

OBJEKTUM KÖVETŐ ALGORITMUS TELJESÍTMÉNYÉNEK VIZGÁLATA KÜLÖNBÖZŐ MOZGÁSMODELLEK ESETÉN

ANALYSIS OF PERFORMANCE OF OBJECT TRACKING ALGORITHM WITH DIFFERENT MOTION MODELS

Vincze Zsolt^{0000-0002-5813-3530, 1*}

¹Gépjárműtechnológia Tanszék, Közlekedésmérnöki és Járműmérnöki Kar, Budapesti Műszaki és
Gazdaságtudományi Egyetem, Magyarország
<https://doi.org/10.47833/2024.1.CSC.008>

Kulcsszavak:

objektum nyomkövetés,
Kálmán szűrés,
IMM szűrő

Keywords:

object tracking,
Kalman filter,
IMM filter

Cikktörténet:

Received
Revised
Accepted

Összefoglalás

A szerző a Matlab TrackerGNN nevű objektum nyomkövető algoritmusának performanciáját vizsgálja egy szimulált és egy valós mérésen alapuló adatszett segítségével. Az egyes objektum detekciók becsült és megfigyelt pozíciói közötti különbségek statisztikai vizsgálata is megtörtént. Az eredmények ismeretében a szimulált és a valós adatszett esetében a legjobban teljesítő szűrő típus meghatározásra került. A valós adatszethez választott szűrő típus alkalmazására került egy RTMaps Tracker komponens fejlesztése során.

Abstract

The authors make an analysis regarding the performance of the TrackerGNN object tracker function of Matlab software with a synthetic and real-life measurement based dataset. The statistical evaluation of the distances between the estimated and observed positions of the detected objects was performed. Based on the evaluation results the best performing filter types for the synthetic and the real-life-based data set were selected. The selected filter for the real-life-based dataset has been deployed in an RTMaps Tracker component.

1. Bevezetés

Amikor egy videofelvételen egy mozgó gyalogos vagy jármű tűnik fel, akkor ezen objektumok pozíciója minden képkockán kissé máshol helyezkedik el az előző képkockához képest. A detektor által az adott képkockán felismert objektumok függetlenek a korábbi képkockákon felismertektől. Ennek következtében nincs információ arról, hogy mely korábban felismert objektum lett ismételtelen felismerve az aktuális képkockán, tehát az aktuálisan felismert objektumok korábbi pozícióira vagy korábbi felismerésére vonatkozó adatok nem megadhatóak. Példaként, ha egy piros autó halad át a képen, a szemlélő számára egyértelmű, hogy ugyanaz az objektum látható minden képkockán, ellenben a számítógép nem képes megteremteni ezt a logikai kapcsolatot csupán azzal, hogy minden képkockán felismeri az objektumokat. Az említett logikai kapcsolat megteremtésére sokféle objektum követési (object tracking) módszer lett kifejlesztve [1][2] így az objektumok nyomkövethetővé

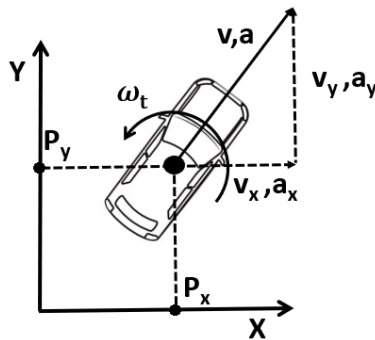
*Vincze Zsolt. Tel.: +36 1 463 1205;
E-mail cím: vincze.zsolt@kjk.bme.hu

válnak a felvételen. Ennek egyik fajtáját képezik a Kálmán-szűrésen alapuló trackerek, amelyek jól használhatóak video felvételeken [3][4][5], főleg, ha pozíció koordinátákat dolgoznak fel [6][7]. A tracker fő feladata a nyomvonalak (track) létrehozása, karbantartása és törlése a felismert objektumok pozíció információi alapján. A trackelés során információkat gyűjthetünk egy objektumról: hányszor lett felismerve korábban, mi volt előzőleg a pozíciója, illetve a nyomvonalak is elmenthetők. A tracker kezelni tudja azon eseteket is amikor egy képkockán bizonyos objektum nem lett felismerve, vagy fals-pozitív detekció adódott.

