure-t is alkalmaznak. Bundle adjustment képek transzformációit próbálja egymáshoz igazítani, hogy a képeken látott közös feature pontok konzisztensek legyenek. A tér azon pontjait hívjuk feature pontoknak, melyek jellegzetesek, eltérő transzformációjú képeken is robusztusan detektálhatóak. SLAM algoritmusokban akkor nyílik lehetőségünk loop-closure-re, mikor egy már korábban feltérképezett térrészbe érünk a bejárás során. Ilyenkor globális konzisztenciára törekedve frissíthetőek a térreprezentáció bizonyos részei. Egy globális nemlineáris optimalizálással csökkenthető a térreprezentációnk által jósolt és az aktuális becsült pozíciónk közötti, a drift hatására kialakult eltérés.

Hasonló algoritmusok szerint működnek például a kereskedelmi forgalomban kapható háztartási robotporszívók navigációja is. Térreprezentációra többnyire kétdimenziós pontfelhőt használnak, melyet egy feltérképezési fázisban állítanak elő LIDAR (Light Detection and Ranging) szenzorokkal. A feltérképezéshez SLAM algoritmust használnak, de egy elkészült statikus térkép birtokában már elegendő odometriát alkalmazni a pozíciómeghatározására.

Odometriában gyakran egy belső kinematikai modellt vagy IMU (Inertial Measurement Unit) szenzort használnak a pozícióból, orientációból és sebességből álló állapotvektor frissítésére. Vizuális odometriának hívjuk az odometria azon típusát, mely az aktuális állapot becslésére képeket használ. Ha egy idő-lépcsőhöz egyetlen kép tartozik, akkor monokuláris, ellenkező esetben stereo képek alapján történő vizuális odometriáról beszélünk. Képsorozatok állnak rendelkezésünkre, amelyeknek egy korlátos idő-ablakba eső részét használhatjuk az állapot becslésre. Vizuális odometriának többféle változata is ismert, de a legtöbb megvalósításban az ORB-SLAM-hez [1] hasonlóan feature pontokon alapul. Ebben az esetben a bemeneti képeken feature pontokat detektálunk [12][13][14]. A kapott feature pontokat összepárosítjuk a képek között, majd minden képtérbeli feature ponthoz egy mélységet rendelünk. Az előálló feature pontok három és kétdimenziós reprezentációi felhasználhatóak egy PnP (Perspective-n-Point) feladat megoldására, mely a bemeneti képek transzformációira ad közelítő algoritmust.

Az ORB-SLAM-ben [1] látott megoldások mellett a drift relokalizációval is eltüntethető. Ilyenkor egy rendelkezésünkre álló statikus térreprezentációhoz illesztjük az aktuális pozíció és orientáció becsléseinket különféle szenzor adatok alapján. Relokalizációra SLAM problémákban is lehetőségünk adódhat, amennyiben már kellően sikerült feltérképezni a teret. Bár SLAM és odometria alkalmazásokban alapvető elvárás a valós-idejűség biztosítása, relokalizációt elég ritkábban, akár offline módon párhuzamosítva futtatni. A relokalizáció eredménye egy transzformáció, amely pontosan megadja az ágens pozícióját és orientációját a referencia

térreprezentációhoz képest. Ilyen transzformáció birtokában a korábbi állapotbecsléseink is pontosíthatóak, a drift hatását visszamenőlegesen is eliminálhatjuk.

Az ORB-SLAM-nél is látott feature pontokon alapuló megközelítést sparse-nak (ritka) hívjuk, mivel a bemeneti kép pixeleinek csak egy részhalmazát veszi figyelembe. Beszélhetünk dense (sűrű) megközelítésről is, mely során a bemeneti kép összes pixele felhasználható az optimalizálás során. Az algoritmusok csoportosításának egy további szempontja, hogy a bemeneti megfigyeléseket közvetlen, vagy közvetett módon használjuk-e fel. Előbbi esetben direkt, utóbbiban indirekt megközelítésről beszélünk. Direkt esetben fotometrikus hibát próbálunk csökkenteni, azaz pixelek között definiáljuk az eltéréseket. Indirekt esetben geometrikus hibát használunk, azaz megfelelően kiválasztott pontok közötti távolságokat minimalizálunk. Ezeknek a kategorizálási szempontoknak tetszőleges kombinációjára ismertek algoritmusok [7].

A legtöbb jelenleg használt feature-based SLAM és vizuális odometria algoritmus sparse-indirekt megközelítést használ [1][2]. Dense-indirekt esetről beszélünk például az optical-flow-t alkalmazó megoldásoknál [3][4]. Direkt megközelítésben a bemeneti szenzor-adatokat előfeldolgozás nélkül, közvetlen módon használjuk az optimalizációhoz. Neurális hálózatok elterjedésével egyre több kutatás foglalkozik ezzel az iránnyal. Itt is beszélhetünk dense [5][6] és sparse [7] módszereket alkalmazó algoritmusokról.

Bár az ismertett megoldások jelentős múltal és számos használható implementációval rendelkeznek, mindegyikük szenved némi hiányosságtól. Minden hasonló megközelítésben kihívást jelent a nézeti iránytól való függés, azaz egy objektum adott pontja más szögből nézve más színű lehet, ami különösen igaz a nem lamberti felületekre. További probléma, hogy statikus térreprezentációban nem szereplő dinamikus objektumok félrevihetik a keresett transzformáció rögzítését. Feature-base megoldásokban jelentős lehet a key-frame-ek tárigénye, továbbá, ha túl nagy a látott kép és a key-frame-ek transzformációinak eltérése, a kevés összepárosítható feature pont miatt pontatlan lesz a becslés.

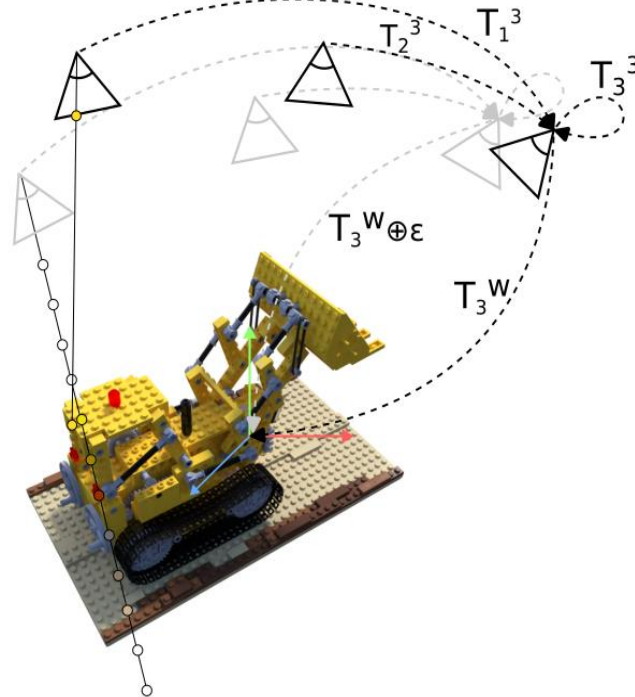
Kutatásunk célja, hogy megmutassuk, neurális hálózatok alkalmazásával a fenti problémák többsége elkerülhető. A neurális hálókkal tipikusan nagyobb hatékonyságú tömörítés érhető el, mint a hagyományos térreprezentációs módszerekkel. NeRF [10] modelleket használva tetszőleges kép szintetizálható, így használhatunk dense megközelítést. A NeRF algoritmus modelljei a nézeti irányányokat is figyelembe veszik, melyek között akár interpolációra is képesek. Ez a tulajdonság növeli a NeRF modellek képrekonstrukciós eljárásának pontosságát. Amennyiben a NeRF modelleket közvetlenül relokalizációra használjuk, mint a későbbiekben részletesen bemutatott INeRF [15] algoritmus esetében is látjuk, nagyobb konvergenciatartományú, robusztusabb optimalizáció valósítható meg a feature pontokon alapuló algoritmusokhoz képest.

Hosszútávú célunk egy neurális radiancia mezőkkel megvalósított térreprezentáción alapuló, teljes SLAM algoritmus felépítése. Ebben a dokumentumban csak relokalizációval foglalkozunk, azaz alkalmazhatóak-e NeRF-k robusztus relokalizációs algoritmus megvalósítására. Bár a feladattal több kutatás is

foglalkozik [16][17][15], tudtunkkal mi vagyunk az elsők, akik mélység-információval is rendelkező képsorozatot használnak NeRF alapú relokalizációra.

Probléma

Az algoritmus bemenete véges hosszúságú RGB-D kép sorozat ismert relatív transzformációkkal és egy kezdeti becslés valamelyik (továbbiakban utolsó) kép abszolút transzformációjára világ koordinátarendszerben. A relatív transzformációk az utolsó kép lokális koordinátarendszerében értelmezettek a 2. ábraának megfelelően.



2. ábra Több kép alapú relokalizáció egy virtuális objektum koordinátarendszerében.
A k -adik kép abszolút transzformációjának hibájának hatását a világosabb színnel jelölt nézeti irányok mutatják.
A sötétebb színnel jelölt nézeti irányok a képek eredeti transzformációikkal egyeznek meg.

Rendelkezésünkre áll továbbá egy pontos neurális statikus térreprezentáció, melyhez képest a becsült transzformáció hibáját kell korrigálni. Az algoritmus kimenete az ettől a hibától mentes transzformáció az utolsó, azaz a legfrissebb képre. Ennek birtokában a többi kép abszolút transzformációja is származtatható.

Tökéletes illesztés esetén az egyes bemeneti képeknek és a végleges abszolút transzformációk alapján rekonstruált képeknek egyezniük kell.

Egy darab RGB képre a fenti probléma felírása a 1. egyenlet szerinti formában írható fel:

$$T^* = \arg \min_{T \in SE3} (L_{rgb}(T|I, \Theta))$$

$$L_{rgb}(T|I_{rgb}, \Theta) = \frac{1}{|R|} \sum_{r \in R \in I_{rgb}} \|\hat{c}_r - c_r\|_2^2$$

1. egyenlet Egy darab RGB képre adott optimalizálási feladat. \hat{c}_r az r sugárhoz szintetizált rgb szín. θ a modell paraméterei. R az I kép pikelehez rendelhető sugarak halmaza.

Kapcsolódó irodalom

Neurális radiancia mezők

A neurális radiancia mezők [10] egyszerű feed-forward neurális hálókat alkalmaznak térreprezentációra. A hálók bemenete egy pozíció és orientáció páros a betanított térrész egy adott pontjában, magasabb dimenzióba kódolva. Az enkódotást különböző frekvenciájú trigonometrikus függvényekkel valósítják meg [19], így a modell finomabb részleteket is meg tud tanulni. Erre azért van szükség, mert a mély neurális hálózatok tipikusan sima, egyszerű felületek illesztésére törekszenek, melyet az irodalom implicit smoothness (simaság) regularizációként ismer [18].

A betanított hálók kimenete a nézeti iránytól függő RGB sugársűrűség adott pontban, illetve a ponthoz tartozó volumetrikus sűrűség érték. A két kimenet akár felírható két különálló neurális háló segítségével, melyeket egy szín és egy volumetrikus sűrűségmező implicit leképezésére tanítunk be. Implementáció során hatékonyabb a két modell paramétereinek egy részét megosztani, de a képletek felírása egyszerűbb, ha két külön modellként gondolunk rájuk. Ismert volumetrikus sűrűségmező birtokában az RGB színek modellezésére egy sekély, egy-két rétegű neurális háló is elég, ezért volumetrikus sűrűség-érték meghatározásához felhasznált súlyok foglalják el a NeRF-k struktúrájának nagyrészét.

A számítógépes grafikában régóta ismert volumetrikus képalkotás problémája. A témában elért eredmények segítségével és a fenti hálók felhasználásával a reprezentált térrészen belül tetszőleges transzformációjú kép szintetizálható. Ehhez először sugarak definiálására van szükség. Adott külső és belső paraméterekkel rendelkező virtuális kamera esetében egyszerűen megadhatóak a szintetizálandó kép pixeleihez tartozó sugarak világ-koordinátarendszerben értelmezve. Perspektív pinhole kamera modellt alkalmazva írható fel a 2. egyenlet:

$$\begin{aligned}o_{cam} &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; d_{cam} = \begin{bmatrix} x_{norm}/f_x \\ y_{norm}/f_y \\ -1 \end{bmatrix} \\ T_{cam}^{world} &= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}; R \in SE(3), t \in \mathbb{R}^3 \\ o_{world} &= o_{cam} \cdot R^T + t \\ d_{world} &= d_{cam} \cdot R^T \\ r(t) &= o_{world} + t \cdot d_{world}\end{aligned}$$

2. egyenlet o_{cam} a sugár kezdőpontja, illetve d_{cam} az iránya a kamera koordinátarendszerben. f_x és f_y jelli a virtuális kamera fókusztávolságát. T a kamera világkoordinátarendszerbeli transzformációja. $r(t)$ megadja a sugár t távolságára eső pontot világ-koordinátarendszerben.

