

# Motion Tracker Beta: nyílt forráskódú „motion-tracking” szoftver

## Motion Tracker Beta: an open-source motion-tracking software

*FLOCH Kristóf<sup>1</sup> hallgató, Dr. KOSSA Attila<sup>2</sup> egyetemi docens*

Budapesti Műszaki és Gazdaságtudományi Egyetem, Gépészmérnöki Kar,  
Műszaki Mechanikai Tanszék,  
1111 Budapest, Műegyetem rkp. 3.  
www.mm.bme.hu

<sup>1</sup>kristof.floch@edu.bme.hu, <sup>2</sup>kossa@mm.bme.hu

### Abstract

*Motion-tracking is a widely used image processing technique, where the goal is to follow and register the movement of specified objects over time, from video recordings. Despite the increasing interest in the utilization of motion-tracking in engineering applications, there is currently no free and open source motion tracking software, which can satisfy the demands of the scientific community. This paper introduces Motion Tracker Beta: a GUI based open-source motion tracking application.*

**Keywords:** motion-tracking, motion-analysis, Python, graphical user interface, experimental mechanics

### Kivonat

*A „motion-tracking” egy széles körben alkalmazott képfeldolgozási technika, mely során a cél előre meghatározott objektumok időbeni mozgásának követése és rögzítése. Bár az eljárás iránti érdeklődés a mérnöki területeken is egyre növekszik, jelenleg nincs olyan ingyenes és nyílt forráskódú mozgáskövető szoftver, amely kielégítené a tudományos közösség növekvő igényeit. Jelen cikk a „Motion Tracker Beta” szoftvert mutatja be, amely egy grafikus felhasználói felülettel rendelkező nyílt forráskódú „motion-tracking” alkalmazás.*

**Kulcsszavak:** mozgáskövetés, mozgásanalízis, Python, grafikus felhasználói felület, kísérleti mechanika

## 1. Bevezetés

Az optikai eszközök folyamatos fejlődésének köszönhetően napjainkban egyre elterjedtebbek az olyan mérési eljárások, melyek során dedikált szenzorok alkalmazása helyett, a kísérletről készült videofelvételek feldolgozásával jutnak adatokhoz a mérnökök. A „motion-tracking” egy olyan folyamat, amely során a cél videofelvételekben előre meghatározott objektumok időbeni mozgásának követése és rögzítése. Ez a módszer, amellet, hogy kis költséggel jár, univerzálisan alkalmazható. Emiatt a kísérletek kivitelezése során is előnyös, ugyanis a kamerák a legtöbb esetben nem zavarják meg a mérési folyamatot, míg az objektumra szerelt dedikált szenzorok módosíthatják a mozgás dinamikáját.

Ennek köszönhetően a „motion-tracking” eljárás számos kutatási területen növekvő népszerűségnek örvend. Alkalmazták többek között mechanikai kísérleteknél, mint anyagvizsgálat [1], vagy rezgésanalízis [2], de biomechanikai méréseknél [3] is. Ezenfelül etológiai kutatásokban is szerepet kapott mely során állatok mozgását vizsgálták [4]. Ezek csak néhány kiragadott példák a számos alkalmazási lehetőség közül.