A Matlab program több beépített tracker funkciót is tartalmaz, amelyek az alkalmazott összerendelési algoritmusban térnek el egymástól. Az egyik ilyen a Globális Legközelebbi Szomszéd (Global Nearest Neighbour) algoritmus [8], amely meghatározza a nyomvonalak és a detektált pozíciók közötti távolságokat, majd ez alapján rendeli össze a nyomvonalakat és a detekciókat. A cikk fókuszában a Matlab TrackerGNN algoritmusának vizsgálata áll, különböző mozgásmodellek használata és konfigurációs beállítások mellett.

2. A pozíció Kálmán-szűrése

Az alkalmazott szűrő az objektum pozícióját és az 1. ábrán jelölt dinamikai paramétereit egy állapotvektorként reprezentálja, majd egy mozgásmodell felhasználásával ad becslést az aktuális lépésre vonatkozó állapotvektorra az előző lépésben ismert állapotvektorból kiindulva.



1. ábra. A középpontjával reprezentált jármű pozíció, sebesség, gyorsulás és szögsebesség paramétereiből áll össze az állapotvektor a mozgásmodelltől függően

Az aktuális lépésre vonatkozó becsült állapot \hat{x}_t , az alábbi formulával határozható meg, ahol w_t a különböző bizonytalanságokból eredő zaj.

$$\hat{x}_t = f(x_{t-1}) + w_t \quad (1)$$

A felhasznált mozgásmodellek közül legegyszerűbb a konstans sebesség (CV) modell, ahol a szűrő egy egyenes vonalú egyenletes mozgást feltételez. Bármilyen gyorsulás, vagy az egyenes pályától való eltéréstől adódó bizonytalanság zajként van kezelve. Egy objektum állapotát leíró vektor síkbeli mozgás esetén az alábbi módon írható fel: $[x, v_x, y, v_y]$.

Az x, v_x, y, v_y változók az x és y koordinátákat, és az x és y irányú sebességeket reprezentálják. Amennyiben a Gauss fehér zajt ami az x és y irányokban értelmezett bizonytalanságot reprezentálja $\eta_t = [\eta_{tx}, \eta_{ty}]$ és az eltelt időt T jelöli, a mozgásmodell felírható az alábbi egyenlettel [12, 13]:

$$\hat{x}_t = \begin{pmatrix} x \\ v_x \\ y \\ v_y \end{pmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{t-1} + \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ T & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \end{bmatrix} \eta_t \quad (2)$$

Az első modell kiterjesztése a konstans gyorsulás (CA) modell, ahol a szűrő egyenletesen változó függvényvel kezeli a sebesség változását. A bizonytalanság a változó gyorsulástól és az egyenes

vonulási pályától való eltéréseiből adódik, amely az irányonkra vonatkoztatott mérési zajhoz $\eta_t = [\eta_{tx}, \eta_{ty}]$ adódik. A mozgáshoz kötődő állapotvektor az x és y irányokra vonatkozó gyorsulásértékekkel a_x és a_y egészül ki: $[x, v_x, a_x, y, v_y, a_y]$.

A mozgásmodell az alábbi összefüggéssel adható meg [12, 13]:

$$\hat{x}_t = \begin{pmatrix} x \\ v_x \\ a_x \\ y \\ v_y \\ a_y \end{pmatrix} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{1}{2}T^2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_{t-1} + \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ T & 0 \\ 1 & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \\ 0 & 1 \end{bmatrix} \eta_t, \quad (3)$$

A harmadik gyakran használt mozgásmodell a konstans elfordulás (CT) modell. Ebben az esetben egy körpályán történő állandó sebességű elmozdulásból adódó egyenletes elfordulást feltételez a modell. A bizonytalanságból eredő zajt a körpályától való eltérés, valamint a sebesség változása adja. Ez a zaj adódik hozzá a mérési zajhoz. Az eredő zaj irányokra vonatkoztatott komponensei mellett a szögsebességre vonatkoztatott komponenssel is kiegészül: $\eta_t = [\eta_{tx}, \eta_{ty}, \eta_{t\omega}]$. Ilyenkor az állapotvektor a következő formában adható meg, ahol ω a szögsebességet jelenti: $[x, v_x, y, v_y, \omega]$.