A 2. egyenletben felírt sugarak a tér egy korlátos részén vannak értelmezve. A sugarakhoz tartozó pixelek színeinek kiszámításához a nézeti ponttól indulva végighaladunk a sugarak mentén és megfelelő súlyozással összeadjuk az egyes pontokban a modell kiértékelésével kapott RGB színértékeket. A súlyok

meghatározásához felhasználható a pontok adott sugárhoz tartozó transzmittanciája. Egy p pont transzmittanciája megadja annak a valószínűségét, hogy a nézeti pontból a p látható, azaz, a köztük haladó sugár nem terminál. Ennek valószínűsége, hogy a sugár éppen p pontban terminál nem más, mint a pont transzmittanciájának és volumetrikus sűrűségének szorzata. Ezekkel a valószínűségekkel felírható egy valószínűségi eloszlás a sugár pontjaira a virtuális kamera közeli és távoli vágósíkja között. Folytonos esetben az 3. egyenletben látható módon írható fel az eljárás.

$$T(t) = e^{-\int_{near}^t \sigma(r(s)) ds}$$

$$w(r, t) = T(t)\sigma(r(t))$$

$$C(r) = \int_{near}^{far} w(r, t) \cdot c(r(t), d_{world}) dt$$

3. egyenlet Volumetrikus képalkotás. T jelöli a t távolságra eső pont transzmittanciáját. $\sigma()$ adott ponthoz a volumetrikus sűrűségét, $c()$ pedig rgb sugársűrűségét rendeli. $near$ és far a virtuális kamera két vágósíkját jelöli

A fenti képlet megoldása gyakorlati alkalmazásban nem írható fel zárt alakban. NeRF-k esetén numerikus közelítést használunk helyette:

$$\delta_i = t_{i+1} - t_i$$

$$\alpha_i = 1 - e^{-\sigma_i \delta_i}$$

$$T_i = e^{-\sum_{j=1}^{i-1} \sigma_j \delta_j}$$

$$w_i = T_i \cdot \alpha_i$$

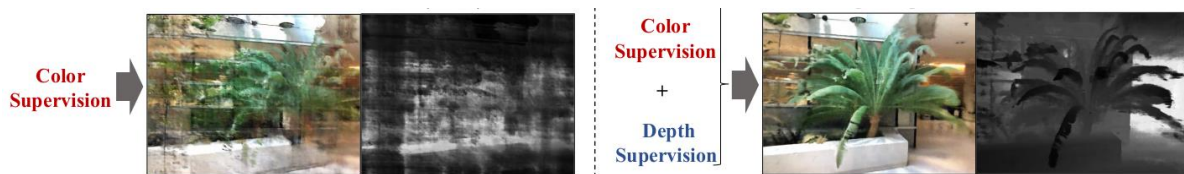
$$C(r) = \sum_{i=1}^N w_i \cdot c_i$$

4. egyenlet Volumetrikus képalkotás diszkrét közelítése. δ két szomszédos t távolság különbségét jelöli.

Fontos kiemelni, hogy az eredeti NeRF implementációban [10] egyszerre két neurális modellt is használnak, egy durva (coarse) és egy finom (fine) felbontásút, az úgynevezett hierarchikus [10] mintavételezés megvalósítására. A megközelítés háttérben az áll, hogy többnyire üres térben szigorú határokkal rendelkező zárt objektumokat modellezünk, így az egyenletes mintavételezés hatására sok felesleges pontban értékelnénk ki a modellünket, hiszen azok csak kis mértékben járulnak hozzá a kimenethez (T_i vagy α_i közel 0). Hierarchikus mintavételezés során először a durva modellt értékeljük ki egyenletes mintavételezéssel a közeli és távoli vágósík között. Ezután a kapott w_i súlyok felhasználásával egy pontosabb mintavételezési eljárás írható fel normális eloszlással, mely előnyben részesíti az eltalált objektum felületéhez közel eső pontokat. A finom felbontású modellt már eszerint a módosított mintavételezési eljárásnak megfelelően értékeljük ki.

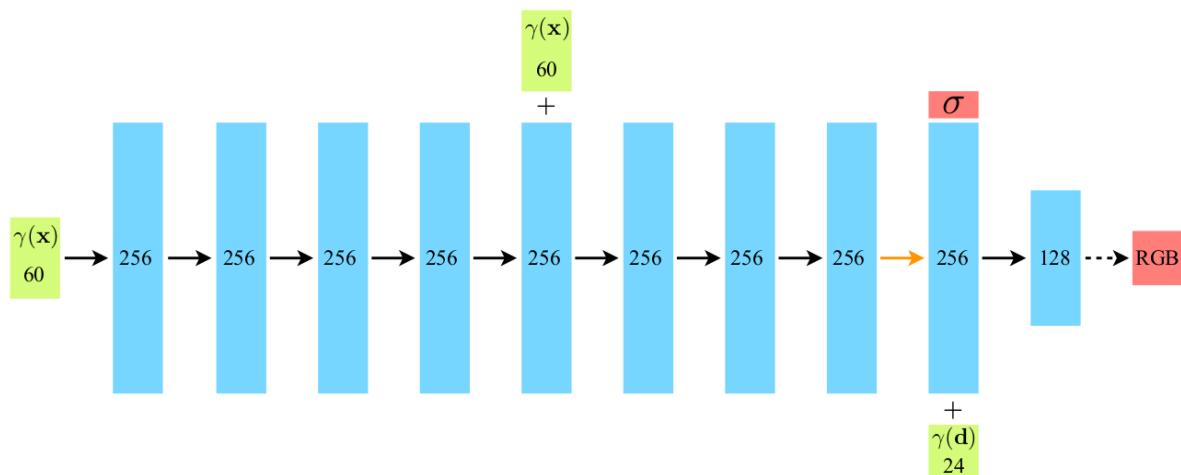
A numerikus közelítéssel felírt (4. egyenlet) formalizmus triviálisan deriválható, így könnyen alkalmazható elsőrendű optimalizálásra. Tanítás során ismert transzformációjú képeket próbálunk újra szintetizálni és a kapott eltérésekből adódó hibát vissza-terjesztjük a modell súlyaira automatikus gradiens számítás és Adam optimalizáló [21] eljárás segítségével. A kép-szintézis hibáját a látott és szintetizált

pixelek közötti átlagos négyzetes hibával írhatjuk fel, az 1. egyenlethez hasonló módon.



3. ábra Mélységinformáció hatása a betanult volumetrikus sűrűségmezőre [20].

Depth supervised NeRF [20] megmutatta, hogy NeRF modellek tanítása nagymértékben leegyszerűsíthető, amennyiben mélység-információk is a rendelkezésünkre állnak. NeRF-k tanítására felírt veszteségfüggvény egy alulhatározott problémát fogalmaz meg. A tanító adathalmazban szereplő képek rekonstrukciójára többféle geometriai struktúrájú volumetrikus sűrűségmező is illeszkedik. Következésképpen a neurális hálók könnyen túlilleszkedhetnek a tanító adathalmazra. NeRF-k esetében ez a hatás úgy nyilvánul meg, hogy csökken a modell extrapolációs képessége, azaz tetszőleges transzformációjú kép szintetizálása során hibás eredményeket kapunk, lásd 3. ábra. Egy mélység-információt is figyelembe vevő tanító eljárással regularizációt vihetünk be a volumetrikus sűrűségmezőre vonatkozóan. Ennek eredményeképp a betanított modell extrapolációs képessége megnövelhető, mint a hagyományos NeRF algoritmushoz képest. Megfelelő mélység-információ birtokában akár a modell kiértékelésének sebessége is növelhető, mert a kép szintetizálására felhasznált sugarak hatékonyabban mintavételezhetőek az objektum felülete közelében. Ezt a gondolatot követi a DOnERF [26], mely egy mélység orákulum hálót használ a sugármenti mélységek becslésére.



4. ábra Eredeti NeRF modellek felépítése [10].

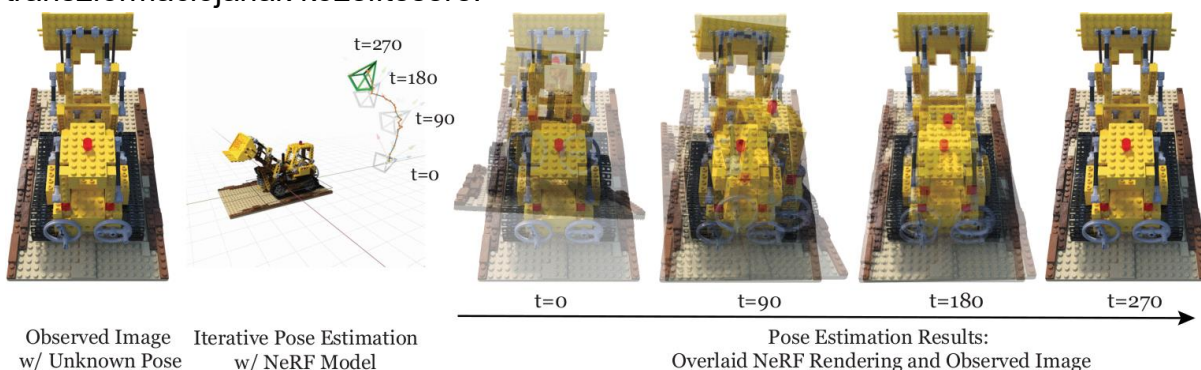
Az eredeti NeRF [10] implementációban használt neurális hálók felépítését a 4. ábra mutatja. 8 réteget használnak egyenként 256 neuronnal a volumetrikus sugársűrűség becslésére, melyhez még egy további 128 neuronból álló réteget alkalmaznak az RGB spekuláris sugársűrűséghez. A hálón belül minden rétegben ReLU aktivációs függvényt használnak. A magasabb frekvenciás térbe való kódolás kimenetét, mely a háromdimenziós pozíció vektort 60 dimenziós térbe képi, a háló

negyedik rétegnek is továbbadják bemenetként. A hierarchikus mintavételezés megvalósításához használt durva és finom felbontású hálók felépítésükben nem térnek el egymástól, viszont a durva modell kiértékeléséhez 64, míg a finom modell esetében 128 minta pontot használnak sugaranként.

Az eredeti NeRF implementáció [10] elég lassúnak mondható. Egy 800x800 pixelből álló kép szintetizálásához közel 123 millió alkalommal kell predikciót futtatni a neurális hálókra, melynek kiértékelése két Nvidia Titan V 12GB-os videokártyával akár 20-200 másodpercet is igénybe vehet. A tanítás akár napokat is igénybe vehet. Az utóbbi években számos cikk megjelent, ami az algoritmus implementációjának gyorsításával foglalkozik [22][23][24][32]. Fast-NeRF-ben [22] a publikált eredmények alapján csaknem 3000-szeres gyorsítást értek el az eredeti implementációhoz képest. 2022 nyarán megjelent az Instant NeRF [23], ami közel valós idejű képkalkotásra is képes, továbbá megfelelő inicializációval akár 5-6 másodperces tanítás után is értelmes rekonstrukció érhető el kisebb terek esetén. Ezt az eredményt kisebb neurális hálók alkalmazásával érték el, melyek bemeneteit egy tanítható paraméterekkel rendelkező magas felbontású hash táblával kódolják feature vektorokba. Számos kutatás foglalkozik a megoldás skálázhatóságával is, amelyek megmutatták, hogy akár városnyi méretű komplex terek is reprezentálhatóak hatékony módon [24][25][31]. Többnyire valamilyen hatékony térfelosztó megoldást alkalmaznak és a tér különböző, akár átlapolódó részeinek reprezentálására külön betanított NeRF modelleket használnak.

Inverz neurális radiancia mezők

A NeRF-vel megvalósítható leképezés tekinthető úgy, hogy merev transzformációkhoz rendel szintetizált képeket. A volumetrikus renderelés differenciálhatóságának köszönhetően a hozzárendelés iránya megfordítható IneRF [15]. A képkalkotásra adott algoritmus nem csak a neurális hálók paraméterei, hanem akár a transzformációt leíró vektor értékei szerint is deriválhatóak. Az így kapott gradienssekkel tetszőleges elsőrendű optimalizáló algoritmus megvalósítható képek transzformációjának közelítésére.



5. ábra Relokalizáció futtatása [15]

Adott egy NeRF modell, mely egy korlátos statikus térrészt reprezentál, egy ismeretlen transzformációjú kép, illetve egy közelítő becslés a kép transzformációjára valamilyen hibával. Amennyiben ez a hiba kismértékű, a becsült transzformációval a

látotthoz hasonló képet szintetizálhatunk. Ha felírjuk a két kép pixeleinek eltérését négyzetes átlagos hibafüggvény szerint, egyfajta least-squared problémát kapunk. A keletkező eltérés gradiense visszavezethető a kezdeti transzformációbecslésünkre. A kapott gradiensekkel elsőrendű optimalizáló algoritmus írható fel. Az ötlet szerzői az Inverz-NeRF [15] nevet adták a megoldásnak, utalva arra, hogy ezúttal inverz módon transzformációt rendelünk képhez. Az algoritmus futtatását az 5. ábra szemlélteti.

A hibafüggvény felírásához teljes képszintézisre nincsen szükség, elegendő kevesebb, megfelelően kiválasztott pixeleken kiértékelni a modellt. INeRF-ben 2048 sugarat használnak [15]. Ekkora mintaszám mellett újabb NeRF implementációk [22][23][26] esetén akár valós idejű algoritmus is elérhető. Az INeRF-el publikált implementációban [15] az eredeti NeRF algoritmust használták, melynek hiperparaméterein nem módosítottak.