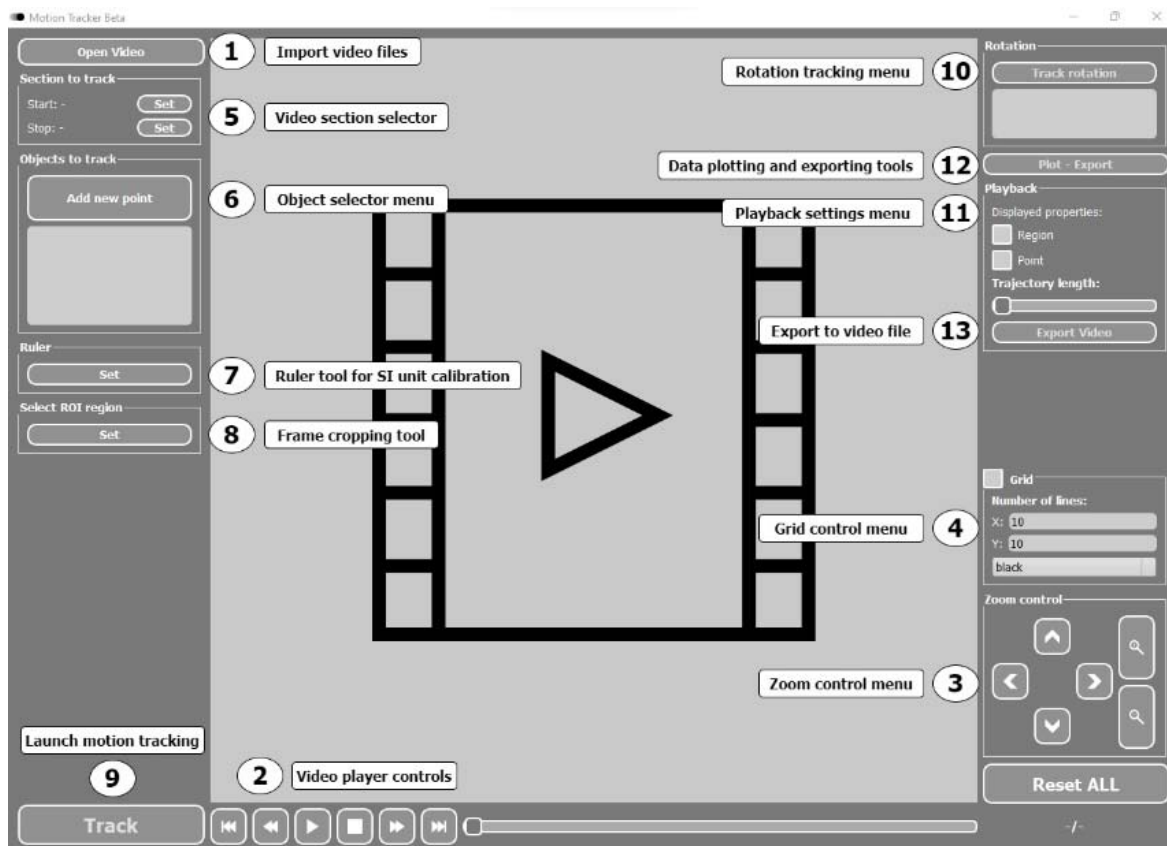
A növekvő népszerűség ellenére a videó alapú mozgásanalízis elterjedését nehezíti, hogy a felvételek feldolgozása sokszor körülményes és akár rendkívül időigényes is lehet. Ennek fő oka, hogy a mozgáselemzés megvalósításához rendelkezésre álló erőforrások napjainkban igen korlátozottak. Ugyan léteznek professzionális videovágó szoftverek (Adobe After Effects [5], Final Cut Pro [6]), melyekben objektumkövető funkciók vannak beépítve, ezek jellemzően magas licenc díjakkal rendelkeznek és használatuk is sokszor körülményes. Bár vannak nyílt forráskódú és ingyenes alternatívák is, mint például a Blender [7] vagy a DaVinci Resolve [8], ezen szoftverek a korábbi megoldásokkal szemben még korlátozottabb funkciókat biztosítanak. Az említett alapvetően videovágásra alkalmas szoftverek mellett azonban lehetőség van kifejezetten videó analízishez fejlesztett szoftverek alkalmazására is, mint a Xcitex ProAnalys [9], vagy a

Vicon Motus [10]. Ezeknél azonban szintén probléma a magas ár és a zárt forráskód, ami kevés teret hagy a testreszabhatóságra. Bár a bemutatott programoknak szintén ismert nyílt-forráskódú ingyenes alternatívája úgy, mint a Tracker [11], de ennek a szoftvernek a funkcionalitása meglehetősen hiányos és korlátozott. Éppen ezért számos kutatócsoport saját szoftveres megoldást dolgozott ki a „motion-tracking” megvalósítására. Ilyen például a pathtrackr [12] és a Tracktor [13] amelyek R, illetve Python programozási nyelv alapú könyvtárak, amelyek ugyan kellő testreszabhatóságot nyújtanak, az alkalmazásuk programozási előismeretekhez kötött. Emellett számos fejlesztés történt etológiai mozgásvizsgáló szoftverek területén [4], ezek funkciói azonban eléggé specifikusak.

