# 2022 Match Move 2. mérföldkő

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

## 1. Frissített feladatleírás

A feladatunk egy algoritmus megtervezése, elkészítése, és tervezése, amely segítségével egy valós kamera felvételbe be lehet majd szúrni 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 következőek:

- 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 [1] 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 helyzetének 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-on[1]. 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. Tesztek

A tesztek tervezése során elsődleges szempont volt, hogy a lehető legszélesebb spektrumot fedjünk le. A kamera mozgási sebessége, iránya, a környezet megvilágítása, a jelenetben található tárgyak színei, formái, a geometria relatív pozíciója a kamerához, és számos további potenciálisan jelentős faktort próbálunk figyelembe venni, és lefedni. Természetesen a tesztjeink közel sem képesek a teljes probléma teret lefedni, azonban megpróbáltuk azonosítani azokat az eseteket, amik a lehető legnagyobb teret fedik le.

A tesztek során elsődleges célunk az algoritmus limitjeinek meghatározása volt, nem csupán az, hogy példákat gyűjtsünk helyes, és helytelen működésre. A megfelelő működést pontosan úgy definiáljuk, hogy a geometria egy vizuális hibahatáron belül helyezkedik el, azaz nem feltétlenül szükséges, hogy a geometria tökéletesen stabilan a megfelelő pozícióban maradjon, ez a különböző kerekítési hibák, és szükséges közelítések miatt szinte lehetetlen. Az helyes működés így könnyebben elérhető, ezáltal a határok viszont tágabbak, ami nehezebbé teszi a meghatározásukat.

[1] https://github.com/ChiWeiHsiao/Match-Moving

Az elsősorban figyelembe vett paraméterek, a kamera mozgásának komplexitása, a környezet komplexitása, és a megvilágítás volt.

A kamera mozgásának komplexitását 2 fő faktor mentén határozhatjuk meg. Az első, hogy a kezdeti pozícióból egy tetszőleges időpillanatban mennyire komplikált transzformáció során vihetnénk át. Tehát, ha például a videó során csak eltolások történnek, különösen, ha ezek egyetlen tengelyt követnek, az kevésbé komplikált, mintha a kamera forogna 1, vagy több tengely mentén.

A második, hogy a kamera által felvett pozíció mennyire eltérő transzformáció során jöhetett létre 2 tetszőleges időpillanat között. Ebben az esetben ha a kamera több tengely mentén forog/mozog, de egy irányba teszi azt konzisztensen, akkor azt állítjuk, hogy kevésbé komplex a mozgása, mintha például egy egyszerűbb utat járna be, de időközönként egy új véletlen irányba kezdene forogni.

A környezet komplexitása szintén potenciálisan több faktortól függ, a jelenlévő tárgyak számától, azok sarokpontjainak beazonosíthatóságától, és így tovább. Természetes közegekben ezt a komplexitást nehéz korlátozni, hiszen nehéz inkrementálisan csökkenteni egy fa asztal mintájának összetettségét, vagy egy pohár sarkainak kerekítettségének mértékét befolyásolni, így ezt a faktort kevéssé vizsgáltuk.

Fontos kiemelni, hogy egy feladat, ami komplexebb részekkel rendelkezik nem feltétlenül lesz nehezebb, mint egy olyan, ami egyszerűbb, hiszen például minimális környezeti komplexitás esetén az ORB-SLAM nem fog tudni kellő mennyiségű pontmegfeleltetést elvégezni ahhoz, hogy megfelelően tudja követni a kamerát.

## 4. Metrikák

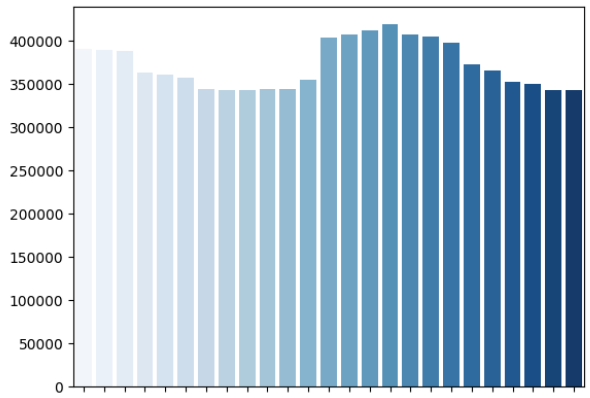
Annak érdekében, hogy számszerűsíteni tudjuk az algoritmusunk pontosságát 2 metrikát vezettünk be. Mindkét metrika során a keyframe-eket vizsgáljuk, hiszen jelenleg csak ezeket dolgozzuk fel.