Az INeRF kis kezdeti hibájú relokalizációval foglalkozik. SLAM alkalmazásokban viszont nem mindig áll rendelkezésünkre közelítő kezdeti becslés vagy a drift hatására már nagy a pozíció-becslésünk hibája. Ilyenkor nagy konvergenciatartományú relokalizációra van szükség. A későbbiekben bemutatott algoritmusunk az INeRF továbbfejlesztése, melyben kifejezetten a konvergenciatartomány növelésére törekedtünk. Az INeRFben publikált eredmények [15] nem rekonstruálhatóak a közzétett implementációval, így az algoritmushoz saját implementációt készítettünk.

Lie algebra

Az INeRF-ben megoldott probléma a 1. egyenlet leírtak szerint formalizálható [15]. Bár elméletben az euklideszi térben értelmezett merev transzformáció tetszőleges módon parametrizálható (eltolás vektor és kvaternió a forgatáshoz, vagy 4x4-es mátrix...), az optimalizálás ezeken a paramétereken nem triviális. Az optimalizáló algoritmusok euklideszi térben működnek. Kimenetük egy vektor, mely megadja, melyik paramétert milyen irányba és mekkora mértékben kell módosítanunk az optimalizáció egy lépése során. Mivel ez a művelet lineáris, nem lineáris felületekből, mint amilyen az SE(3) illetve az SO(3) is, könnyen kivezethet. Például ha egy 3x3-as mátrixal adjuk meg a transzformáció forgatásáért felelős komponensét, a frissítés során 9 dimenziós térben lépünk, holott az SO(3)-nak csak 3 szabadsági foka van, így jó eséllyel nem SO(3)-beli mátrixot kapunk.

A probléma általánosabban is felírható Lie csoportelmélet segítségével. Lie csoportoknak hívjuk azokat a differenciálható sima felületű vektortereket, melyek eleget tesznek bizonyos axiómáknak [27]. Minden Lie csoportnak létezik egy lineáris megfelelője, melyek között egyértelmű bijektív leképezés lehetséges. Ezt a, gyakran alacsonyabb dimenziós, vektorteret hívják a Lie csoport Lie algebrájának és a leképezés két irányát exponenciális, illetve logaritmusos leképezésnek hívjuk.

$$\begin{aligned}\tau &\rightarrow \mathcal{X} = \text{Exp}(\tau) \\ \mathcal{X} &\rightarrow \tau = \text{Log}(\mathcal{X})\end{aligned}$$

5. egyenlet Leképezések Lie csoport eleme és globális tangenstérbeli megfelelője között. X jelöli a lie csoport egy elemét. míg τ a Lie algebrabeli vektort.

Valójában egy Lie csoporthoz végtelen, egymással izomorf lineáris vektortér rendelhető. Egyik megkülönböztetett ilyen vektortér a csoport nullelemére merőleges vektortér. Valójában ezt hívjuk a csoport Lie algebrájának. A Lie csoport tetszőleges elemére állítható egy a Lie algebrával izomorf merőleges vektortér, továbbiakban tangens tér. A szokásos \exp és \log függvények egy Lie csoport belső elem és a hozzá tartozó lokális merőleges vektortér között valósítják meg a bijektív leképezést. Az 5. egyenletben szereplő nagybetűs függvények tartalmaznak egy további lineáris transzformációt is, ami a lokális tangens teret a csoport Lie algebrájába tolja [27]. Ennek megfelelően a képletben X jelöli a Lie algebra egy elemét és τ a hozzá tartozó Lie algebra-beli vektort.

Levezethető, hogy az $SO(3)$, az $SE(3)$ illetve az \mathbb{R}^n is Lie csoportot alkot [27], tehát létezik egy-egy velük bijektív leképezésben álló lineáris vektortér, mely merőleges a csoport nullelemére. \mathbb{R}^n -re például triviálisan következik, hogy a Lie algebrája önmaga. $SO(3)$ Lie algebrája, $\mathfrak{so}(3)$, háromdimenziós, míg $SE(3)$ -é hatdimenziós.

$SE(3)$ -beli optimalizáció során felírhatóak a transzformációk a csoport Lie algebrájában, $\mathfrak{se}(3)$ -ban parametrizálva. Az Exp illetve Log leképezések az előbbi csoportokban differenciálhatóak. Ennek köszönhetően az optimalizációban használt hiba-függvény gradiense visszavezethető ezekre a paraméterekre.

A megoldás előnye, hogy az optimalizálás során végig $SE(3)$ -ban maradunk. Ha például merev transzformációkat 4×4 -es mátrixokkal reprezentálnánk és az optimalizálási frissítési lépéseket erre a 16 paraméterre számolnánk ki, akkor egyből kilépnénk $SE(3)$ -ból. Ez a hiba kezelhető lineáris projekcióval, de akkor nem közvetlenül a kiszámolt gradiensek szerint lépdelnénk a felületen, így lassítva az optimalizálandó algoritmust.

Ahogy korábban említettük, egy Lie csoportbeli elemhez rendelhető egy globális (Lie algebra) és egy lokális tangens vektortér is. Ennek megfelelően a Lie algebra egy elemének perturbációja tangens térbeli vektorral kétféleképpen is felírható. Az elemet leképezzük a lokális vagy globális tangens-térbe, eltoljuk egy delta vektorral, majd az eredményt visszaképezzük a csoportba. Elméletben lokális tangens tér használatával stabilabb optimalizációs eljárás kapható, így a továbbiakban csak ezzel a tangens térrel foglalkozunk [27]. Egy Lie csoportbeli elem az előbb felírt módon való perturbálására a 6. egyenletben szereplő jelölést használjuk.

$$\mathcal{X} \oplus \delta = \mathcal{X} \circ \text{Exp}(\delta) \in \mathcal{M}$$

6. egyenlet X Lie csoportbeli elem perturbálása egy Lie algebrabeli vektorral.

Optimalizációban való hatékony implementáció érdekében az exponenciális és logaritmusos leképezések Jacobi mátrixai is levezethetőek, így könnyedén integrálhatóak hagyományos JVP (Jacobian Vector Product) vagy VJP (Vector Jacobian Product) megoldásokon alapuló keretrendszerekben is.

Az INeRF-ben megemlítik a Lie algebra használatát, de a publikált implementációjában közvetlenül $\mathbb{R}^{4 \times 4}$ -ben optimalizálnak. Ezzel ellentétben algoritmusunk implementációjában követjük a Lie algebra-ban való optimalizáció megvalósításához szükséges lépéseket.

Az utolsó kép abszolút transzformációjára adott becslés felírható a tökéletes transzformáció egy valamilyen lokális tangenstérbeli hibával perturbált változataként (6. egyenlet).

$$T_{init}^{world} = T_k^{world} \oplus \epsilon$$

$$\delta^* = \arg \min_{\delta \in \mathbb{R}^6} \left(\left\| (T_{init}^{world} \oplus \delta) \ominus T_k^{world} \right\|_2^2 \right)$$

7. egyenlet *Optimalizálás Lie algebrában. T_{init} a kezdeti, hibás transzformációbecslést jelöli.*

Ezáltal átalakítható az 1. egyenletben felírt probléma is.

$$T^* = T_{init}^{world} \oplus \arg \min_{\delta \in \mathbb{R}^6} (L_{rgb}(T_{init}^{world} \oplus \delta | I, \Theta))$$

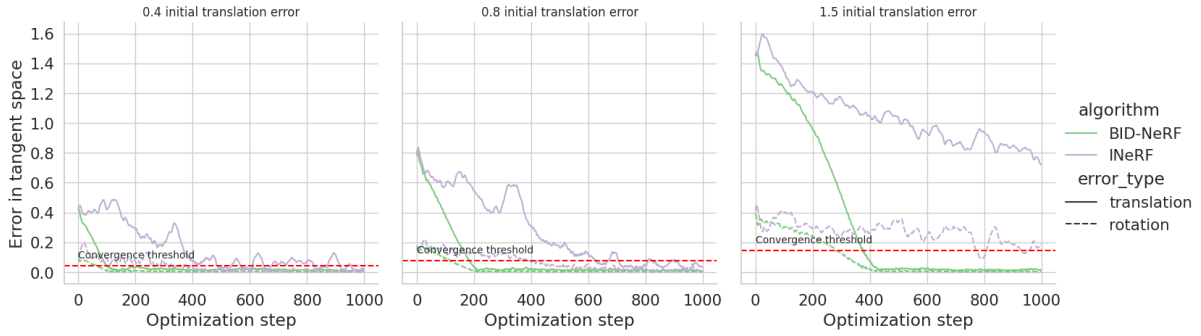
8. egyenlet *Optimalizálás SE(3)-ban.*

Ezt a globális minimumot viszont csak iteratívan tudjuk közelíteni elsőrendű optimalizációs algoritmus segítségével. Az i -edik lépésben kapott transzformációt a 9. egyenlet alapján írhatjuk fel, ahol δ_i az optimalizáló által kiszámolt lépést reprezentáló tangenstérbeli vektor.

$$T_i = T_{i-1} \oplus \delta_i$$

9. egyenlet *Iteratív transzformáció-frissítés Lie algebrebeli vektorral való perturbációval*

Bundle-adjusted inverse depth supervised NeRF



6. ábra Eltolási és forgatási hiba alakulása relokalizáció futtatása során. A három ábra három különböző kezdeti hibájú optimalizációt mutat. Lila szín jelöli az INeRF teljesítményét, míg a mi algoritmusunkhoz zöld szőn tartozik. Vízszintes szaggatott piros vonal jelöli a konvergenciához rendelt küszöbértéket.

Kutatásunk során az INeRF-ben leírt algoritmust fejlesztettük tovább [15]. A referencia megoldáshoz képest gyorsabb, azaz kevesebb lépést igénylő optimalizálást értünk el, továbbá megnöveltük a konvergenciatartományt is (6. ábra). Ennek elérése érdekében a következő módosításokat alkalmaztuk az INeRF-hez képest:

- RGB képek mellett mélységképeket is felhasználunk, így új regularizációs tag vehető fel a veszteségfüggvénybe.
- Az algoritmus bemenete egy helyett k darab képből áll.
- Mellőzzük a NeRF modellek hierarchikus mintavételezését.
- A sugarak generálásához használt pixeleket az iteratív optimalizáció minden lépésében újra mintavételezzük.

Az így kapott algoritmust, a NeRF-el foglalkozó cikkek hagyományát követve, BID-NeRF (Bundle-adjusted Inverse Depth-supervised Neural Radiance Fields) névvel látjuk el, mivel az INeRF-el ellentétben az optimalizáláshoz képsorozatokat és Depth-supervised NeRF-eket [20] alkalmazunk.

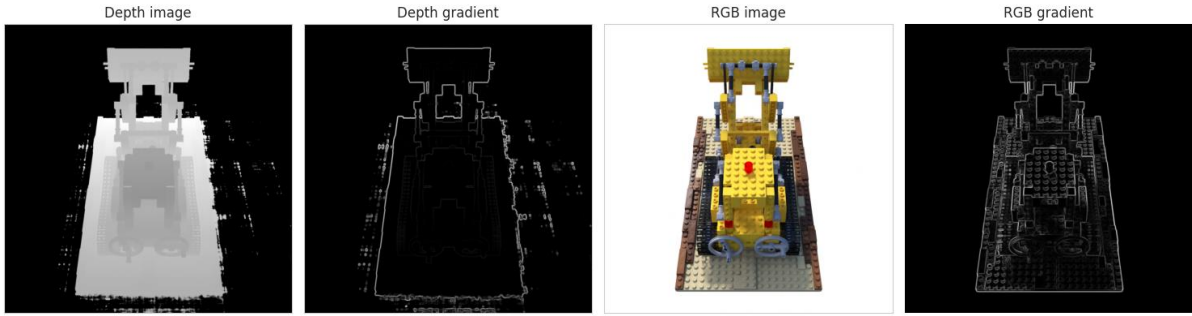
A megoldandó probléma a formálisan az 10. egyenlet szerint írható fel szerepel.

$$T^* = T_{init}^{world} \oplus \arg \min_{\delta \in \mathbb{R}^6} \left(\frac{1}{k} \sum_{i=1}^k L((T_{init}^{world} \oplus \delta) \circ T_i^k | I_i, \Theta) \right)$$

10. egyenlet Optimalizáció több képre.

Mélységtérkép

A modellezendő terekről rendelkezésünkre álló RGB képek képtérbeli gradiense gyakran zajosak. Ez annak köszönhető, hogy az RGB képek nagyfrekvenciásak, mivel az objektumok geometriájától és textúrájától is függenek. Mélységtérképek csak az objektumok geometriájától függenek, így általánosságban kevésbé zajosak, gradienseik kisebbek, simábbak. Ezen tulajdonságukból adódóan kiválóan alkalmazhatóak robusztus optimalizációs eljárásokban.



7. ábra Mélység és RGB képek és képtérbeli gradienseiknek normája.

Napjainkban mélységérzékelésre már számos megoldás elérhető, akár megfizethető áron is. Adódik tehát a kérdés, hogy fejleszthető-e a relokalizációs eljárás mélységtérképek RGB képek mellett való alkalmazásával. A mélységtérkép nem kell, hogy feltétlen kétdimenziós legyen, használhatóak például LiDAR szenzorból származó pontfelhők is. A továbbiakban valamilyen monokuláris time-of-flight kamera meglétét feltételezzük, melynek felbontása megegyezik az RGB kameránk felbontásával.

