

2022 Match Move 3. mérföldkő

Vándor Norbert, Mosolygó Balázs, Amstadt Zita, és Félegyházi Máté

1. Feladatleírás

A feladatunk egy algoritmus megtervezése, elkészítése, és tervezése, amely segítségével egy valós kamera felvételbe be lehet majd szűrni egy tetszőleges virtuális geometriát, ami hihetően reagál a kamera mozgására, és a környezetére. A feladat nehézsége miatt számos megszorítással, és feltétellel fogunk dolgozni. Ezek a következők:

- A geometria a környezetéhez vett relatív pozíciója statikus, tehát a felvételen csak a kamera mozog
- A geometria nem reagál a környező fényhatásokra, nem lesz árnyékolva, és nem is vet árnyékot
- A felvétel előre fel lesz véve, az algoritmus nem lesz alkalmas valós idejű használatra
- A geometria kezdeti pozíciója előre, manuálisan meghatározott
- A geometria és a kamera közé a felvétel során nem kerül obstrukció

2. Az algoritmus összefoglalva

Az algoritmusunk nagyobb részre osztható:

Első lépésként meghatározzuk a videót felvevő kamera mozgását az ORB-SLAM2 [2] segítségével. Az algoritmus részletesebb leírását az előző mérföldkő során bemutattuk, így itt eltekintünk tőle. Az ORB-SLAM2 a videó analízise után biztosítja a végső keyframe-ek gyűjteményét, illetve az ezekben a pillanatokban általa azonosított kamera pozíciót, ezt felhasználva próbáljuk majd a kérdéses geometriát megfelelően transzformálni. Fontos kiemelni, hogy az ORB-SLAM2 futás közben optimalizálja a keyframe-ek számát, azaz elveti azokat, amelyeket redundánsnak tart, vagyis túl hasonlóak egymáshoz.

A keyframe halmaz előállításához elengedhetetlen a felvevő kamera pontos kalibrációja, pontatlan kalibráció esetén az ORB-SLAM2 gyakran nem képes kellően elvégezni a szükséges pontmegfeleltetéseket, ezáltal nem lesz képes inicializálni a kamera pozícióját, ami üres kimenethez, és egy feldolgozatlan videóhoz vezet. A kalibrációt videó alapján végeztük, így ugyanis könnyebben tudtunk nagy mennyiségű képhez jutni, illetve el tudtuk kerülni a telefonjaink videó és fényképező módjai közötti előfeldolgozási diszparitás okozta hibákat.

A keyframe-ek meghatározása mellett fontos előfeldolgozási feladatként elhelyeztük az objektumot az adott jelenetbe úgy, hogy végig látható lehessen. Az elhelyezést mindig a videó első képkockáján végeztük, arra való tekintet nélkül, hogy a geometria a videó fókuszpontjában legyen.

A második főbb lépés a kérdéses geometria betöltése, illetve helyes transzformálása a kamera aktuális pozíciója alapján. Az ORB-SLAM2 kimenetében a kamera pozíciója az inicializálás pillanatában vett helyzetnek transzformációival van megadva, így járható útnak tűnik, hogy ha a geometriát a kamerára alkalmazott transzformációk inverzének vetjük alá, akkor a térben felvett relatív pozíciója statikus marad, ezáltal elérve a kívánt hatást. Ennek a megközelítésnek egy konkrét implementációja elérhető GitHub-on¹. Egyelőre mi is elvetjük a nem keyframe szakaszait a vizsgált videónak, ugyanis a geometria a köztes pillanatokban felvett pozíciójának meghatározása nem javítana az algoritmus pontosságán.

3. Metrikák és tesztek összefoglalva

A második mérföldkő során meghatározott teszteket, és metrikákat megvizsgáltuk ezen alkalommal is. A tesztek elsősorban a mozgás, és a háttér komplexitására fókuszáltak, a metrikák pedig az emberi elképzelést foglalták számokba. Az első, egyszerűbben mérhető metrika azt határozza meg, hogy az adott képkockák hány százalékában van, egy "elfogadható" hibahatáron belül az objektum, ahhoz képest, amire számítanánk, abban az esetben, ha valóban a felvétel része lenne. A második is hasonló, ekkor a meghatározott keyframe-ekre manuálisan elhelyezett "ground truth" állapothoz hasonlítjuk mátrix normák segítségével.

A tesztelt videókat kibővítettük olyanokkal, amelyek mozgási, illetve környezeti komplexitása meghaladta azt, amit a korábbi mérföldkőre elő tudtunk állítani. Habár ezekhez nem biztosítunk közvetlen összehasonlítást, hiszen a korábbi gyengébb algoritmusunk eredményeit nem mértük újra az új felvételen, a teljesítmény javulás elég szignifikáns ahhoz, hogy ne legyen rájuk szükség. Az új videók a Visual-Inertial Datasetből [1] származnak.