Az első során egy bináris döntést hozunk arról, hogy az adott pillanatban helyesen számítottuk-e ki a geometriára alkalmazandó transzformációt, vagy sem. Ebben az esetben a videó keyframe-ein egyesével végigmenve a kiértékelő meghatározza, hogy az objektum egy elfogadható hibahatáron belül helyezkedik-e el vagy sem, azaz viszonylag közel van-e ahhoz a pozícióhoz, ahol lennie kellene, ha az algoritmus helyesen működne. A metrika több szempontból is pontatlan, hiszen nehéz objektív konzisztenciát elvárni egy emberi megfigyelőtől, abban az esetben, ha a transzformáció az esetek nagyobb részében közel helyes. Ilyenkor előfordulhat, hogy egyes képeken nagyobb eltérést fogadunk el, mint máshol, szimplán azért, mert "természetesebben néz ki", vagy mi magunk sem tudjuk pontosan eldönteni, hogy hogyan kellene kinéznie az adott jelenetnek helyes működés esetén. Az algoritmusunk jelenlegi verziója mellett ezek a problémák nem jelentősek, hiszen nagyon szignifikáns hibákat vét az esetek jelentős arányában, így egyelőre ez a durva metrika is használható. Abban az esetben, ha sikerül olyan magas szintet elérni az algoritmusunkkal, hogy a megfigyelő szubjektivitásának zavaró hatása jelentőssé válna, ez a metrika elhagyható, hiszen már elértünk egy "elég jó működést", ami után már csak mérőszámokon tudunk javítani.

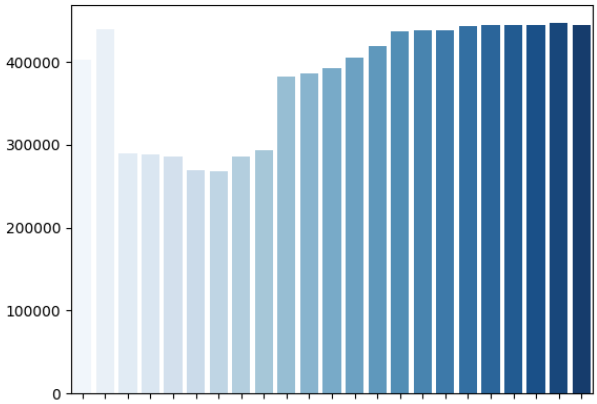
A második metrika is manuális értékelésen alapszik, azonban a kimenete már számszerűsített, így jobban használható az inkrementális javulások mérésére. A számításához egy kiválasztott keyframe halmazra, vagy akár az összes keyframe-re manuálisan meghatározzuk, hogy mi lenne az objektum helyes pozíciója, majd a transzformációs mátrixot, ami ehhez a pozícióhoz vezetett összehasonlítjuk a Frobenius mátrixnorma mentén azzal, amit a módszerünk határozott meg az adott képkockára. A normák különbségét használjuk, mint pontosság. Ebben az esetben a kézzel meghatározott pozíció pontossága egy határozott gyengepont, hiszen emberek számára is nagyon nehéz megmondani, hogy pontosan hol kellene egy objektumnak elhelyezkedni, az ez által bevezetett hiba azonban az előzőhöz hasonlóan akkor lenne szignifikáns, ha a módszerünk nagyságrendekkel szofisztikáltabb lenne. Jelenleg a távolság a "helyes" és a számított pozíció között elég nagy ahhoz, hogy az emberi hiba ne legyen komoly befolyással a pontosságra. Abban az esetben pedig, ha egy olyan pontra jutnánk, amikor ez egy szignifikáns probléma lenne, a módszer által meghatározott pozíciót használhatnánk kiinduló pontnak, a pontos helyzet meghatározásához, hiszen ebben az esetben már csak minimális javításokra lenne szükség.

A második metrika nem csak könnyebben számszerűsíthető, hanem könnyen egyszerűsíthető is, hogy csak bizonyos paraméterek változására legyen érzékeny. Elkészíthető egy forgás, vagy eltolás invariáns verzió, ami segítségével jobban szétbontható az egyes irányokban elért fejlődés.