Korábban láttuk, hogy NeRF-k [10] tanításának gyorsítására, extrapolációs képességük javítására felhasználhatóak mélység-adatok, mint regularizációs tényezők [20]. Hasonló hatás érhető el akkor is, amikor NeRF modelleket nem képszintézisre, hanem relokalizációra használjuk. Jobb rekonstrukciós extrapoláló képességgel pontosabb referencia képek állíthatók elő, illetve a mélységképek képtérbeli gradiensei is simábbak, ami növeli az optimalizáció konvergenciatartományát.

Ahhoz, hogy a veszteségfüggvényünket kiegészíthessük egy mélység adatokat is figyelembe vevő taggal, meg kell adnunk a képszintézis során használt sugarakhoz tartozó mélység kiszámolásának módját. Sugár mélységén azt a világkoordinátarendszerbeli távolságot értjük, amit a sugár megtesz, míg a nézeti pontból az első felületig elér. Ez a definíció csak szigorúan korlátos objektumokra értelmezhető, de relokalizációban ritkán kell másfajta objektumokkal foglalkozni. A korábban felírt képlethez (4. egyenlet) hasonlóan NeRF-kben mélységtérképeket is szintetizálhatunk a 1. egyenlet alapján.

$$Depth(r) = \sum_{i=1}^N w_i t_i$$

11. egyenlet Mélység szintetizálása NeRF-kben

A 11. egyenletben szereplő tagokat az RGB kép szintetizálása során is kiszámítjuk, így a mélység adatok többlet-számítások nélkül előállíthatók. Az eddigiek alapján a következő (12. egyenlet) frissített hibafüggvény írható fel:

$$L(T|I, \Theta) = L_{rgb}(T|I_{rgb}, \Theta) + \lambda_{depth} \cdot L_{depth}(T|I_{depth}, \Theta)$$

$$L_{depth}(T|I_{depth}, \Theta) = \frac{1}{|D|} \sum_{d \in D \subset I_{depth}} \|\hat{d} - d\|_2^2$$

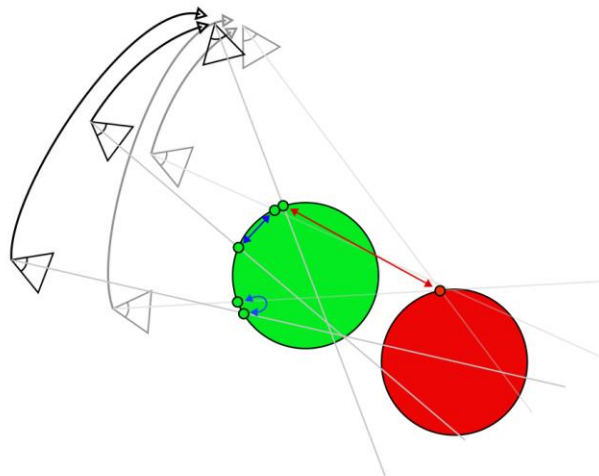
12. egyenlet Mélységinformációt is tartalmazó hibafüggvény. λ_{depth} egy súlyozó hiperparaméter.

A kereskedelemben kapható mélységérzékelő szenzorok ritkán szolgáltatnak minden pixelre megbízható mérési adatokat. Fontos megjegyezni, hogy algoritmusunkban bemeneti képek pixeleinek mintavételezése során figyelembe kell venni a kapott mélységtérképek hiányosságait. A mintavételezésre kijelölt pixelekből kiszűrjük azokat, amelyekre nem áll rendelkezésünkre értelmes mélység adat.

Első hipotézisünk, hogy mélység-adatok használatával megnő a relokalizációs algoritmus konvergencia-tartománya, illetve drasztikusan lecsökken a konvergenciához szükséges optimalizációs lépések száma.

Hipotézisünket arra alapozzuk, hogy a mélységtérképek gradiensei tipikusan simábbak, mint RGB képekéi, továbbá arra, hogy mélységtérképeknek a NeRF-k tanításában is jótékony regularizációs szerepe van [20].

Többképes optimalizáció



8. ábra Sugarak hibája több kép alapján történő relokalizáció közben.

Odometriával megvalósított pozíciókövetés alkalmazásával relokalizációra csak ritkán van szükség. Ha idővel mégis elsodródik az ágens pozíciójára és orientációjára adott aktuális becslés a referencia térreprezentációra illeszkedő tökéletes megoldáshoz képest, a relokalizációhoz felhasználható legfrissebb adatok kiegészíthetők korábbi megfigyelésekkel is.

Odometriában az aktuális állapot becslésének frissítésére felhasznált modellek és szenzorok egy lépés alatt elkövetett hibája elhanyagolható a sok lépés során felgyülemelő hasonló hiba együttes hatásához képest. Ennek tudatában élhetünk azzal a feltételezéssel, hogy a korábbi megfigyelések egy véges halmazának a legfrissebb adatokhoz vett relatív eltérései tekinthetők tökéletesen pontosnak. Több kép alapján történő relokalizáció esetében a képeknek elég a legfrissebb képhez mért relatív transzformációit számításba venni. Nem használható az összes korábbi megfigyelés, hiszen a megfigyelések közötti relatív transzformációk hibája nő a megfigyelések közti lépésszám függvényében, így bizonyos idő után már nem lesz elhanyagolható ez a hiba a drift-hez képest. A felhasználható képek számának optimális értéke nagyban

függ az odometriában használt állapot becslés hibájától és a relokalizáció gyakoriságától.

NeRF alapú relokalizáció során csak akkor számíthatunk konvergenciára, amennyiben a mintavételezett sugarak a megfelelő pixelekhez hasonló színű területet találnak el a térben. Nem kell feltétlen minden sugárnak így viselkednie, csak az a fontos, hogy a kapott gradiensnek átlaga mentén megfelelő irányba módosítsuk a becsléseinket. Mivel egyetlen kép alapján való illesztés folyamán a mintavételezett sugarak közös nézeti pontból indulnak, egészen apró kezdeti hiba is elfajuló helyzetbe viheti az optimalizációt. Például ha az illesztendő képen egy zöld objektum vetületéből mintavételezzük a pixeleket és a hozzájuk tartozó sugarak a becsült transzformáció feltételezett hibája miatt (ez lehet például egy pár fokos eltérés az orientációban) egy piros objektumot találnak el a referencia térben, a kapott gradiensnek nem lesznek hasznosak számunkra (8. ábra). Ha tudnánk több nézeti pontból és irányból is sugarakat indítani egyszerre, akkor lecsökken a hasonló elfajuló esetek statisztikai valószínűsége.

Az előző példát folytatva (8. ábra), ha a zöld objektumot körbejárva több szögből is indítunk sugarakat, akkor számíthatunk rá, hogy legalább pár sugár ugyanazt az objektumot fogja eltalálni a referencia térben, ha nem is feltétlen a képek pixeleinek megfelelő pontokban. Ez az elgondolás vezetett arra, hogy a relokalizációhoz használt sugarak mintavételezése során több, egymáshoz képest rögzített kép pixeleit vehessük figyelembe.

Második hipotézisünk, hogy NeRF-kkel való relokalizációban a sugarak elállítása során felhasználható nézeti irányok és pozíciók számának növelésével javítható az optimalizálás robusztussága, azaz növelhető a nagy kezdeti hibájú illesztések konvergenciájának esélye.

A képek egymáshoz vett helyzetét és a kezdeti transzformáció hibájának hatását a sugarak kiértékelésére, a 8. ábra szemlélteti. Több kép felhasználásával a 10. egyenlet szerint írható fel a sugarak előállítása. Változás az INeRF-ben használt képlethez (8. egyenlet) képest, hogy az összes képre külön kiszámoljuk a hibát és ezeket átlagoljuk. Az egyes képekhez tartozó abszolút transzformáció becsléseket a 13. egyenlet alapján származtathatjuk.

$$T_i^k = (T_k^{world})^{-1} \cdot T_i^{world}; i \in [1..k]$$

$$T_i^{world} = T_k^{world} \cdot T_i^k \approx T_{init}^{world} \cdot T_i^k$$

13. egyenlet i -edik kép relatív transzformációja a k -edik képhez képest

Durva felbontású modell

Az INeRF a NeRF implementációt módosítás nélkül alkalmazza [15]. Korábban láttuk, hogy a konvergencia-tartomány növeléséhez kisebb varianciájú gradiensre van szükségünk a sugarak mentén. Ennek biztosítására használtuk a mélységtérképek simább gradiensait. Az így kapott veszteségfüggvény (12. egyenlet) viszont továbbra is zajos gradiensű hibafelületet eredményez, mivel kiértékeléséhez

RGB- és mélységképek eltéréseit használjuk, melyek képtérbeli gradiensei nagy variációjúak. Ez annak köszönhető, hogy az objektumok geometriái is tartalmazhatnak magasfrekvenciás komponenseket, melyre a NeRF modellek igyekeznek minél jobban illeszkedni a jobb rekonstrukciós képesség érdekében.

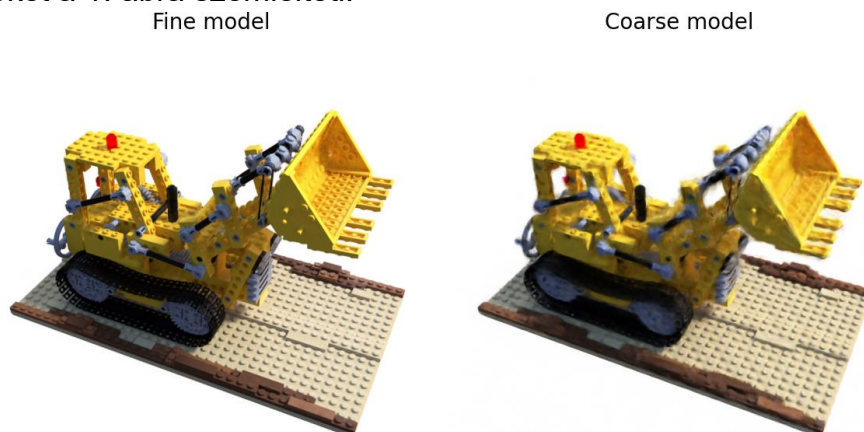
Ezt a gondolatot felhasználva fogalmaztuk meg a harmadik hipotézisünket:

NeRF-ket alkalmazó relokalizációs algoritmusok konvergencia-tartománya növelhető, ha alacsonyabb frekvenciás tér reprezentálására betanított NeRF modelleket használunk.

Az illesztés első fázisában elegendő az eredeti térnek csak egy alacsonyabb frekvenciás reprezentációját használni, majd pontosabb eredmény érdekében a robusztusabb konvergencia után finomíthatóak az eredmények magasabb frekvenciák figyelembevételével. NeRF-k esetén a reprezentált tér spektrális felbontásában szereplő maximális frekvencia korlátozása kétféleképpen valósítható meg.

Triviális megoldás a neurális modellek tanítása során csökkenteni a pozíció és a nézeti irány kódolásában szereplő trigonometrikus függvények frekvenciáit. Igaz, a kapott modellek rekonstrukciós képessége csökken, de relokalizációra vett alkalmazásuk során nő a konvergenciatartomány. Speciális felhasználásra szánt NeRF-ként is gondolhatunk ezekre a modellekre, melyek célja nem a minél pontosabb képszintézis elérése, hanem a kapott képek transzformációinak minél robusztusabb illesztése. Ide tartozik még a NeRF által használt neurális hálók komplexitásának csökkentése is, ami csökkenti a betanítható leképezés komplexitását is.

Másik, egyszerűbben implementálható megközelítés, hogy relokalizáció során a sugarakhoz tartozó RGB és mélység értékeket csak a durvább felbontású neurális háló felhasználásával szintetizáljuk. Bár a durva modell hiperparamétereiben azonos a finom felbontású modellel, tanításakor csak egyenletes mintavételezést alkalmazunk, így, a finom felbontású modellel ellentétben, nem tud ráilleszkedni a tér magasfrekvenciás komponenseire. A két modellel rekonstruálható képek közötti különbségeket a 1. ábra szemlélteti.



9. ábra Finom és durva felbontású modellek által rekonstruált képek.
A durva felbontású modell elmosottabb képet eredményez.

Mindkét megoldás teljesítményére hasonló eredményeket várunk, viszont kutatásunk során csak a második módszert állt módunkban kipróbálni. A megközelítés egyik előnye, hogy a már rendelkezésre álló magasfrekvenciás terekre betanított NeRF modellek esetén triviális az implementáció, csak el kell hagyni a hierarchikus mintavételezést, míg a bemeneti pozíció és orientáció kódolásakor felhasznált függvények maximális frekvenciájának csökkentésével újra kéne futtatni a neurális hálókat tanítását. Szintetikus kicsi terek esetén ez a többlet számítás elhanyagolható, egész városokat reprezentáló modelleknél viszont komoly költségekkel járhat egy hasonló módosítás. Az első módszer választásával, ha képszintézisre is szükségünk van, akkor két NeRF modellt is be kellene tanítani és eltárolni, egy csökkentett bemeneti frekvenciásat relokalizációra és egy hagyományos képszintézisre.