Az egyenletes elfordulási mozgásmodell az alábbi formában adható meg, ahol ω_{t-1} a $t-1$ időhöz tartozó szögsebességet jelenti [12, 13]:

$$\hat{x}_t = \begin{pmatrix} x \\ v_x \\ y \\ v_y \\ \omega \end{pmatrix} = \begin{bmatrix} 1 & \frac{\sin(\omega_{t-1}T)}{\omega_{t-1}} & 0 & \frac{-1+\cos(\omega_{t-1}T)}{\omega_{t-1}} & 0 \\ 0 & \cos(\omega_{t-1}T) & 0 & -\sin(\omega_{t-1}T) & 0 \\ 0 & \frac{1-\cos(\omega_{t-1}T)}{\omega_{t-1}} & 1 & \frac{\sin(\omega_{t-1}T)}{\omega_{t-1}} & 0 \\ 0 & \sin(\omega_{t-1}T) & 0 & \cos(\omega_{t-1}T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_{t-1} + \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & 1 \end{bmatrix} \eta_t, \quad (4)$$

A megfigyelési modell az egyes objektumokat azok középpontjába zsugorítja és a megfigyelésekből származó adatok ezen pont koordinátái minden detektált objektum esetén:

$$z_t = H_t x_t + w_t. \quad (5)$$

Ahol w_t a megfigyelést terhelő zaj és H_k a megfigyelési mátrix amely megadja, hogy az állapotvektor mely komponenseit tartalmazza a megfigyelés. A megfigyelt komponensek minden objektum esetén a pozíció x és y koordinátái. Ebből következően a H_t mátrix a CV mozgás modellben alkalmazott állapotvektorhoz igazodva a következő lesz:

$$H_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6)$$

A Kiterjesztett Kalman-szűrő változattal (EKF) gyengén nemlineáris rendszereket is tudunk kezelni. Ennek alapötlete a munkapont körüli linearizálás, ahol a mátrixok szerepét a lineáris rendszerben a nemlineáris függvény állapotváltozók szerinti deriváltjainak mátrixa veszi át [14].

Az állapotvektorok meghatározását követően a tracker hozzárendeli a detekciókat a nyomvonalakhoz, majd frissíti a nyomvonalhoz tartozó állapotvektort a hozzárendelt detekció pozíciója alapján.

Blom [9] bemutatta az Interactive Multiple Model (IMM) szűrő algoritmust amely több mozgásmodellt egyidejűleg képes figyelembe venni. A szűrő becslést ad arra, hogy melyik mozgásmodell írja le leginkább az objektum pillanatnyi mozgását. A folyamat közben az egyes módok (mint konstans sebesség) szerinti elmozdulások valószínűségei valamint a módok közötti váltások valószínűségei meghatározásra kerülnek az utolsó ismert állapot alapján. Ezt követően (a módokhoz tartozó mozgásmodellek alapján) a módokhoz tartozó Kálmán-szűrők becslést adnak az állapotvektorra, majd minden szűrő esetében az állapotvektor frissítésre kerül az összes állapotvektor súlyozott összege alapján. Az összegzésnél használt súlyok a különböző módokhoz tartozó valószínűségekből adódnak. Ebből kifolyólag minél nagyobb egy adott módhoz tartozó valószínűség, annál inkább befolyásolja a végeredményt. A nagyobb valószínűségű módokhoz tartozó állapotvektorokat a kisebb valószínűségű módokhoz tartozó állapotvektorok is befolyásolják a valószínűségek arányában [10]. A szűrők

becslést adnak a módosított állapotvektorok alapján, majd a végeredmény a frissített becslések súlyozott összegéből adódik. Ennek a módszernek az az előnye, hogy a kis valószínűségű módok (amelyek nem jól írják le az adott objektum pillanatnyi mozgását) kimenete is frissítésre kerül, így egy a tényleges mozgásban bekövetkező modellváltás esetén az a modell amely immár legjobban leírja az aktuális elmozdulást is képes rövid időn belül jó becslést adni.