Ebből a rövid összefoglalásból jól látszik, hogy a számos lehetőség ellenére jelenleg nincsen egy olyan, nyílt forráskódú, ingyenesen elérhető és grafikus felhasználói felülettel rendelkező mozgáskövető szoftver, amely ki tudná elégíteni a kutatók és a mérnökök igényeit precizitásával és funkcionalitásával. Jelen cikkben ezért egy olyan saját fejlesztésű „motion-tracking” szoftvert mutatunk be, ami kimondottan mérési és elemzési célokra készült, különös tekintettel a mechanikai kísérletekre, melyekre a korábban ismertetett programok igen kevés figyelmet fordítottak. Az elkészült szoftvert ingyenesen elérhetővé tettük, a forráskód és a felhasználói dokumentáció elérhető a következő linken: <https://github.com/flochkrstof/motiontracker>.

## 2. Szoftver felépítése

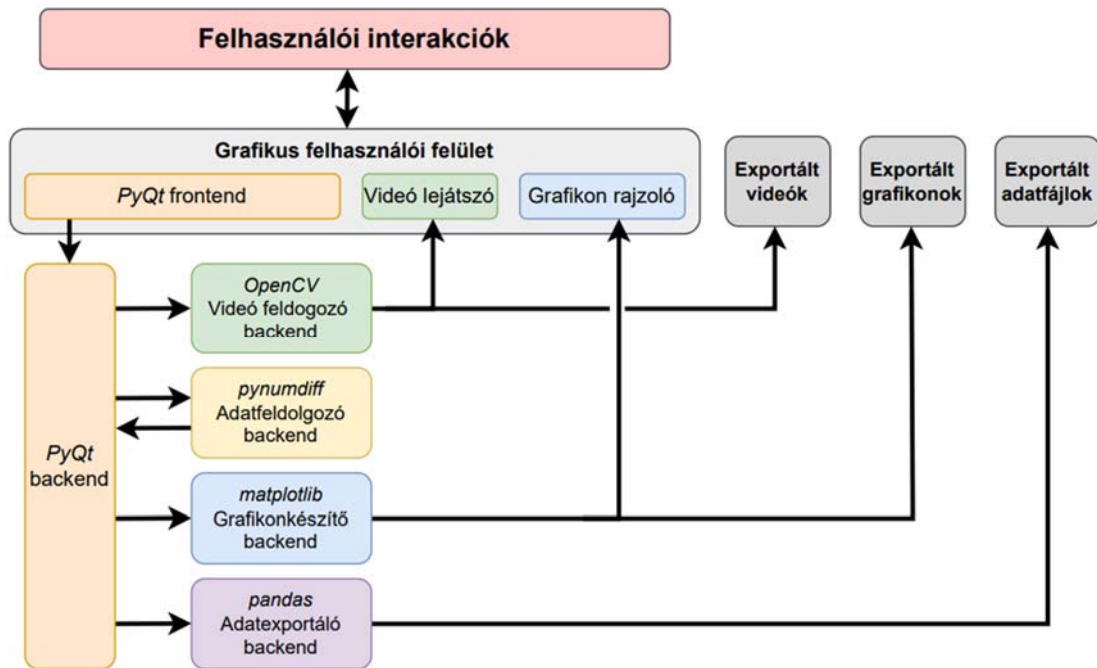
A szoftver megalkotása során fontos szempont volt számunkra, hogy egy könnyen használható, programozási előismeretet nem igénylő, platformfüggetlen lehetőséget kínáljon, melyben kompromisszumok kötése nélkül egy applikáción belül elvégezhető a mozgáskövetéshez kapcsolódó feladatok. Emellett, a nyílt forráskód révén testreszabhatósági lehetőséget is biztosítson, amennyiben az elérhető funkciókon bővíteni kívánnak a jövőbeni felhasználók. A szoftver kezelőfelületét az 1. ábra mutatja.