A második módszer további előnye az elsőhöz képest, hogy relokalizáció során csak a durva felbontású modell kiértékelésére van szükség. Mivel az optimalizáció egy lépésében a neurális modellek predikciója a domináló, legszámításigényesebb művelet, így majdnem felére csökken a lépés végrehajtásához szükséges idő. Következésképpen durva felbontású modell alkalmazásával lényegében egyszerre tudjuk az optimalizáció konvergenciájához szükséges lépések számát csökkenteni, a lépések kiértékelésének sebességét csaknem kétszeresére növelni, valamint a konvergencia-tartományt is bővíteni.

Sztochasztikus képtérbeli mintavételezés

Az INeRF számos megoldást bemutat az optimalizáláshoz felhasznált sugarak kiválasztására. Egy ilyen sugár megfeleltethető a bemeneti kép egy pixelével. Legegyszerűbb esetben alkalmazhatunk véletlenszerű mintavételezést [15]. Ennél kifinomultabb eljárásként az illesztendő képen valamilyen feature detektáló algoritmust [12][13][14] futtathatunk és a kiválasztott pixeleket súlyaik alapján mintavételezhetjük [15]. A különböző feature detektáló algoritmusokat a 10. ábra szemlélteti. A véletlenszerű és a feature-pontokon alapuló mintavételezés ötvözésének tekinthető az az eljárás, mely során a detektált feature pontok egy előre meghatározott kis környezetéből választjuk véletlenszerűen a sugarainkat [15].



10. ábra Különböző feature detektáló algoritmusok eredményei.

Az illesztendő kép kezdeti becsült transzformációjának nagymértékű hibája esetén semmi sem garantálja, hogy a becsült nézeti pontból a bemeneti kép egy kiválasztott feature pontján keresztül indított sugárral a referencia térben ugyanazt a feature

pontot találjuk el. Minél nagyobb a transzformáció hibája, annál nagyobb a távolság egy pixelen keresztül indított sugár által eltalált felületi pont és azon pont között, melynek az adott pixel a képtérbeli vetülete. Feature pontokhoz tipikusan nagyobb képtérbeli gradiensek tartoznak, így a pixel lokális környezetében is nagyobb a kép varianciája. Nagy kezdeti hibájú optimalizáció során ezért nem érdemes feature pontokon alapuló mintavételezést alkalmazni.

Amennyiben a kezdeti becsléseink már nagyon megközelítik a tökéletes optimumot, így a szintetizálható kép és a bemeneti kép csak pár pixelben tér el egymástól, pontosabb optimalizálás érhető el feature pontokon alapuló mintavételezés alkalmazásával. Kis kezdeti hibájú finom illesztés az eredeti INeRF algoritmussal is megvalósítható. A mi algoritmusunkban viszont a konvergenciatartomány növelése volt a cél, anélkül, hogy túlságosan megnövekedne a konvergenciához szükséges lépések száma. Emiatt véletlenszerű képtérbeli mintavételezést alkalmazunk.

További kérdés a pixelek mintavételezésének gyakorisága. Elméletben a hibafüggvény kiértékeléséhez elegendő a sugarakat az optimalizáció elején meghatározni egy offline lépésben és az optimalizáció során csak a sugarak transzformációját módosítani. Ahogy a legtöbb optimalizációs probléma esetében, most is úgy tapasztaltuk, hogy a hibafüggvény becsléséhez használt eljárás sztochasztikussága robusztusabb optimalizációhoz vezet, mert könnyebben elkerülhetőek vele a lokális minimumok. Végeredményben arra jutottunk, hogy érdemesebb a bemeneti képet minden optimalizációs lépés elején mintavételezni. Így megnő az optimalizálás egy lépésének számításigénye, de ez a hatás elhanyagolható a lecsökkent konvergenciához szükséges lépések számához képest.

Algoritmus

A bevezetett hipotézisekhez tartozó módosítások összegzésével a következő algoritmus írható fel. Tegyük fel, hogy adott egy vizuális odometria algoritmus, mely diszkrét idő-lépcsőnként RGB és mélység képeket szolgáltat becsült abszolút, világkoordinátarendszerbeli transzformációkkal. Rendelkezésünkre áll továbbá egy NeRF modell, melynek neurális hálói pontosan reprezentálnak egy statikus teret. Ennek a térnek a koordinátarendszere lesz a referencia, melyben a képek transzformációit rögzítenünk kell. Relokalizációra akkor van szükség, ha az odometria során felgyülemelő drift hatására a becsült transzformációk már jelentős hibával rendelkeznek.

Relokalizációhoz vegyük a legfrissebb k darab képek abszolút transzformációit és számoljuk azok relatív transzformációit az utolsó képhez képest az 13. egyenletben is látott módon. Ezt tegyük meg az utolsó, azaz k -edik képre is. Bár a k -edik kép esetében a kapott relatív merev transzformáció identitás lesz, az algoritmus további részeire ennek nincsen hatása, miközben megkönnyíti az implementációt. A relatív transzformációk hibáit elhanyagolhatónak tekintjük. Tetszőleges, hogy melyik kép transzformációját választjuk referenciának. Azért az utolsó használjuk, mert a

relokalizáció így tökéletes eredmény szolgáltat az utolsó képre, míg a korábbi képekre továbbra is hatással lesz az odometria során használt állapot-frissítés apró hibája.

A relokalizációt egy iteratív elsőrendű optimalizáló algoritmussal valósítjuk meg. Az optimalizáló egy lépése a következő allépésekre bomlik:

- A bemeneti képek pixeleit egyenletes eloszlással mintavételezzük.
- Minden pixelhez kiszámolható a hozzájuk tartozó sugarak paramétereit 2. egyenlet lokális koordinátarendszerbe (2. egyenlet).
- A sugarak a képek relatív transzformációk figyelembevételével felírhatóak az utolsó kamera koordinátarendszerében (13. egyenlet 2. egyenlet).
- Az utolsó kép hibás abszolút transzformációjával a kapott sugarak áttranszformálhatóak a NeRF modell koordinátarendszerébe (2. egyenlet).
- A NeRF modell durva felbontású neurális hálójának segítségével előállíthatóak a sugarakhoz tartozó RGB és mélység értékek (4. egyenlet és 11. egyenlet).
- A kapott értékek és a mintavételezett pixelek közötti eltérések felírhatóak átlagos négyzetes hiba segítségével.
- A hiba gradiense láncszabály alkalmazásával visszavezethető az utolsó kép abszolút transzformációjának lokális tangensterében értelmezett δ vektorra.
- Tangensterbeli perturbációval frissíthető az utolsó kép abszolút transzformációjára adott becslés tetszőleges elsőrendű optimalizáló algoritmus szerint.

Implementáció

A hipotézisek teszteléséhez saját implementációt készítettünk el, mind a saját (BiD-NeRF), mind a referenciaként szolgáló INeRF algoritmusokra [15]. A NeRF modellekhez az eredeti NeRF publikációhoz tartozó implementációt használtunk [10] és nem foglalkoztunk új modellek tanításával sem. A sugarak mentén való mintavételezésre a durva felbontású neurális hálóban 64, míg a finom felbontásúban 128 pontot használtunk. Az INeRF-ben publikáltakkal párhuzamosan, egy optimalizálási lépéshez 2048 pixelt használtunk [15], akkor is, ha az optimalizáció során több képből mintavételeztük a sugarakat. Ilyenkor a képek között egyenletesen osztottuk el a sugarak számát. Az optimalizáláshoz Adam [21] algoritmust használtunk 0.02-es learning rate-tel. A gradiensek finomítására gradiens clipping-et alkalmaztunk 0.05-ös paraméterrel. Négyzetes hiba helyett robusztusabb optimalizáció érdekében Huber [28] hibafüggvényt használtunk 0.2-es α paraméterrel.

Eredmények

Ebben a fejezetben bemutatjuk az algoritmusba felvett új megoldások teljesítményét validáló tesztek eredményeit. Implementációknak nem volt célja valós idejű szoftvertermék fejlesztése, így a teszteket a módosítatlan NeRF algoritmus JAX-es [29] változatával futtattuk. Az optimalizálás sebességét a konvergenciához szükséges optimalizáló lépések számában mérjük és a konvergencia-tartomány becslésére 8 különböző szintetikus térben futtatunk teszteket. A kezdeti hibákat az SE(3)-beli merev transzformációk lokális tangensterében definiáljuk adott normájú véletlenszerűen kiválasztott vektorokként. A kiindulási transzformációk permutálását külön végezzük a transzformáció eltolási, illetve forgatási komponenseire az (14) alapján. A permutációs vektorok hosszával változtatható a kezdeti transzformáció becslés hibájának mértéke. A tesztek során előforduló eltérő mértékű kezdeti hibák miatt egy optimalizálás konvergenciáját a kezdeti és végső hibák arányával mérjük. Akkor tekintjük a relokalizációt sikeresnek, amennyiben 1000 lépés alatt az aktuális transzformáció becslés transláció szerinti komponensének hibája eléri a kezdeti hiba 10%-át. Relokalizáció futtatása során az aktuális hiba eltolás és forgatás komponenseit külön szemléltetjük. Az optimalizáció konvergenciájánál azért csak az eltolás hibáját nézzük, mert empirikus tapasztalatok alapján ez a metrika jobban szemlélteti az algoritmus teljesítményét, mint a forgatás hibája.

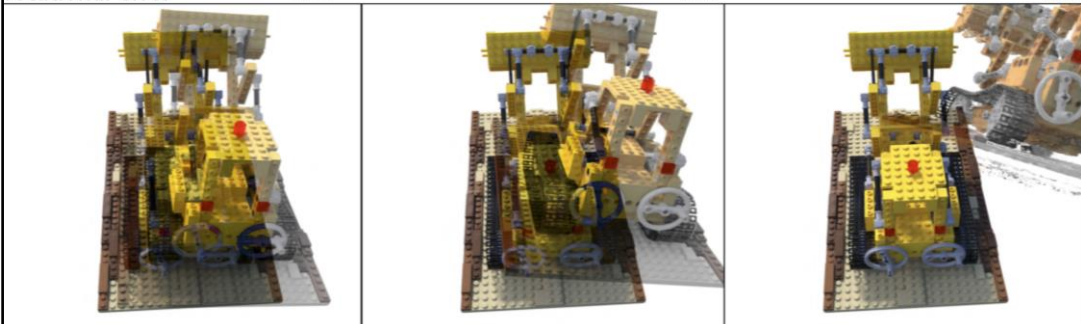
$$R_{twisted} = R_{true} \oplus (\eta \cdot v_R)$$

$$t_{twisted} = t_{true} \oplus (\mu \cdot v_t) = t_{true} + \mu \cdot v_t$$

14. egyenlet Forgatás és eltolás permutálása tangensterben definiált hibavektorokkal. v -k jelölik a tangensterbeli egységnyi hosszúságú hibavektorokat. η és μ pedig a kezdeti hiba mértékét szabályozó hiperparaméterek.

A 11. ábra szemlélteti mekkora is a transzformáció hibáinak hatása.

translational error	0.9	1.8	3.6
rotational error	0.2	0.6	0.8



11. ábra Kezdeti eltolási és forgatási hiba mértékét befolyásoló hiperparaméterek hatása.

Adathalmaz

A tesztek futtatásához a Blender adathalmazt használtuk [10]. A NeRF, illetve az INeRF is ez alapján az adathalmaz alapján publikálta az eredményeit, így értelmes összehasonlítási alapként szolgál. 8 darab virtuális objektumról egyenként 200 darab szintetikusán készült 800x800 pixelből álló képet tartalmaz, lásd 12. ábra. A tesztekhez felhasznált NeRF modellek rekonstrukciós pontosságát az 1. táblázat

mutatja. A modellek rekonstrukciós teljesítményét PSNR metrikában (Peak signal-to-noise ratio) adjuk meg [30].

Scene	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean
PSNR	34.08	25.03	30.43	36.92	33.28	29.91	34.53	29.36	31.69

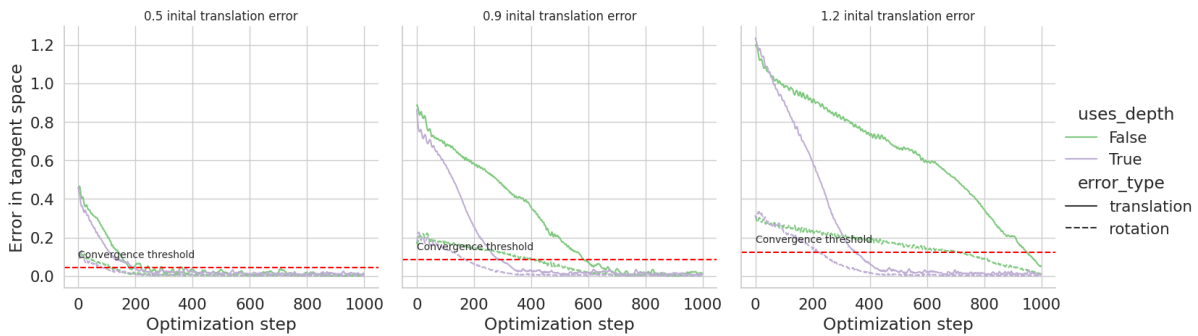
1. táblázat Blender adathalmazra betanított NeRF modellek PSNR értékei.