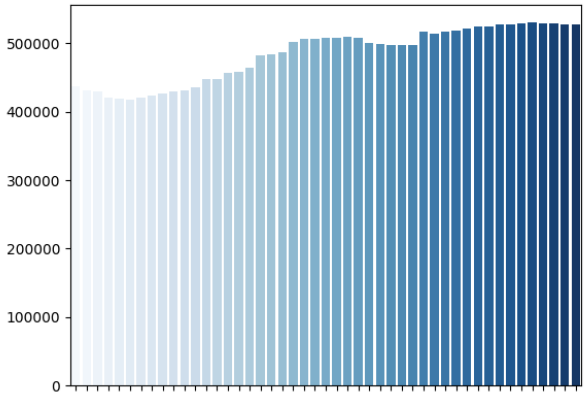
Mindkét metrika erősen emberi bemenetre hagyatkozik, azonban mivel ennél a problémánál a "helyes" transzformáció legegyszerűbben azzal definiálható, hogy mi tűnik "természetesnek" egy emberi megfigyelő számára, így ez szerintünk elkerülhetetlen.



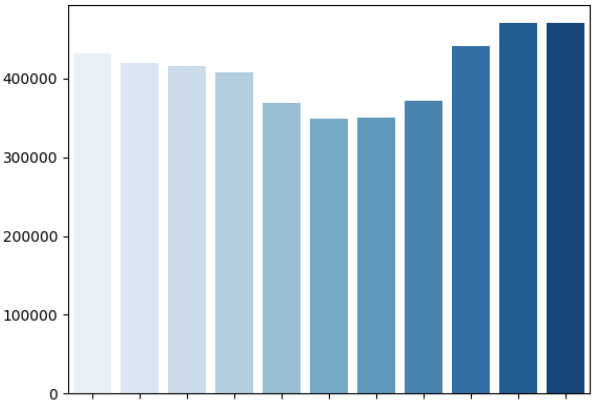

**(a)** Sok fény, egyszerű mozgás, nincs forgatás




**(b)** Közepes fény, komplex mozgás, kevés forgatás




**(c)** Sok fény, komplex mozgás, nincs forgatás




**(d)** Kevés fény, egyszerű mozgás, sok forgatás

**1. ábra.** Az algoritmusunk teljesítményének vizualizációja

# 5. Eredmények

Itt tárgyaljuk a kezdeti demó verzió által elért eredményeket, illetve a felfedezett gyenge pontokat.

## 5.1. Inicializáció

Az ORB-SLAM első lépésként megpróbál egy alkalmas ponthalmazt azonosítani, aminek segítségével képes lesz a kamera mozgását meghatározni. Ez a lépés nagyobb kihívást jelent, mint arra számítottunk, és a tesztjeink jelentős részén nem is sikerült. Ennek a hiányosságnak oka lehet, ha az adott kamera mátrixa helytelenül van meghatározva, ha a felvétel nem elég kontrasztos, számos egyéb lehetőség mellett. Azokat a teszteket, ahol nem sikerült az inicializáció nem vetjük el, hiszen a következő mérföldkő keretein belül ezt javítandó paraméterként tartjuk számon. Közel 30 teszt videó készült, 3 különböző kamerával. Ezek közül összesen 8 videóra sikerült inicializálnia az eszköznek, amik mind egy kamerával készültek.

Mind a 8 videó ugyan azzal a kamerával készült, így joggal gondolhatnánk, hogy a helytelen kalibráció okozta a nehézséget a többi esetben, azonban az ezzel az eszközzel készült felvételeken is nehézséget okozott ez a folyamat. A 3. mérföldkő során javítani tervezzük az inicializációs rátát.

### 5.2. Bináris metrika

Az algoritmus jelenlegi verziója nem produkál egyetlen képkockát sem, amin az objektum a kezdeti pozíciója hihető hibahatáron belül lenne. A geometria mozgásának mértéke lényegesen eltér a kameráétól, így a viszonylag egyszerű mozgásokat tartalmazó képsorozatokat sem követi helyesen.

### 5.3. Transzformációs különbség metrika

A 1. ábrán látható a 4 legtöbb keyframe-el rendelkező videó transzformációs mátrixok különbségének Frobenius normája. Ez az érték a két mátrix "távolsága", esetünkben tehát a módszerünk hibája. Az árbán látható, hogy amikor az ORB-SLAM sikeresen inicializál, viszonylag konzisztens pontosságot képes produkálni, tehát nincsenek nagyságrendbeli eltérések a legtöbb esetben az adott időpillanatok között.

A legnagyobb hibát azzal a videóval kapcsolatban tapasztaljuk, ahol viszonylag komplex mozgást végez kamera, és ebben az esetben készül a legtöbb keyframe is. A mozgás követése tehát nehézséget okoz, ami kiemelendő, hiszen a legjobb előrejelző faktor azzal kapcsolatban, hogy sikeres lesz-e az inicializáció a jelenet megvilágítása, azonban a méréseink alapján állíthatjuk, hogy a jelenet megvilágításának mennyisége, ha sikerül megkezdeni a követést nem annyira jelentős, mint a végzett mozgás komplexitása.