1. ábra. A Motion Tracker Beta szoftver kezelőfelülete.

A szoftver alapja a Python programozási nyelv, amely amellel, hogy garantálja a platformfüggetlenséget számos hasznos szoftverkönyvtárral és egyszerű telepítési és alkalmazási lehetőségekkel rendelkezik. A grafikus felhasználói felület a PyQt5 [14] keretrendszerben készült. A képfeldolgozási és objektumkövetési algoritmusok az OpenCV [15] könyvtárra épülnek. Továbbá az adatok feldolgozásához a PyNumDiff [16] numerikus differenciáló szoftvercsomagot alkalmaztuk. Végül fontos

megemlíteni a pandas [17] illetve a matplotlib [18] könyvtárakat, melyek az eredményként kapott adatok menedzselésében és exportálásában, valamint grafikonok készítésében kaptak szerepet. A szoftver felépítését és fő komponenseit az 2. ábra szemlélteti.



2. ábra. A Motion Tracker Beta szoftver felépítése.

A szoftver felhasználásának lépéseit, valamint az egyes beállítási, illetve konfigurációs lehetőségeket jelen cikk a terjedelembre való tekintettel nem fejt ki. Részletes információk és dokumentáció elérhető a szoftver GitHub oldalán (<https://github.com/flochkrstof/motiontracker>), emellett a funkciók bemutatásra kerültek korábbi munkánk során is [19].

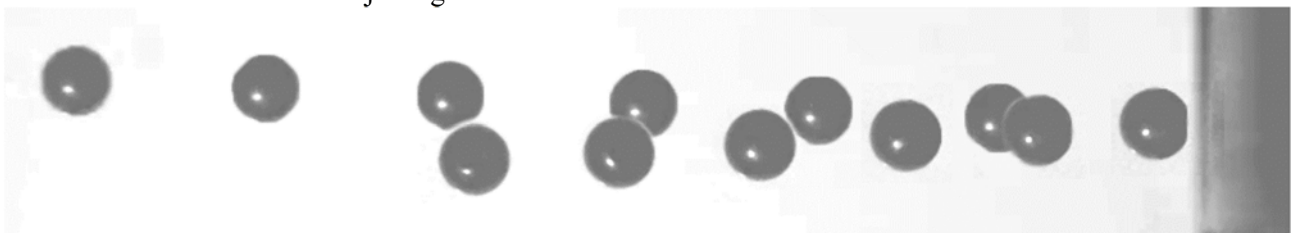
### 3. Alkalmazási példák

A következő alfejezetekben különböző alkalmazási mintapéldák szemléltetjük a szoftver alkalmazási lehetőségeit. Ezen példák mellett korábban a szoftver pontossága is validálásra került mesterségesen generált videófelvételek segítségével [19].