12. ábra Blender adathalmaz objektumai.

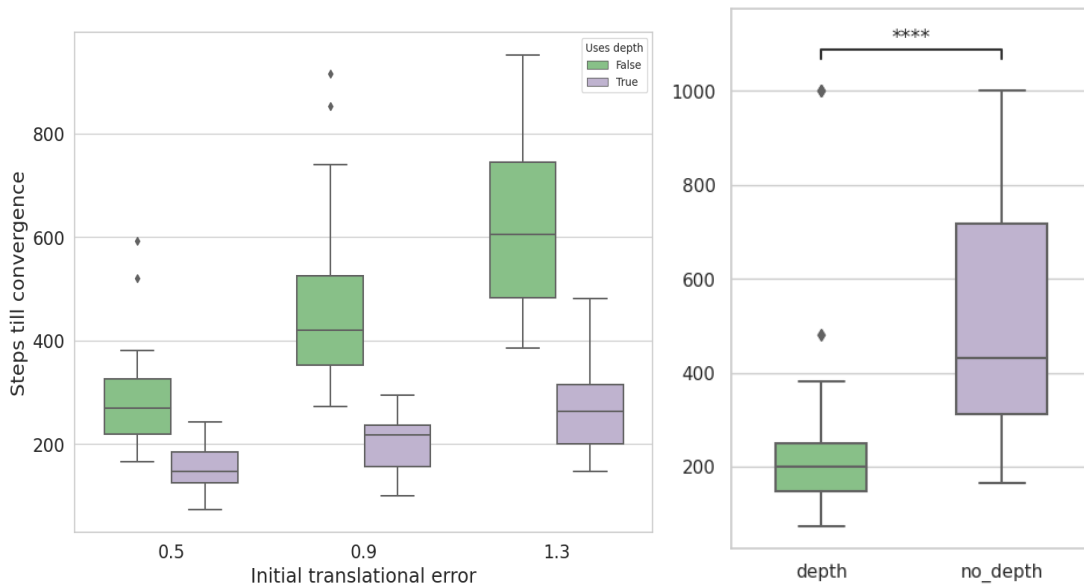
Rendelkezésünkre áll továbbá a képek pontos világ-koordinátarendszerbeli transzformációja mátrixos formában, illetve a képszintézishez használt virtuális kamera-modell fókusztávolsága. Az objektumok egy origó középpontú, kettő oldalhosszúságú kocka alakú térrészbe lettek skálázva, így normalizálva a bemenetet a NeRF modellek tanításához. A normalizálás elengedhetetlen fontosságú, mert nélküle instabilabb lenne a NeRF modellek tanítása, illetve egy új kép előállításakor nem tudnánk mekkora térrészt kéne mintavételeznünk [10]. Előre ismert felépítésű tér normalizációja egy egyszerű skálázással megoldható, dinamikusan feltérképezett tér esetében viszont összetettebb eljárásokat kell alkalmazni [23][32]. A NeRF-ben megemlítik az NDC (Normalised Device Coordinate system) használatának előnyeit [10][15], de ez a megközelítés csak olyan terek esetén alkalmazható, amikor az összes referencia kép közel azonos irányba néz. Dinamikusan bejárt ismeretlen terek normalizálása kívül esik kutatásunk keretein.

Mélység-információ hatása



13. ábra Mélység használatának hatása az optimalizáció során. A három diagram három különböző mértékű kezdeti hibájú relokalizációt ábrázol.

Első hipotézisünk alátámasztására mind a 8 Blender objektumra futtattunk 9 optimalizációt 3 különböző mértékű kezdeti hibával. Az optimalizáció során a durva felbontású neurális hálót használtuk. Minden tesztet kétszer futtattunk le, egyszer L_{depth} (12. egyenlet) tag felhasználásával és egyszer nélküle. A tangenstérbeli hibák alakulását két tesztesetre a 13. ábra szemlélteti. Az összes teszt eredményeit a 14. ábraán látható boxplot diagramon mutatják.



14. ábra Mélységinformáció használatának összesített eredményei. Balra a konvergált tesztek konvergenciához szükséges lépéseinek száma box-plot diagramon kezdeti hiba mértéke szerinti bontásban. Jobbra az összes teszt eredménye.

A 14. ábraáról könnyen leolvasható, hogy nagyobb kezdeti hibájú relokalizáció optimalizációs eljárásának több lépésre van szüksége konvergenciához. Az 14. ábra bal oldalán található boxplot diagram nem tartalmazza az 1000 lépés alatt nem konvergált tesztek eredményeit. Az ábra jobb oldalán az összes érintett teszt eredménye látható, aszerinti felosztásban, hogy használtunk-e a teszt során mélységinformációt vagy sem. Jól látható, hogy mélységinformáció használatával szignifikánsan megnőtt a konvergencia sebessége. A szignifikancia vizsgálatához Mann-Whitney [33] tesztet és t-próbát [34] is alkalmaztunk, mindkét esetben hasonló

eredményt kaptunk. Az ábra jobb oldalán található diagramon szereplő négy darab csillag azt szemlélteti, hogy p értéke 10^{-4} alatt van.

Első hipotézisünket tehát alátámasztják a tesztlejtek, vagyis valóban érdemes mélységinformációt használni NeRF-kkel való relokalizációs algoritmus gyorsítására.

Térreprezentáció frekvenciája

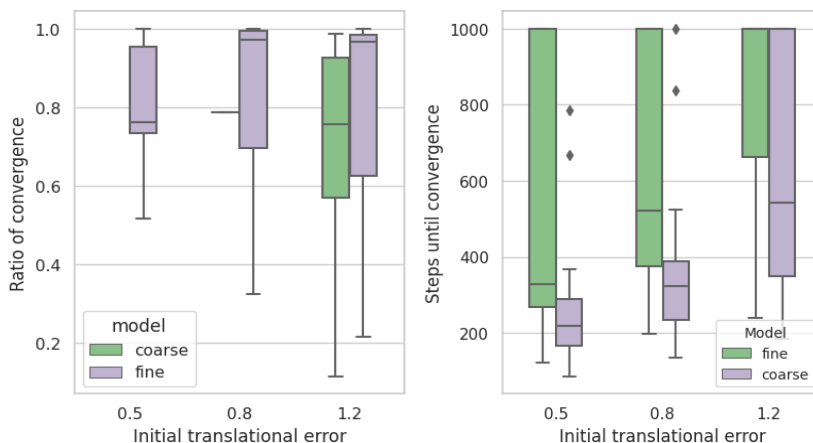
A harmadik hipotézisünk alapján a NeRF modellel reprezentált tér spektrális felbontásában szereplő magasabb frekvenciás komponensek kiszűrésével megnövelhető a modell relokalizációra való alkalmazása során az optimalizáció konvergencia-tartománya.

Korábban láttuk, hogy az ötlet kétféleképpen is megvalósítható. Kutatásunkban nem vizsgáljuk a neurális hálók bemeneteinek kódolásában felhasznált trigonometrikus függvények frekvenciáinak csökkentésének hatását.

A hipotézis teszteléséhez az előző tesztekhez hasonló tesztek futtattunk. Különbség, hogy ezúttal azt is változtattuk, hogy a NeRF modell kiértékeléséhez melyik neurális hálót alkalmazzuk. Pontosabban, hogy felhasználjuk-e a finomabb felbontású neurális hálót és hierarchikus mintavételezést alkalmazunk, vagy maradunk a durvább felbontású neurális hálónál és egyszerű egyenletes mintavételezést használunk. A 2. táblázaton jól látható, hogy finom felbontású modell használatával jelentősen lecsökken a relokalizációs algoritmus robusztussága. Összességében a tesztek 56.3%-ában értünk el konvergenciát a finom modellel, míg hierarchikus mintavételezés esetében ez az arány 85.4% volt. Ezt az eredményt szemlélteti a 15. ábra bal oldala is. Itt csak azokat az eseteket mutatjuk, mikor az optimalizáció nem érte el a korábban definiált 10%-os küszöbértéket. Ebből adódóan csak kevés olyan tesztet szerepeltetünk ezen a diagramon, ami a durva modellt alkalmazta volna. A konvergenciához szükséges lépések száma helyett itt azt ábrázoljuk, hogy mekkora az 1000 lépés alatt elért és a kezdeti becslés aránya. Minél kisebb ez az arány, annál jobb a becslés pontossága. Látható, hogy a durva felbontású modellt alkalmazó tesztek nagy része elérte a konvergenciához szükséges küszöbértéket.

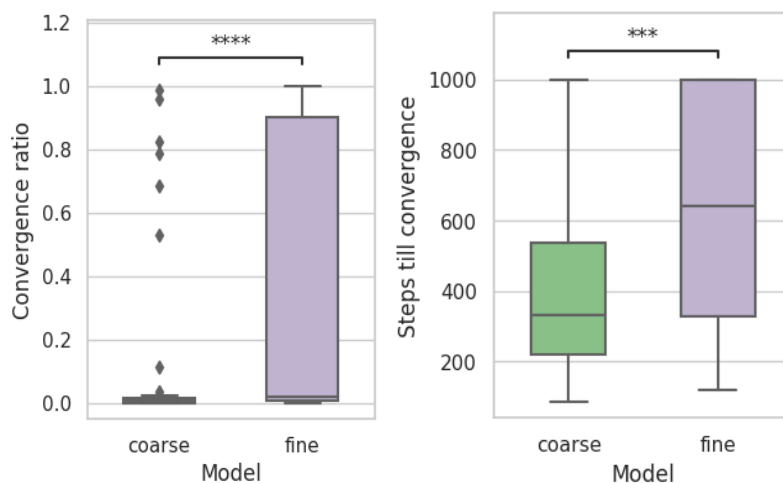
Convergence percentage	Fine	Coarse	All
Depth	87.5%	95.83%	91.7%
No Depth	25.0%	75.0%	50.0%
All	56.3%	85.4%	

2. táblázat Konvergencia valószínűsége a tesztek alapján. Az oszlopokat a felhasznált NeRF modell felbontása alapján bontottuk fel, míg a sorokat aszerint, hogy használtunk-e mélységinformációt az optimalizációs tesztekhez



15. ábra A NeRF modell felbontásának hatása boxplot diagramokon. Balra a nem konvergált tesztek végső konvergencia aránya (minél kisebb, annál jobb), balra az összes test konvergenciájának sebessége kezdeti hiba mértékétől függően.

A 15. ábra jobb oldalán található boxplot diagram azt ábrázolja, hogy a durvább felbontású modell használata a konvergenciához szükséges lépések számát is lecsökkentette, a konvergencia-tartomány növelése mellett.



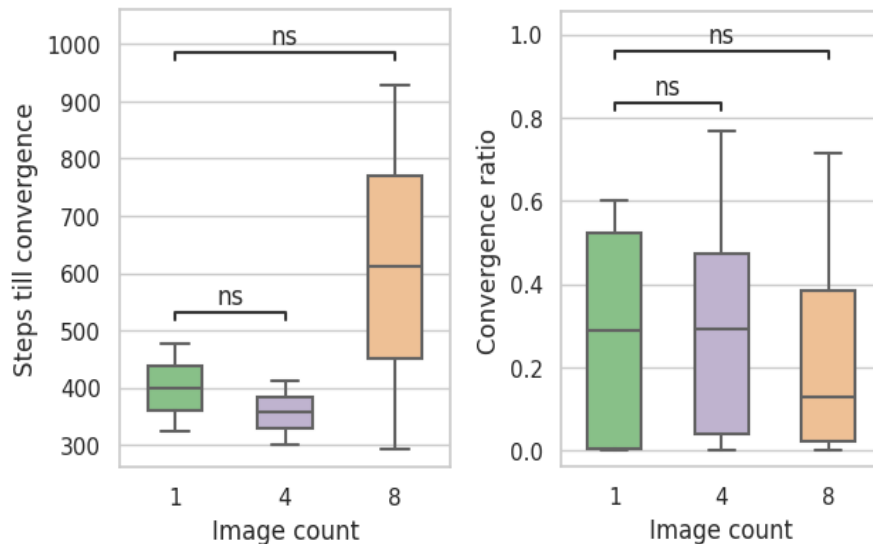
16. ábra A NeRF modell felbontásának hatása a konvergenciatartományra és a konvergencia sebességére nézve. A bal oldali ábrán a relokalizáció előtti és utáni hiba aránya látható, jobbra pedig a konvergencia sebessége boxplot diagramon.

Ezeket az eredményeket támasztja alá az 16. ábra is. Az ábra két oldala ezúttal az összes tesztet tartalmazza, attól függetlenül, hogy konvergálnak tekintjük az optimalizáció eredményét vagy sem. A szignifikancia vizsgálathoz ezúttal csak Mann-Whitney tesztet alkalmaztunk, mivel az adatok nem normális eloszlást követnek.

Végeredményben a tesztjeink a harmadik hipotézisünket is alátámasztották, tehát valóban érdemes nagy kezdeti hibájú, NeRF modelleken alapuló relokalizáció esetében csökkenteni a reprezentált tér frekvenciáját. További eredmény, hogy a csak a durva neurális hálót alkalmazó algoritmusnak a kiértékelése is gyorsabb, mintha a finomabb felbontású modellt is használnánk.