3. A TrackerGNN algoritmus tulajdonságai

A Matlab TrackerGNN funkciója a nyomvonal kezelések tekintetében több hasznos funkcióval bír. Az algoritmus a "Konfirmált" állapotot akkor rendeli egy nyomvonalhoz, ha előre meghatározott számú alkalommal kerül detekció hozzárendelésre a nyomvonalhoz adott számú lépésen belül. Például az algoritmus minden olyan detekcióhoz létrehoz egy nyomvonalat amely nem lett egy már létező nyomvonalhoz társítva, ugyanakkor a konfirmált jelzőt csak az a nyomvonal kapja meg, ahol az elmúlt 5 lépés esetében legalább 3 alkalommal detekció is lett a nyomvonalhoz társítva. Ezzel a megoldással az esetlegesen megjelenő fals detekciók által létrejövő rövid életű nyomvonalak kiszűrhetőek. Az összerendelési küszöb (assignment threshold) paraméter kezeli a különböző forrásokból eredő bizonytalanságokat az egymáshoz rendelési folyamatban. Nagy érték esetén egy nyomvonal és egy detekció akkor is egymáshoz rendelődik, ha messze esnek egymástól. A nyomvonal törlése a konfirmációs eljáráshoz hasonló folyamat. Ebben az esetben akkor törlődik a nyomvonal, ha adott számú lépésen keresztül kevesebb mint az előre meghatározott számú alkalommal kerül hozzárendelésre detekció az adott nyomvonalhoz.

A Kálmán-szűrés és a különböző mozgásmodellek segítségével a szűrő képes becslést adni az állapotvektorra egy tetszőlegesen megadott adott későbbi időpontban.

4. Különböző mozgásmodelleket alkalmazó szűrők vizsgálata

4.1. Teszt adatkészlet

A különböző mozgásmodelleket alkalmazó trackerek teljesítményének kiértékeléséhez két különböző adatkészlet lett felhasználva. Az első egy szimulált körforgalom, ahol öt különböző objektum (object 1 - object 5 a 2. és 3. ábrákon) halad két köralakú pályán, különböző sebességekkel. A szimulált adatkészlet rendelkezik néhány előnyös tulajdonsággal a valós életből származó adatkészlettel szemben: minden lépésben van detekció; a pozíciók nem terheltek mérési zajjal; a detekciók száma (ebből következőleg a nyomvonalak száma) állandó marad a szimuláció során. Az említett tulajdonságok hasznosnak bizonyultak a különböző mozgásmodelleket alkalmazó trackerek összehasonlítása során.

A második készlet egy valós mérésből származó adatszett. A felvétel egy olyan járművel készült melynek a tetején egy LiDAR szenzor helyezkedett el. A jármű egy parkolón ment keresztül. Egy LiDAR pontfelhőket feldolgozó detektor algoritmus a PointPillars [11] szolgáltatja a pozíció adatokat, a detektált háromdimenziós befoglaló dobozok (bounding box) alapján. A pozíciók egy két dimenziós $[x,y]$ koordináta vektorral voltak reprezentálva, amely a befoglaló doboz közepének talajra vetített pontjának felelt meg. Ez az adatkészlet a detektor bizonytalansága miatt komoly kihívást jelentett a tracker algoritmus számára. Az egyes pontfelhőkben számos alkalommal fordul elő nem detektált objektum. A jármű a parkolón való áthaladás közben számos manővert hajtott végre ezért az objektumok trajektóriái meglehetősen bonyolult formát vettek fel.

Mindegyik adatkészlet egy-egy .csv kiterjesztésű fájl, ahol az oszlopok az x koordinátát, y koordinátát, objektum típust, időbélyeget, lépés sorszámot (frame count) jelentik. Az objektum osztálya minden bejegyzés esetében 1 (személygépjármű).

4.2. Alkalmazott módszerek a szimulált adatszett esetében

A különböző mozgásmodelleket használó szűrők performanciájának kiértékelésére a nyomvonalhoz tartozó objektum $t + \Delta t$ időpontra vonatkozó becslült pozíció, valamint az ugyanezen időponthoz tartozó aktuális pozíció került összehasonlításra. A szimulációs adatkészletben a lépésköz 0,02s volt. A tracker esetében $\Delta t = 1s$, tehát 50 lépésnek megfelelő idővel későbbi pozícióra vonatkozó becslést ad. Az összehasonlítás során tehát adott időpontban az 50 lépéssel későbbi becslült pozíció valamint az 50 lépéssel későbbi tényleges pozíció került összehasonlításra a különböző mozgásmodellek valamint az IMM szűrő esetén.

A második kiértékelés során a szűrők aktuális lépésre tett pozíció becsléseit hasonlította össze az aktuális lépésben megjelenő detekciók pozícióival. Egy valós mérés esetében a mérési bizonytalanság miatt a pozíciók közötti különbség megnövekedne, a szimulált esetben nem jelenik meg ebből eredő bizonytalanság. Ennek az a következménye, hogy a pozíciók közötti eltérés csak a modell mozgásra történő illeszkedésének hibájából ered.