#### 3.1 Airsoft golyó becsapódásának vizsgálata

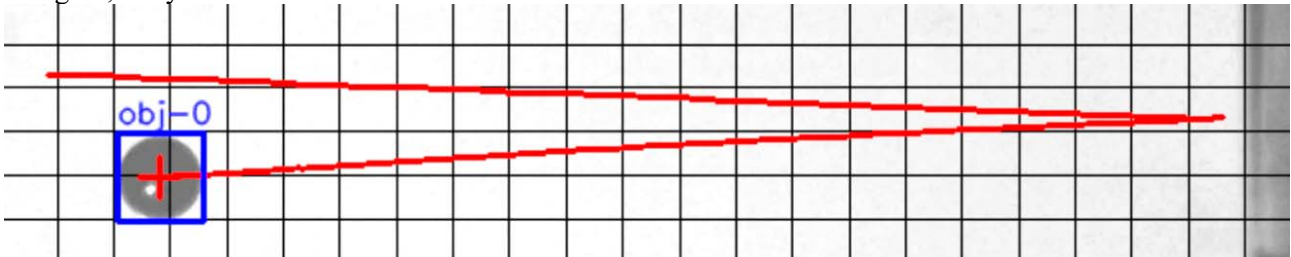
Ütközések és becsapódások vizsgálata az objektumokra szerelt szenzorokkal meglehetősen körülményes. Továbbá az airsoft golyók apró mérete ellehetetleníti a gyorsulásmérők alkalmazását. Jelen példában egy becsapódásról készült videót elemeztünk, melyben az objektumkövető algoritmusok segítségével rögzítettük a lövedék mozgását, majd numerikus differenciálás segítségével számítottunk sebesség és gyorsulás adatokat.

A lövedék mozgását és a 3. ábra szemlélteti több képkocka egymásra rajzolásával. Mivel a golyó átmérője ismert, lehetőség van a szoftverben SI mértékegységet társítani a felvételekhez, így a mért és számított adatokat is SI-ban adja meg a szoftver.



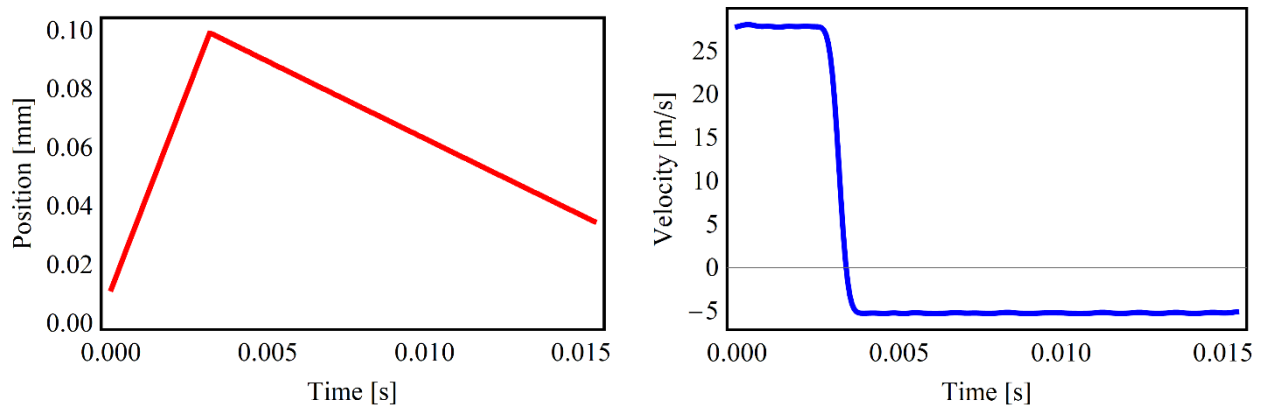
3. ábra. Az airsoft lövedék mozgása. A videó elérhető online: <https://tinyurl.com/airsoft-sample>.

A videó betöltése és az objektum kijelölése után fontos a megfelelő „motion-tracking” algoritmus kiválasztása. Jelen példában a BOOSTING algoritmust használjuk. Emellett a feldolgozáshoz fontos a megfelelő numerikus differenciálási módszer alkalmazása: a zajszűrés érdekében egy Savitzky-Golay filter alapú deriválást alkalmaztunk, melynek paramétereit a program automatikusan, optimalizáció útján határozta meg. Az algoritmusok futtatása után a felvétel visszajátszható, amelyen jól látszik a követett objektumok mozgása, melyet a 4. ábra szemléltet.



4. ábra. Az airsoft lövedék rögzített pályája.