Optimalizálás több kép alapján

Második utolsó hipotézisünk azt állította, hogy több kép alapján robusztusabb relokalizáció érhető el. Ennek vizsgálatára, a korábbiakhoz hasonlóan mind a 8 rendelkezésünkre álló objektumon futtattunk tesztek, három különböző kezdeti hibával.



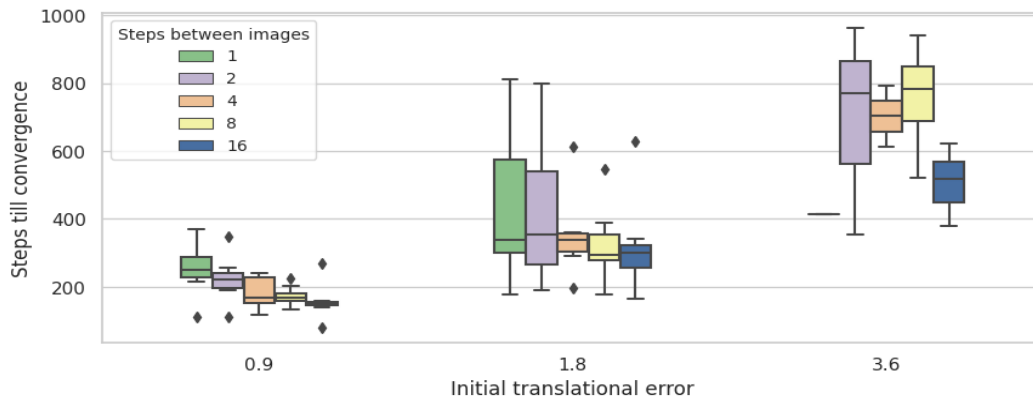
17. ábra Többképes optimalizáció eredményei. Balra a konvergált tesztek, jobbra a nem konvergált tesztek.

Az ábra jobb oldalán látható diagramból kiszűrtük azon tesztek eredményeit melyek sikeresen konvergáltak. Mivel a robusztusság növelése volt a cél, kifejezetten nagy kezdeti hibákat alkalmaztunk. Bár a jobb oldali ábra azt sugallja, hogy sikerült némi javulást elérni az optimalizáció robusztussága szempontjából, az eredményeink statisztikailag nem szignifikánsak, így a harmadik hipotézisünket nem tudjuk teszteredményekkel alátámasztani.

Az ábra bal oldalán szereplő diagramról leolvasható, hogy 4 kép használata javította a konvergencia sebességét egy kép használatához képest. A 8 képhez tartozó kiugró értéket az okozhatta, hogy a Blender adathalmazban a 12. ábraán látható objektumok szerepelnek háttér nélkül. A NeRF modellek kiértékelése során nagy hiba esetén azok sugarak, melyek nem találják el az objektum felületét, fekete pixeleket fognak eredményezni, amik nem használhatóak fel optimalizációra. Minél több képet használunk, annál nagyobb lesz a köztük lévő relatív transzformáció hibája. Minél nagyobb a transzformáció hibája, annál kisebb az esélye, hogy a sugár eltalálja az objektumot. A legfrissebb kép transzformációjának perturbálása annál nagyobb hatással van egy korábbi kép származtatott transzformációjára, minél nagyobb a köztük lévő távolság. 8 kép használata során, ha az utolsó kép transzformációját kis mértékben elforgatjuk, akkor könnyen lehet, hogy a legelső nézeti pontot olyan helyzetbe transzformáljuk, ahonnan már egyik onnan indított sugár sem metszi a virtuális objektumot.

A kapott eredmények kiértékelése során felmerült a kérdés, hogy mi a hatása a felhasznált képek közötti transzformációk nagyságának. Ezt a hatást szemlélteti a 18.

ábra. A teszt során továbbra is 8 képet használtunk, de fokozatosan növeltük a köztük távolságokat.



18. ábra Képek közötti távolság hatása az optimalizáció sebességére 8 kép esetén, aszerinti felbontásban, hogy mekkora volt a kezdeti hiba mértéke.

Láthatóan javul az optimalizáció robusztussága annak függvényében, hogy mennyire változatos nézeti irányokból tudunk sugarakat mintavételezni. Ezeket az eredményeket viszont nem tudtuk statisztikailag alátámasztani szintetikus adathalmazon. A felhasznált transzformációk egy 200 lépésből álló körsétából lettek mintavételezve az objektum körül. Az ugyanolyan színekkel jelzett tesztekhez rendelt számok azt jelzik, hogy az adott tesztekben mekkora lépést tettünk meg a körsétában a képek mintavételezése során. Az 18. ábraán kék színnel jelzett tesztek esetében a képek közötti távolság akkora, hogy csaknem körbejárjuk az egész objektumot. Odometria esetén egy hasonló mértékű elmozdulás után már kevésbé lesznek megbízhatóak a becsléseink a felgyülemelő drift miatt, így megkérdőjelezhető az eljárás alkalmazhatósága.

Az 18. ábraáról leolvasható továbbá, hogy nagyobb kezdeti hiba esetében csökken a képek közötti transzformációk hatása a konvergencia gyorsítására.

Továbbra is fenntartjuk annak a lehetőségét, hogy valós környezetben több kép alkalmazása növelheti a NeRF-kkel való relokalizáció robusztusságát, de a hipotézis alátámasztásához további vizsgálatokra van szükség.

Összesítés

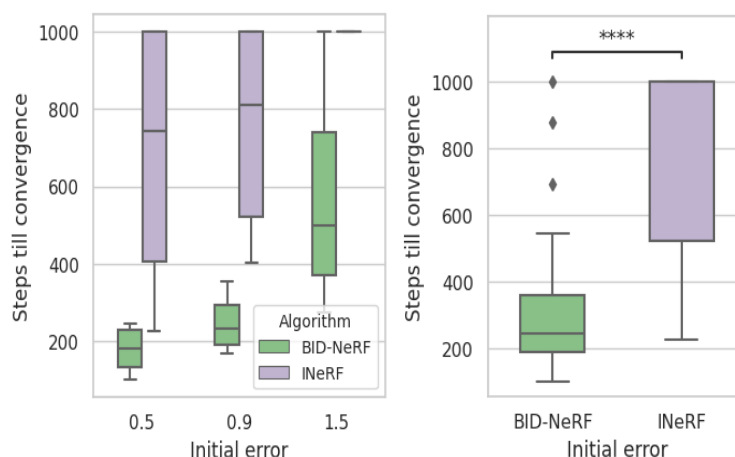
Végezetül összevetjük algoritmusunk (BID-NeRF) teljesítményét a referenciaként szolgáló INeRF-éhez képest. Az optimalizálásokat ugyanazokra a kezdeti transzformáció becslésekre futtatjuk, ugyanazzal a NeRF modellel. Az optimalizáló algoritmus paramétereiben sincsen eltérés. A publikált INeRF implementációval [15] ellentétben a mi implementációnk a transzformáció frissítését valóban lokális tangenstérben végzi, de ez legfeljebb csak javíthatja az algoritmus teljesítményét [27]. A két algoritmus közötti eltérést az adja, hogy BID-NeRF esetében 8 kép alapján generálunk sugarakat, az INeRF-ben használt egy helyett, a veszteség-függvényünkben szerepet játszik a mélység-információ is és a pixelek szintéziséhez csak a durva felbontású neurális hálót használjuk. Mindkét algoritmus összesen 2048 sugarat szintetizál az optimalizáció egy lépésének kiértékeléséhez.

Kezdeti transláció hiba	0.5	0.9	1.5	Összesen
BID-NeRF	100%	100%	87.5%	95.83%
INeRF	50%	50%	0%	33.3%

3. táblázat BID-NeRF és IneRF konvergenciájának valószínűsége a Blender adathalmazon.

Az 12. ábraán látható a materials objektum egyik képe. Az objektum érdekessége, hogy több különálló objektumból áll, melyek színe egymástól eltérő. Ha a relokalizáció kezdeti hibája akkora, hogy adott színű gömbökhöz tartozó pixeleken át indított sugarakat egy másik színű gömbhöz tolja, akkor esélytelen, hogy csak RGB adatok alapján értelmes gradienseket tudnánk előállítani.

Hasonló helyzet fordul elő akkor is, ha a fenti képre egy olyan relokalizációs algoritmust futtatunk, ami csak a mélységinformációkat veszi figyelembe. Ilyenkor az algoritmus nem tudja megkülönböztetni egymástól az objektumokat, hiszen azok geometriája megegyezik. Következésképpen könnyen ragadhatunk lokális minimumban, amikor rosszul párosítjuk össze a gömböket a képen látott projekcióikkal. A Függelék-ben bemutatunk egy éppen ezt a hibát elkövető optimalizációt.



19. ábra BID-NeRF és IneRF teljesítménye a Blender adathalmazon. A bal oldali boxplot diagram csak a konvergált tesztek tartalmazza.

A 19. ábra szemlélteti az INeRF és BID-NeRF összevetésének eredményeit. Korábban láttuk, hogy az INeRF az esetek felében nem tudott konvergálni. A 19. ábra bal oldaláról kiszűrtük ezeket a tesztek. Ezzel szemben a BID-NeRF csak egyetlen tesztben nem érte el a konvergenciához szükséges küszöböt 1000 lépés alatt. Jobb oldalon az összesített eredmény látható. A szignifikancia vizsgálatához megint csak Mann-Whitney tesztet alkalmazhattunk, hiszen a divergálódott tesztek miatt az INeRF eredményei nem normális eloszlásúak. A jobboldali boxplot diagramról leolvasható, hogy az összes bevezetett módosítással szignifikánsan felülmúltuk a kiindulási teljesítményét. Az INeRF-nek Blender adathalmazon konzisztensen legalább kétszer több lépésre van szüksége konvergenciához, ráadásul egy lépés kiértékelésének sebességét is sikerült a felére redukálni. A konvergenciatartományt is jelentősen kibővítettük, így teljesítettük jelen kutatás előre kitűzött céljait.

Összegzés

Bevezettük a relokalizáció problémájához kapcsolatos legfontosabb fogalmakat. Bemutattuk a NeRF algoritmust [10], mellyel pár RGB kép alapján kompakt neurális térreprezentáció rekonstruálható. Az INeRF [15] algoritmusán keresztül formalizáltuk, hogyan lehet NeRF-et relokalizációra használni. Itt külön kitértünk a Lie algebra alkalmazhatóságára $SE(3)$ -beli optimalizáció megvalósítására. Bemutattuk az BID-NeRF nevezetű algoritmusunkat, mely az INeRF továbbfejlesztett változata. Az optimalizáció veszteség-függvényébe bevezettünk egy mélység-információtól is függő regularizációs tagot, megmutattuk, hogy a térreprezentáció frekvenciájának csökkentésével jelentősen növelhető a relokalizáció konvergencia-tartománya és sebessége, továbbá leírtuk, hogyan lehet több képet is bevenni a relokalizációba.

Egy kivételével minden hipotézisünket teszteredményekkel támasztottuk alá majd végül bemutattuk, hogy mekkora mértékben sikerült a kiindulási algoritmus teljesítményét mind a konvergencia-tartomány méretében, mind az optimalizáció sebességében felülmúlni. Bár több kép használata nem bizonyult hatásosnak az optimalizáció robusztusságának növelésében a rendelkezésünkre álló adathalmaz alapján, valós környezetben jobb teljesítményt várunk.

Eredményeinkhez hozzátartozik, hogy a felvázolt algoritmus nagy kezdeti hibájú lokalizációra lett optimalizálva. Ideális esetben az BID-NeRF-el való relokalizáció után még sor kerül egy finomabb illesztésre is. Ezt a lépést az BID-NeRF egy módosított változatával is végre lehet hajtani.

Fontos kiemelni, hogy az algoritmus konvergenciatartományára kapott eredmények a NeRF modell terében értendők. Metrikus térbe nehezen átválthatóak a távolságok, hiszen virtuális objektumokat vizsgáltunk egy normalizált térben. Nehezen becsülhető az algoritmus teljesítménye valós környezetekben. A tesztekben felhasznált hibák hatását az 11. ábra szemlélteti a modell terében.

Végeredményben egy dense típusú relokalizációs eljárást kaptunk, amely figyelembe veszi a betekintési szöveget, kompakt térreprezentációt kínál kép-rekonstrukciós lehetőséggel és akár valós idejű optimalizációra is képes. A valós-idejűség biztosításához azonban az eredeti NeRF modellt le kell cserélni valamilyen hatékonyabb implementációra [22][23][24][32].

További lehetőségek

Kutatásunkat az algoritmus valós beltéri környezetekben való viselkedésének vizsgálatával tervezzük folytatni. A megközelítésben hatalmas potenciál rejlik, de először validálni kell, hogy nem csak szintetikus terekben működnek a leírt továbbfejlesztési módszerek.