Az aktuális lépésre vonatkozó becslült és detektált pozíciókra vonatkozó eltérésekből (minden objektum esetében, az összes lépésben, az összes modelltípus esetében) egy adatkészlet is összeállításra került, amelynek statisztikai jellemzői: a középérték, a maximumérték és a szórás kerültek meghatározásra.

4.3. Alkalmazott módszerek a valós mérésen alapuló adatszett esetében

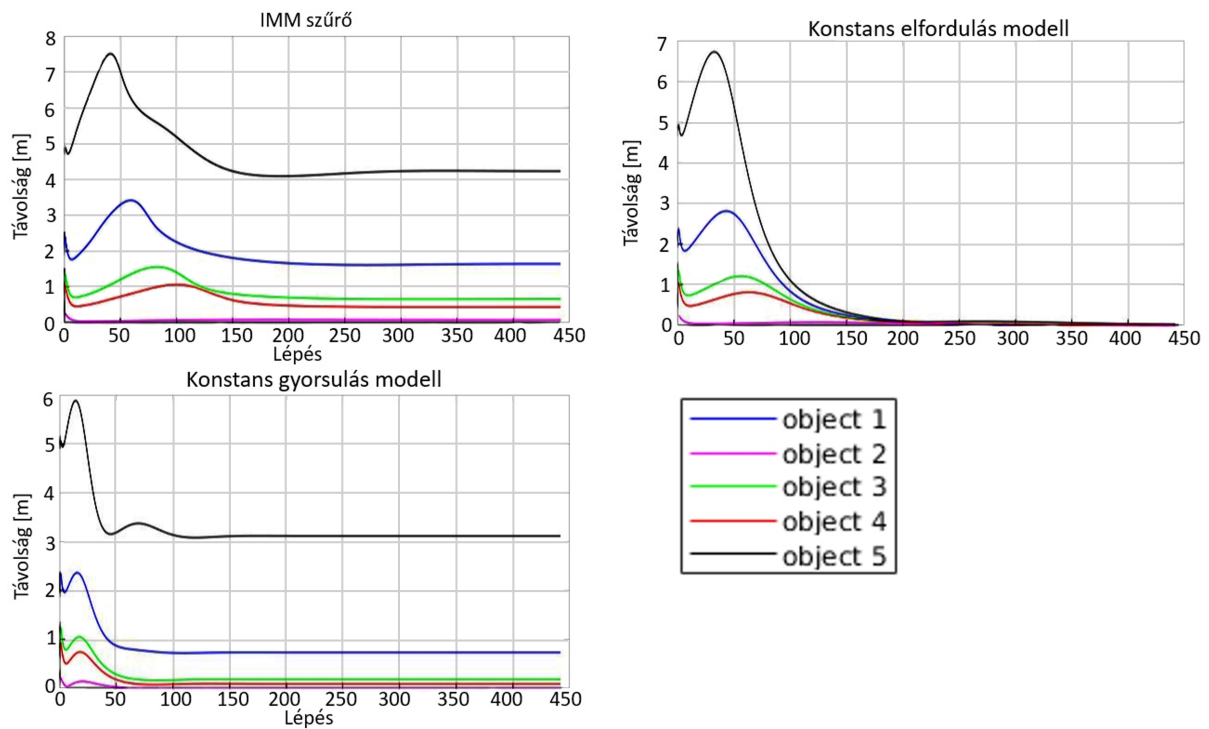
A valós mérési adat esetében (a mérési zaj és más zavaró tényezők miatt) a tracker nem volt képes megfelelő performanciát adni, a nyomvonalak nagyon rövid életűek voltak, és számosságuk nagyban változott. Emiatt a szimulált adatszethoz alkalmazott 50 lépéssel előre történő becslés összehasonlítását az akkori tényleges pozícióértékkel nem lehetett elvégezni. Az egyetlen kiértékelési lehetőség az aktuális lépésre vonatkozó becslések összehasonlítása a lépéshez tartozó detekciók pozícióival. Amikor a tracker egy új nyomvonalat hoz létre a más nyomvonalhoz nem rendelt detekcióknak, a nyomvonal pozíciója megegyezik a detekció pozíciójával, ennél fogva a köztük lévő távolság nulla. Ugyanakkor ez nem mond semmit a használt mozgásmodellről, mert még nem volt lehetőség becslést tenni. Az említett okok miatt ezen esetek nem lettek figyelembe véve a vizsgálat során. A vizsgált mozgásmodellek esetében az összes pozíció különbség érték (a kizárt eseteken kívül) rögzítve lett, melyeknek statisztikai jellemzői: a középérték, a maximumérték és a szórás kerültek meghatározásra.