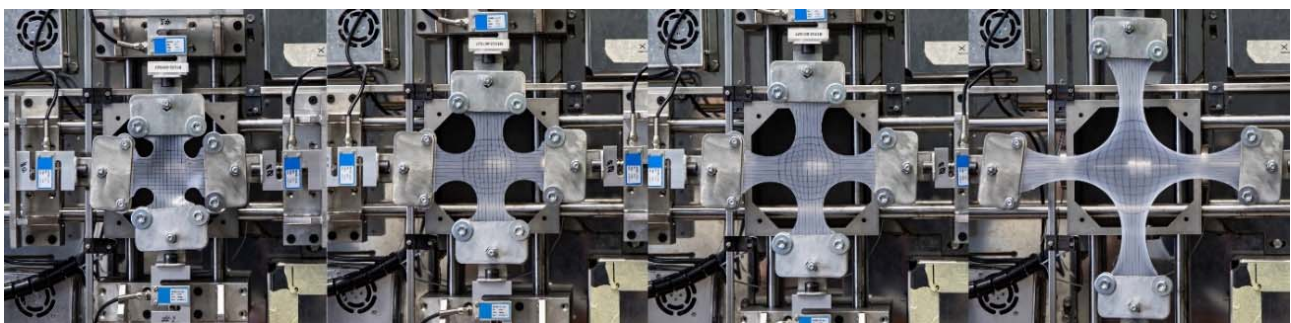
Ezt követően az eredmények kiértékelése következik, mely során jelen példában csak az  $x$  irányú elmozdulással foglalkozunk. A kapott pozíció, valamint sebesség adatokat az 5. ábra mutatja.



5. ábra. Az airsoft lövedék mért pozíciója (bal) és a számított sebessége (jobb).

### 3.2 Kéttengelyű húzókísérlet elemzése

A húzókísérletek fontos eszközei a mérnököknek az alapanyagok mechanikai modellezése során. Ebben a példában egy szilikon lemez kéttengelyű (biaxiális) húzókísérletét elemezzük „motion-tracking” segítségével. Képfeldolgozási eljárások alkalmazása ilyen jellegű méréseknél igen elterjedt. Ennek oka, hogy bár a legtöbb kísérletet végrehajtó gép rendelkezik megfelelő szenzorokkal az elmozdulás mérésére, ezek csak a befogás pontjáról tudnak pontos információkat szolgáltatni. Abban az esetben azonban, ha egy erősen nemlineáris anyagmodell felállítása a cél szükség lehet belső pontok viselkedésének ismeretére is, melyet az objektumkövetés segítségével könnyedén kinyerhetünk. Az bemutatott kísérlet folyamatát a 6. ábra mutatja, de a videó elérhető online is: <https://tinyurl.com/biaxial-sample>.

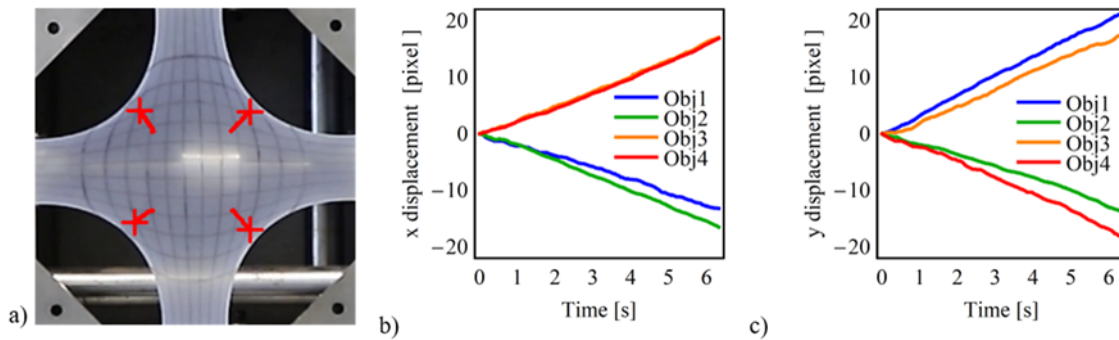


6. ábra. Az húzókísérlet lépései.