¹<https://github.com/ChiWeiHsiao/Match-Moving>

4. Alkalmazott javítások

Az előző mérföldkő végén a következő javítási lehetőségeket vetettük fel:

- A mono kameráink kimenetét RGBD-ként átadni az ORB-SLAM-nek
- Az ORB-SLAM covisibility gráf optimalizálási funkciót kikapcsolni
- Korábbi framek mentén limitálni a maximális transzformációt (eltolási távolság), aminek az objektum alávethető

4.1. RGBD kamera

A mai mobiltelefonok jelentős részén rendelkezésre áll mélységi információ, aminek segítségével különböző utófeldolgozási lépéseket végeznek. Ezt azonban nem mélységi kamerával, hanem mesterséges intelligenciás becsléssel állítják elő, illetve nem is teszik a felhasználó számára elérhetővé.

Az eredeti terveink szerinti megközelítés, miszerint egy konzisztens mélységet rendelünk a különböző pixelekhez, ezáltal egy teljesen lapos, ugyanakkor színes felszint szimulálva nem bizonyult alkalmasnak, hiszen az extra mélységi információ megzavarta a követést. A pontos okok feltárása előtt sikeresen átálltunk az eddig tárgyalt ORB-SLAM2-ről ORB-SLAM3-ra, ahol lehetőség van mono kamera esetén is minden képkockára megkapni a kamera becsült pozícióját. Ahogy a későbbiekben tárgyalni fogjuk, az ORB-SLAM3, illetve néhány technikai javítás ötvözete kellően pontos eredményhez vezetett, amin a fent említett módszerek nem javítottak tovább.

4.2. Covisibility gráf optimalizálás kikapcsolása

Ahogy az előző mérföldkő során említettük, az ORB-SLAM elsősorban valós idejű SLAM eszközként használatos. A futása során karban tartott covisibility gráf segítségével határozza meg a kamera pozícióját. A gráf rendszeres ritkításával képes a valós idejű követésre, azonban a futása végén csak a gráf végső, optimalizált verzióját bocsátja ki, ami ezáltal nem tartalmaz minden keyframe-t. Az optimalizálás kikapcsolásával hozzáférést nyerhettünk volna a teljes keyframe listához, ami simább végeredményhez, és potenciálisan pontosabb követéshez vezethetett volna. A műveletet megkezdtük végrehajtottuk, azonban abban az esetben, ha egyszerűen kikapcsoltuk az optimalizálást, akkor a kamera követés is leállt, ami lényegében azt jelentette, hogy nem kaptunk kívánt kimenetet.

A fent említett áttérés ORB-SLAM2-ről ORB-SLAM3-ra, azonban ezt a lépést is feleslegessé tette, hiszen az ORB-SLAM3 nem csak minden keyframe-t képes kiadni, hanem minden frame-re produkál egy becsült kamera pozíciót.

4.3. Transzformációs limit

Az utolsó javítási javaslatunkban arra a tapasztalásra alapoztunk, miszerint az objektum néhány esetben drasztikus pozícióváltáson esik át 2 keyframe között, ami látványosan helytelen eredményekhez vezetett. Az elképzelésünk lényege az volt, hogy különböző megfigyelések, és előfeltételek alapján limitáljuk a geometria maximális mozgását, ezáltal csökkentve az "ugrások" súlyosságát. Ez különösen a keyframe alapú kimeneinken látszik, amelyek kevés, és időben viszonylag távoli képkockák alapján készülnek. A megközelítés számos hibalehetőséget hordozott magában.

Mivel sikeresen áttértünk egy olyan megközelítésre, amely során minden képkockára konkrét követési eredményeink készültek, az implementáció csak ronthatott volna az eredményen. Vagy folytonosan számolta volna a várható elmozdulást, ami a hirtelen löket szerű felgyorsulásoknál produkált volna helytelen eredményt, míg máskor nem segített volna szignifikánsan, vagy valamilyen előre meghatározott érték alá kényszerítette volna az elmozdulást, amikor is lényegében manuális eredmény optimalizációt végeztünk volna.

4.4. Keyframe szám optimalizálás

Az ORB-SLAM működése, többszálúsága miatt, nem teljesen determinisztikus, ezáltal előfordul, hogy különböző számú keyframe-t állít elő. Mivel a projekt élettartamának jelentős részében a kiadott keyframekre hagyatkoztunk, fontos volt, hogy minél több, minél pontosabban meghatározott keyframe álljon rendelkezésünkre. Ezek kiszámítását egyszerűen úgy végeztük el, hogy számos alkalommal újravittük az ORB-SLAM2-t ugyan azokkal a bemeneti paraméterekkel, hogy kiválaszthassuk ezek közül azokat, amelyekben a legtöbb keyframe maradt a végső halmazban. A pontosság mélyreható tesztelését nem végeztük el, hiszen hamarosan áttértünk a frame szinten működő algoritmusra, amely mindezt redundánssá tette.

4.5. ORB-SLAM3

5. Eredmények

Hivatkozások

- [1] Klenk, S., Chui, J., Demmel, N., Cremers, D.: Tum-vie: The tum stereo visual-inertial event dataset. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp.

8601–8608. IEEE (2021)

- [2] Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)