5. Vizsgálati eredmények

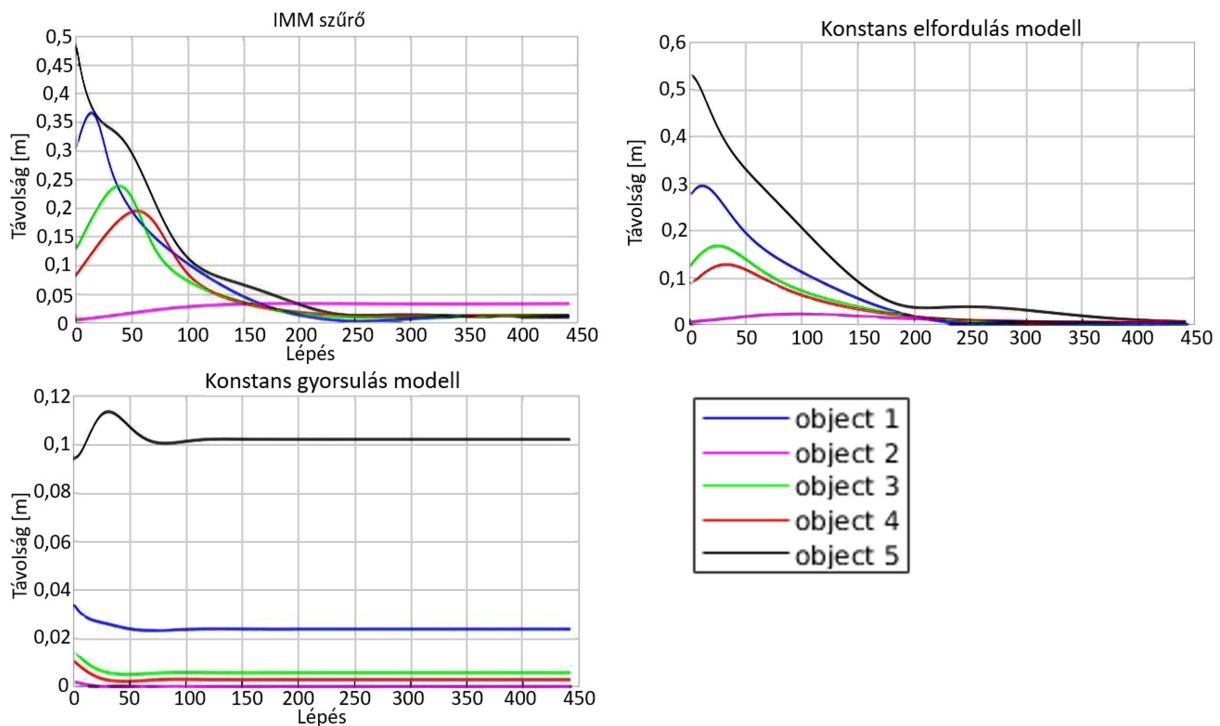
A szimulált adaszettel történő vizsgálat során a csak konstans sebességű mozgásmodelt használó szűrő nem volt képes az objektumok kör alakú mozgását kezelni, és minden lépésben új nyomvonalak keletkeztek és a összerendelés hiányában folyamatosan szüntek meg a régebben létrehozott nyomvonalak, ezért az objektumok nem voltak követhetőek. A többi esetben a tracker az elvárásnak megfelelően működött. Az eredményeket az 2. és 3. ábrák foglalják össze, míg a statisztikai eredményeket az 1. táblázat tartalmazza.