Megbízható relokalizációs algoritmus birtokában felépíthető egy neurális térreprezentáció alapú SLAM algoritmus. Reményeink szerint így csökkenteni tudjuk a hagyományos SLAM algoritmusok reprezentációinak tárigényét és növelni tudjuk a

létrehozott térképek által megvalósítható relokalizációs algoritmusok konvergenciatartományát. A hagyományos SLAM megoldásokhoz képest az elképzelt algoritmusnak további előnye volna, hogy pixelszintű mélység szintézisre képes. Ennek birtokában szegmentálhatóak volnának azok az objektumok, melyek nem szerepelnek a térreprezentációban. Ez az információ felhasználható dinamikus objektumokkal való ütközés elkerülésére és a statikus térreprezentáció frissítésére, karbantartására. Elméletben hasonló eredmény érhető el, ha mélység-információk hiányában a relokalizációra használt sugarakra kapott gradenseket csoportosítjuk valamilyen klaszterező eljárással és kiszűrjük a szomszédos területekkel nem konzisztens gradensekkel rendelkező foltokat. A megközelítés magasfrekvenciás térreprezentáció esetén aligha szolgáltathat értelmes eredményeket, de érdemes lehet megnézni, vajon alacsony frekvenciás esetben kellően stabilak-e a kapott gradiensek.

Az INeRF [15] és a DS-NeRF [20] is megemlíti, hogy a tanító képek transzformációinak aktív optimalizációjával javítható a NeRF modellek rekonstrukciós képessége és csökkenthető a tanításukhoz szükséges idő. Ahogy a relokalizáció esetében is tettük, érdekes lehet megnézni, tudjuk-e segíteni a NeRF-k tanítását, ha egyszerre transzformáció illesztést és mélység-információn alapuló regularizációt is végzünk. Az INeRF a NeRF modellek tanítása során nem használ mélységadatokat, míg a DS-NeRF csak ritka regularizációt támogat. Becsléseink szerint RGB-D képek birtokában megvalósított pixel-szintű mélység regularizáció és lokális tangenstérbeli transzformáció optimalizációval még hatékonyabb tanítás érhető el.

Hivatkozások

- [1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015, doi: 10.1109/tro.2015.2463671.
- [2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-Time Single Camera SLAM,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007, doi: 10.1109/tpami.2007.1049.
- [3] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, “Dense Monocular Depth Estimation in Complex Dynamic Scenes,” Jun. 2016. Accessed: Oct. 31, 2022. [Online]. Available: <http://dx.doi.org/10.1109/cvpr.2016.440>
- [4] L. Valgaerts, A. Bruhn, M. Mainberger, and J. Weickert, “Dense versus Sparse Approaches for Estimating the Fundamental Matrix,” *International Journal of Computer Vision*, vol. 96, no. 2, pp. 212–234, Jun. 2011, doi: 10.1007/s11263-011-0466-7.
- [5] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM,” in *Computer Vision – ECCV 2014*, Cham: Springer International Publishing, 2014, pp. 834–849. Accessed: Oct. 31, 2022. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10605-2_54
- [6] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” Nov. 2011. Accessed: Oct. 31, 2022. [Online]. Available: <http://dx.doi.org/10.1109/iccv.2011.6126513>
- [7] J. Engel, V. Koltun, and D. Cremers, “Direct Sparse Odometry,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, Mar. 2018, doi: 10.1109/tpami.2017.2658577.
- [8] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, “From Coarse to Fine: Robust Hierarchical Localization at Large Scale,” Jun. 2019. Accessed: Oct. 31, 2022. [Online]. Available: <http://dx.doi.org/10.1109/cvpr.2019.01300>
- [9] J. Sivic and A. Zisserman, “Efficient Visual Search of Videos Cast as Text Retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 591–606, Apr. 2009, doi: 10.1109/tpami.2008.111.
- [10] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, Jan. 2022, doi: 10.1145/3503250.
- [11] H. Hirschmuller, “Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information.” Accessed: Oct. 31, 2022. [Online]. Available: <http://dx.doi.org/10.1109/cvpr.2005.56>
- [12] D. G. Lowe, “Object recognition from local scale-invariant features,” 1999. Accessed: Nov. 01, 2022. [Online]. Available: <http://dx.doi.org/10.1109/iccv.1999.790410>
- [13] E. Rosten and T. Drummond, “Machine Learning for High-Speed Corner Detection,” in *Computer Vision – ECCV 2006*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443. Accessed: Nov. 01, 2022. [Online]. Available: http://dx.doi.org/10.1007/11744023_34
- [14] Jianbo Shi and Tomasi, “Good features to track,” 1994. Accessed: Nov. 01, 2022. [Online]. Available: <http://dx.doi.org/10.1109/cvpr.1994.323794>
- [15] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, “iNeRF: Inverting Neural Radiance Fields for Pose Estimation,” Sep. 2021. Accessed: Nov. 01, 2022. [Online]. Available: <http://dx.doi.org/10.1109/iros51168.2021.9636708>
- [16] J. Sun *et al.*, “NeRF-Loc: Transformer-Based Object Localization Within Neural Radiance Fields,” *arXiv.org*, Sep. 24, 2022. <https://arxiv.org/abs/2209.12068>
- [17] Y. Lin *et al.*, “Parallel Inversion of Neural Radiance Fields for Robust Pose Estimation,” *arXiv.org*, Oct. 18, 2022. <https://arxiv.org/abs/2210.10108>
- [18] N. Rahaman *et al.*, “On the Spectral Bias of Neural Networks,” *arXiv.org*, Jun. 22, 2018. <https://arxiv.org/abs/1806.08734>

- [19] A. Vaswani *et al.*, “Attention Is All You Need,” *arXiv.org*, Jun. 12, 2017. <https://arxiv.org/abs/1706.03762>
- [20] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, “Depth-supervised NeRF: Fewer Views and Faster Training for Free,” *arXiv.org*, Jul. 06, 2021. <https://arxiv.org/abs/2107.02791>
- [21] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv.org*, Dec. 22, 2014. <https://arxiv.org/abs/1412.6980>
- [22] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, “FastNeRF: High-Fidelity Neural Rendering at 200FPS,” *arXiv.org*, Mar. 18, 2021. <https://arxiv.org/abs/2103.10380>
- [23] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–15, Jul. 2022, doi: 10.1145/3528223.3530127.
- [24] M. Tancik *et al.*, “Block-NeRF: Scalable Large Scene Neural View Synthesis,” *arXiv.org*, Feb. 10, 2022. <https://arxiv.org/abs/2202.05263>
- [25] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt, “Neural Sparse Voxel Fields,” *arXiv.org*, Jul. 22, 2020. <https://arxiv.org/abs/2007.11571>
- [26] T. Neff *et al.*, “DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks,” *Computer Graphics Forum*, vol. 40, no. 4, pp. 45–59, Jul. 2021, doi: 10.1111/cgf.14340.
- [27] J. Solà, J. Deray, and D. Atchuthan, “A micro Lie theory for state estimation in robotics,” *arXiv.org*, Dec. 04, 2018. <https://arxiv.org/abs/1812.01537>
- [28] P. J. Huber, “Robust Estimation of a Location Parameter,” in *Springer Series in Statistics*, New York, NY: Springer New York, 1992, pp. 492–518. Accessed: Nov. 01, 2022. [Online]. Available: http://dx.doi.org/10.1007/978-1-4612-4380-9_35
- [29] google, “GitHub - google/jax: Composable transformations of Python+NumPy programs: differentiate, vectorize, JIT to GPU/TPU, and more,” *GitHub*. <http://github.com/google/jax> (accessed Nov. 01, 2022).
- [30] Contributors to Wikimedia projects, “Peak signal-to-noise ratio,” *Wikipedia*, Aug. 22, 2022. https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio (accessed Nov. 01, 2022).
- [31] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, “Neural Scene Graphs for Dynamic Scenes,” *arXiv.org*, Nov. 20, 2020. <https://arxiv.org/abs/2011.10379>
- [32] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, “NeRF++: Analyzing and Improving Neural Radiance Fields,” *arXiv.org*, Oct. 15, 2020. <https://arxiv.org/abs/2010.07492>
- [33] H. B. Mann and D. R. Whitney, “On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other,” *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, Mar. 1947, doi: 10.1214/aoms/1177730491.
- [34] B. Schlyvitch, “Untersuchungen über den anastomotischen Kanal zwischen der Arteria coeliaca und mesenterica superior und damit in Zusammenhang stehende Fragen,” *Zeitschrift für Anatomie und Entwicklungsgeschichte*, vol. 107, no. 6, pp. 709–737, Oct. 1937, doi: 10.1007/bf02118337.
- [35] M. A. Fischler and R. C. Bolles, “Random sample consensus,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.

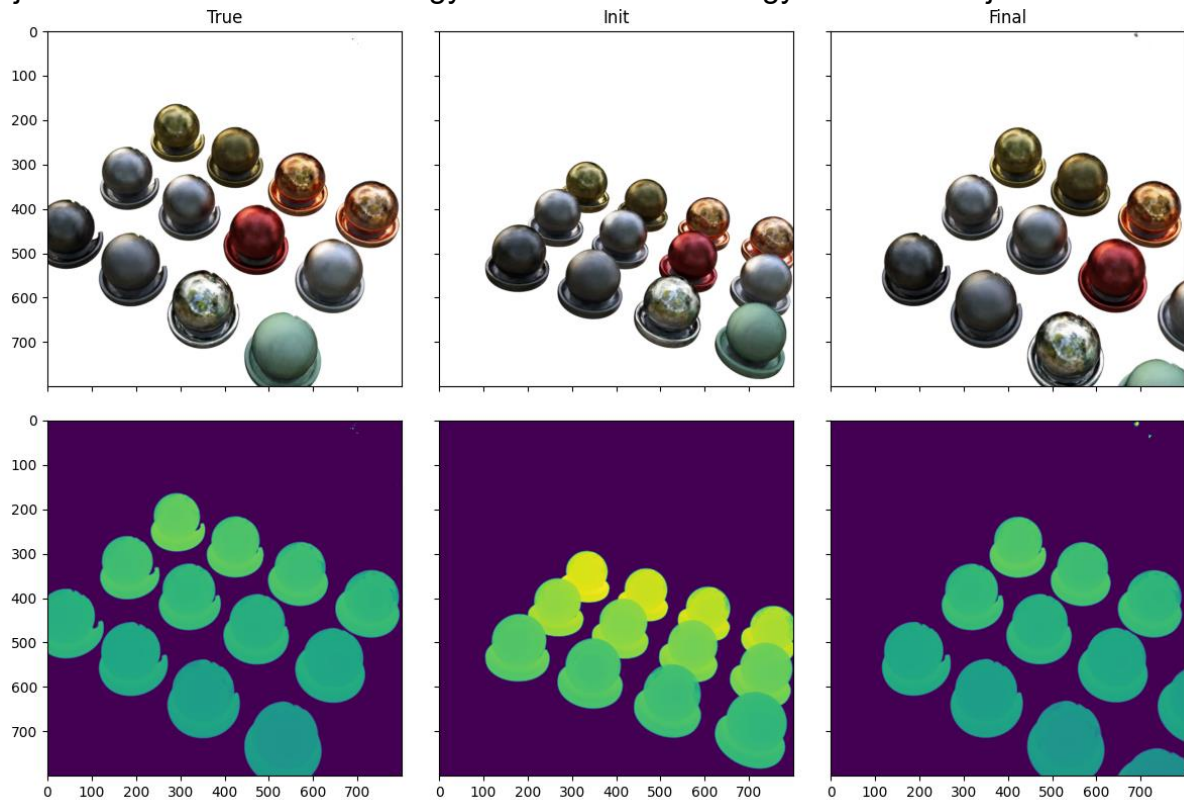
Tartalomjegyzék

Absztrakt	1
Bevezető	2
Probléma	5
Kapcsolódó irodalom.....	6
Neurális radiancia mezők	6
Inverz neurális radiancia mezők	9
Lie algebra.....	10
Bundle-adjusted inverse depth supervised NeRF	13
Mélységtérkép	13
Többképes optimalizáció	15
Durva felbontású modell.....	16
Sztocasztikus képtérbeli mintavételezés	18
Algoritmus	19
Implementáció.....	20
Eredmények.....	21
Adathalmaz	21
Mélység-információ hatása.....	23
Térreprezentáció frekvenciája	24
Optimalizálás több kép alapján.....	26
Összesítés.....	27
Összegzés	29
További lehetőségek.....	29
Hivatkozások.....	31
Tartalomjegyzék.....	33
Függelék	34

Függelék

Az angol nyelvű szakirodalomban a pose kifejezést használják egy objektum pozíciójára és orientációjára. Egy ilyen pose megfeleltethető egy SE3-beli merev transzformációval. Azért hívjuk merevnek, mert a transzformáció csak eltolást és forgatást tartalmaz, nyírást és skálázást nem. Jelen dokumentumban magyar megfelelő hiányában a pose kifejezés helyett a transzformációt használjuk pontatlanul, azaz transzformáció alatt csakis merev transzformációkat értünk.

Más idegen nyelvű kifejezések esetében, melyekre csak ügyetlen magyar nyelvű megfelelőt adhatunk, inkább maradunk az eredeti kifejezés használatánál. Ilyenkor a kifejezések első előfordulásánál igyekszünk értelmes magyarázatot adni jelentésükre.



Kipróbáltuk vajon lehetséges-e csak mélységképeken alapuló relokalizációt megvalósítani. A fenti képen látható egy érdekes eset, mikor a mélység-információk alapján egy lokális minimumba konvergálunk. A mélységképek megfelelő pixeli tökéletes fedésben vannak. Az azonos geometriájú objektumok ebben a szituációban nem különböztethetőek meg csak mélységinformációt felhasználva.