Jelen példában összesen négy különböző objektumot jelöltünk ki, melyeket a szoftver képes párhuzamosan, alacsony időigénnyel lekövetni. Ez komoly előny más feldolgozási módszerek, például a *digital image correlation* (DIC) technikával szemben, melynek hangolása és elemzése is időigényes. A



követőalgoritmusok közül ezúttal a CSRT módszert alkalmaztuk. Jelen helyzetben a sebesség, illetve gyorsulásadatok nem lényegesek, így a hozzájuk kapcsolódó beállítások sem relevánsak. A követés eredményét a 7.a ábra mutatja be.



7. ábra. a) A kéttengelyű húzó kísérlet elemzésének eredménye. Az objektumok elmozdulásának b)  $x$  és c)  $y$  irányú komponensei.

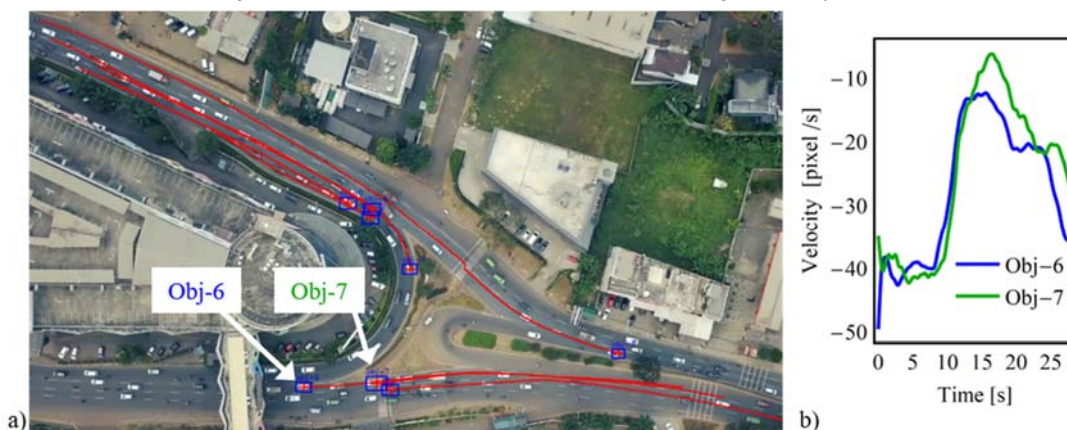
Az eredmények kiértékelése érdekében az elmozdulások grafikonon is ábrázolhatók. Ezt a szoftver beépített grafikonkészítő moduljával ismét könnyedén kivitelezhető. Az eredményeket külön  $x$  és  $y$  irányú elmozdulásokra lebontva a 7.b és 7.c ábra mutatja.

### 3.3 Gépjárműforgalom közlekedési csomópontokban

Az utolsó példában a mechanikai vizsgálatoktól egy kissé elrugaszkodva egy közlekedési csomópont vizsgálatának bemutatásával prezentáljuk a sokrétű felhasználási lehetőségeit az általunk fejlesztett szoftvernek. Jelen videó elemzésével egy drón által készített felvételtől nyerhetünk ki forgalomspecifikus információkat, így mint járművek egymáshoz képesti mozgását, illetve ez egyes sávok haladását. A videó a korábbiakhoz hasonlóan online is elérhető: <https://tinyurl.com/traffic-sample>.

A videóban a követésre több járművet is kijelöltünk. A kiválasztásnál fontos volt, nagy és jól látható autók mellett, kicsi, pár pixelből álló motorosokkal is tudjuk prezentálni, hogy a szoftver fenn tudja tartani az elvárt pontosságot. Továbbá figyeltünk arra, hogy egyes járművek fák, illetve egyéb zavaró tényezők alatt is áthaladjanak, tesztelve ezzel, hogy a program képes-e így is a hatékony követésre.