1. táblázat. A különböző mozgásmodelleket használó és az IMM szűrőknek a szimulált és a valós mérésen alapuló adatszettek statisztikai jellemzői

Adatszét	Szimulált				Valós mérésen alapuló			
	CV ekf	CA ekf	CT ekf	IMM	CV ekf	CA ekf	CT ekf	IMM
Középérték	0.9836	0.0313	0.0601	0.0637	1.6587	1.995	1.6494	1.335
Maximum	5.3862	0.3817	0.5314	0.5185	48.4691	54.1338	48.4691	50.2345
Szórás	1.0397	0.0448	0.0947	0.0894	3.5596	4.9683	3.5892	3.4733



2. ábra. A $t + \Delta t$ időponthoz tartozó becsült és tényleges pozíciók különbsége (IMM szűrővel bal felső, konstans gyorsulás modellel bal alsó és konstans elfordulás modellel jobb felső) a szimulált adatszett esetén



3. ábra. Aktuális lépésre vonatkozó becsült pozíciók és a detekciók pozíciói közötti különbség (IMM szűrővel bal felső, konstans gyorsulás szűrővel bal alsó és a konstans elfordulás szűrővel jobb felső) a szimulált adatszett esetén

6. Diskusszió

6.1. Szimulációs adatszettel történő vizsgálati eredmények kiértékelése

Az eredmények megmutatják, hogy a Kálmán-szűrő rendelkezik egy beállási idővel, amely eltelte után a pozíció különbség beáll egy konstans értékre. Az $t + \Delta t$ időponthoz tartozó becsült és tényleges pozíciók különbsége a konstans elfordulás modell esetében a legkisebb, ami nem meglepő, hiszen a szimulált adatszettben szereplő objektumok mozgására tökéletesen illeszkedik. A konstans gyorsulás modell kisebb eltérést ad, mint az IMM szűrő. Ennek oka az, hogy az IMM szűrőben megjelenő konstans sebesség komponens többlet hibát ad a becslés során.

Az aktuális lépéshez tartozó becsült és detektált pozíció eltéréseinek vizsgálatkor a beállási idő a konstans gyorsulás modell esetében volt a legrövidebb, de a maradó eltérés ebben az esetben vált a legnagyobbá a többi modellhez viszonyítva. A konstans elfordulás valamint az IMM szűrő esetében is a pozíció különbség majdnem nullára csökkent, ugyanakkor az IMM szűrőnél rövidebb volt a beállási idő. Ahhoz, hogy a pozíciókülönbség 10 cm alá csökkenjen, az IMM szűrőnek 110 lépésre azaz 2,2 s-ra volt szüksége.

A statisztikai vizsgálaton a legjobb eredményt a konstans gyorsulás modellt használó szűrő adta, ugyanis a beállási folyamat alatt a becsült pozíció jól megközelítette a valós pozíciót. Más szűrők esetében a beállási folyamat alatt az eltérés lényegesen nagyobb volt.

6.2. Valós mérésen alapuló adatszettel történő vizsgálati eredmények kiértékelése

Az eredmények alátámasztják azt, hogy a valós mérési adatszeten történő nyomkövetés sokkal nehezebb feladat a szimulált adatszeten megvalósított nyomkövetéshez képest. A kalkulált statisztikai mutatók jóval nagyobbak a valós esetben mint a szimulált adatszethnél. A különböző szűrőket tekintve az IMM szűrő adta a legjobb performanciát. A konstans elfordulás és konstans sebesség mozgásmodellt használó szűrők közel azonosan teljesítettek, a konstans gyorsulás alapú szűrő nem bizonyult hatékonynak ezen adatszett esetében.

6.3. RTMaps Tracker Komponens

Az RTMaps egy olyan szenzoradat-gyűjtő keretrendszer, amely az adatrögzítéskor biztosítja a különböző típusú szenzorok által szolgáltatott adatok közti szinkronitás megőrzését is, ezáltal a különböző frekvenciával érkező szenzoradatok visszajátszáskor is ugyan olyan ütemben töltődnek be. Emiatt a visszajátszott felvétel ugyan olyan minőségű, mint a tényleges mérés. A keretrendszer információ áramláson alapuló logika mentén működik ami lehetővé teszi különböző feldolgozó blokkok elhelyezését az adatáramlási csatornába. Ezen blokkok lehetnek a gyártó vagy harmadik fél által kiadottak, vagy pl. python nyelven implementált funkcionalitást biztosító felhasználói programok.

A vizsgálati eredményeket felhasználva valósult meg egy RTMaps tracker komponens fejlesztése is. A tracker először Matlab környezetben került implementálásra. A valós adatszett esetében legjobb performanciát adó IMM szűrő lett a komponenshez felhasználva. Ezt követően a nyomvonal kezelésre vonatkozó paraméterek finomhangolása után C++ kód lett generálva a gyorsabb működés érdekében. A generált C++ kód integrálva lett egy RTMaps komponens sablonba és ezt követően lett beforgatva a keretrendszerbe betölthető komponens.