### 5.4. A manuális meghatározás hatása

Mivel a metrikákban meghatározott értékek egyénileg manuális lettek meghatározva, előfordulhat, hogy kisebb ingadozásokért az ellenőrző felelős, azonban a jelentős pontosságbeli ugrásokat közös vizsgálat során verifikáltuk. E-mellet a hiba nagyságrendje miatt, a kisebb hibák, amiket a feladat monotonitása okozna a manuális értékelés során, elfedődnek.

## 6. Potenciális javítási lehetőségek

3 javítási lehetőség fogalmazódott meg bennünk a mérföldkő folyamán:

- 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

Az ORB-SLAM a keyframek meghatározása közben minden képkockát elemez, ennek az elemzésnek az eredményét azonban Mono kamera esetén nem biztosítja a felhasználónak. A limitáció oka nem meghatározott, így potenciálisan áthidalható az által, hogy a kameráink RGB kimenetét kiegészítjük egy uniform mélységi értékkel, ezáltal nem módosítva az eredeti videó információ tartalmán.

Az ORB-SLAM elsősorban valós idejű SLAM eszközként használatos. Annak érdekében, hogy ebben az esetben használható legyen számos optimalizációt végez a működéséhez szükséges covisibility gráfon, amiket a gráf pontos funkciójával együtt, az előző beadandóban már megfogalmaztunk. Az optimalizáció során elhagy korábbi keyframeket, amiket redundánsnak talál, ez azonban a mi esetünkben komoly nehézségekhez vezet, hiszen így potenciálisan nagy ugrások történhetnek egy-egy jelenet között, ahol a kihagyott keyframe lenne. Ha minden keyframet meg tudnánk tartani fokozatosabb lenne az átmenet adott pozíciók között, ami ha önmagában nem is feltétlenül vezetne szignifikáns pontosságbeli javuláshoz, azonban a harmadik javítási lépés helyességéhez jelentősen hozzátenne.

Jelenleg előfordul, hogy egy-egy keyframen az általunk meghatározott pozíció "ugrik" egyet, tehát egy számítási pontatlanság miatt a két pillanatban felvett pozíció szignifikánsan eltér, annak ellenére, hogy a kamera mozgása, ezt nem indokolja. A hiba a keyframek menti számítások mellett nem terjed tovább, azonban a folytonosság illúzióját rombolja. A módszerünk azon a feltételezésen alapulna, hogy a kamera, így az általa felvett objektumok egy-egy időpillanat között, egy bizonyos távolságnál kevesebb mozgást végez, ami vagy kiszámítható a felvétel alapján (kamera mozgási sebesség, átlagos beesési szög változás stb.), vagy empirikusan meghatározható. A megjelenítési folyamat során számon tartanánk, vagy az objektum előző pozícióját, vagy azt a pontot, ahová a transzformáció átvitte volna, és a fentiek alapján meghatározott értékkel felülről korlátozzuk, amelyet `M`-el jelölünk majd. A két megközelítés közötti központi különbség, hogy ha az előző képkockán általunk meghatározott pozíciójához képest limitáljuk az elmozdulását `M`-el, akkor a limitáció tovább terjed a későbbi eredményekre is. Ebben az esetben, ha a kezdeti becsléseink helyesek, azaz `M` értéke nem túl szélsőséges, és az objektum első meghatározott pozíciója helyes, a geometria nem esne át drasztikus helyváltozásokon, így limitálható lenne a maximális hiba mértéke. Ha `M` értéke helytelenül lett meghatározva, azaz például túl alacsony, akkor nem fogja engedni az objektumnak,

hogy kövesse a kamera mozgásait, így egy folytonos csúszást bevezetve. Ezzel ellentétben, ha `M` csak az előző transzformációs eredményhez képest limitálná az elmozdulást, akkor az értéke kevésbé lenne kritikus, hiszen nem terjedne tovább az általa bevezetett hiba. Azonban, ilyenkor potenciálisan csak késleltetnék a kamera pozíciójának helytelen meghatározásából eredő hiba megjelenését, hiszen a következő képkockán már a drasztikusan hibás pozícióhoz képest limitálnánk a geometria elmozdulását. Lényegében azt korlátoznánk ekkor, hogy mennyit tudunk korrigálni a korábbi hibán.

## Hivatkozások

[1] 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)