7. Összefoglalás

A cikkben bemutatásra kerültek a detektált objektumok nyomkövetésére szolgáló, Kálmán-szűrőt használó algoritmusokban leggyakrabban alkalmazott mozgásmodellek. A különböző modelleket alkalmazó szűrők performanciája egy valós mérésen alapuló és egy szintetikus adatszett segítségével lett kiértékelve. Az összehasonlítás során a bemutatott mozgásmodellek mindegyikét együttesen

alkalmazó IMM szűrő algoritmus performanciája is kiértékelésre került. A vizsgálati eredmények tükrében a legjobban teljesítő változatok kiválasztásra kerültek a szintetikus és a valós mérésen alapuló adatszett esetében is. A valós mérésen alapuló adatszethez kiválasztott IMM szűrő egy RTMaps keretrendszerbe integrált nyomkövető algoritmus fejlesztésénél került alkalmazásra, mint a vizsgálatokkal igazolt legjobb performanciával rendelkező variáns.

Hivatkozások

- [1] Chee-Yee Chong. "Tracking and data fusion: A handbook of algorithms (bar-shalom, y. et al; 2011)[bookshelf]". In: IEEE Control Systems Magazine 32.5 (2012), pp. 114–116. DOI:10.1109/MCS.2012.2204808
- [2] Yilmaz, O Javed, and M Shah. "Object tracking: a survey'ACM computing surveys (CSUR)". In: (2006). <https://doi.org/10.1145/1177352.1177355>
- [3] Nam T Nguyen et al. "Multiple camera coordination in a surveillance system". In: ACTA Automatica Sinica 29.3 (2003), pp. 408–422. https://www.researchgate.net/publication/228959354_Multiple_camera_coordination_in_a_surveillance_system [Accessed: 07-Jun-2023]
- [4] T-H Chang and Shaogang Gong. "Tracking multiple people with a multicamera system". In: Proceedings 2001 IEEE Workshop on Multi-Object Tracking. IEEE. 2001, pp. 19–26. DOI: 10.1109/MOT.2001.937977
- [5] Xin Li et al. "A multiple object tracking method using Kalman filter". In: The 2010 IEEE international conference on information and automation. IEEE. 2010, pp. 1862–1866 DOI: 10.1109/ICINF-A.2010.5512258
- [6] Iván del Pino et al. "Low resolution lidar-based multi-object tracking for driving applications". In: Iberian Robotics conference. Springer. 2017, pp. 287–298. DOI:10.1007/978-3-319-70833-1_24
- [7] Taeklim Kim and Tae-Hyoung Park. "Extended kalman filter (EKF) design for vehicle position tracking using reliability function of radar and lidar". In: Sensors 20.15 (2020), p. 4126. DOI:10.3390/s20154126
- [8] Ito Wasito and Boris Mirkin. "Nearest neighbour approach in the leastsquares data imputation algorithms". In: Information Sciences 169.1-2 (2005), pp. 1–25. DOI:10.1016/j.ins.2004.02.014
- [9] Henk AP Blom. "An efficient filter for abruptly changing systems". In: The 23rd IEEE Conference on Decision and Control. IEEE. 1984, pp. 656–658. DOI: 10.1109/CDC.1984.272089
- [10] Roger Labbe. "Kalman and bayesian filters in python". In: Chap 7 (2014), p. 246. <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python> [Accessed: 07-Jun-2023]
- [11] Alex H Lang et al. "Pointpillars: Fast encoders for object detection from point clouds". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, pp. 12697–12705. <https://doi.org/10.48550/arXiv.1812.05784>
- [12] Blackman, Samuel S., and Robert Popoli. "Design and analysis of modern tracking systems." (No Title) (1999). ISBN: 9781580530064
- [13] Li, X. Rong, and Vesselin P. Jilkov. "Survey of maneuvering target tracking. Part I. Dynamic models." IEEE Transactions on aerospace and electronic systems 39, no. 4 (2003): 1333-1364. DOI:10.1109/TAES.2003.1261132
- [14] Dabóczi, Tamás. 2019. "Új Jelút-Kompenzációs Eljárások." MTMT : 30742762