A követéshez ezúttal is a CSRT algoritmust alkalmaztuk. Az adatok feldolgozásánál egy medián-szűrővel simított véges differencia módszert használtunk 31-es ablakmérettel és 22 iterációval. Ahogy az a 8.a ábrán látható a szoftver ugyanolyan hatékonysággal képes követni a kis objektumokat, mint a nagyobb autókat. Továbbá nem zavarja össze az algoritmust, ha több ugyanolyan színű autó halad egymás mellett, illetve a fák és az útburkolati jelek okozta zavarásnak sincs hatása a teljesítményre.



8. ábra. a) A követett járművek pályája a közlekedési csomópontban. b) A 6-os és 7-es objektumok sebessége.

Emellett a feldolgozást követően az adatok exportálásával további elemzési lehetőségek adóttak. Megfigyelhető az egyes járművek átlagsebessége, sávokra lebontva, de vizsgálható egymás mögött haladó járművek követési távolsága is, melyekből a közlekedésmérnökök hasznos információkhoz juthatnak. A demonstráció céljából a 8. ábrán látható 6-os és 7-es jármű  $x$  irányú sebességprofilja van összehasonlítva egymással a 8.b ábrán.

## 4. Konklúzió

A „Motion Tracker Beta” egy saját fejlesztésű, nyílt forráskódú mozgáskövető alkalmazás, amely grafikus felhasználói felületének köszönhetően lehetővé teszi a gyors és egyszerű használatot előzetes programozási ismeretek nélkül. Az implementált objektumkövető és adatfeldolgozó algoritmusok széles választékának köszönhetően a szoftver különféle kutatási területeken (mechanikai kísérletek, biomechanika, etológia, kémia stb.) is használható. A hatékonyságot bizonyítja, hogy a „Motion Tracker Beta” már jelenleg is használatban van egyetemi hallgatók és oktatók által különböző kutatási projekteken. A szoftver Python nyelven íródott, kizárólag nyílt forráskódú könyvtárak felhasználásával, így használata ingyenes és bármely Python programozási nyelvet támogató platformra elérhető. Továbbá a szoftver a Python Package Indexben (PyPI) is elérhető, ahonnan több mint 3000 alkalommal töltötték le.

## Köszönetnyilvánítás

A szerzők köszönetüket fejezi ki az NKFIH FK 142457 projekt keretében kapott támogatásért.

## Irodalmi hivatkozások

- [1] Todros S., Pianigiani S., de Cesare N., Pavan P., Natali A. *Marker Tracking for Local Strain Measurement in Mechanical Testing of Biomedical Materials*. Journal of Medical and Biological Engineering 39. 2019. 764–772. old.
- [2] Morlier J., Michon G. *Virtual Vibration Measurement Using KLT Motion Tracking Algorithm*. Journal of Dynamic Systems, Measurement, and Control 132.1 2009.
- [3] Lugaresi C., Tang J., Nash H., McClanahan C., Uboweja E., Hays M., Zhang F., Chang C.-L., Yong M., Lee J., Chang W.-T., Hua W., Georg M., Grundmann M., *Mediapipe: A framework for perceiving and processing reality*, in: Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR), 2019.4
- [4] Panadeiro V., Rodriguez A., Henry J, Wlodkowic D., Andersson M. *A review of 28 free animal-tracking software applications: current features and limitations*. Lab Animal 50. 2021. 1–9. old.
- [5] Adobe Inc., *Adobe After Effects*. URL: <https://www.adobe.com/products/aftereffects.html> (2023.02.26.)
- [6] Apple Inc., *Final Cut Pro*. URL: <https://www.apple.com/final-cut-pro/> (2023.02.26.)
- [7] Blender Foundation, *Blender*. URL: <https://www.blender.org/> (2023.02.26)
- [8] Blackmagic Design Pty. Ltd., *DaVinci Resolve 18*. URL: <https://www.blackmagicdesign.com/products/davinciresolve> (2023.02.26.)
- [9] Xcitex Inc., *ProAnalyst 2023*. URL: <https://www.xcitex.com/proanalyst-motion-analysis-software.php> (2023.02.26.)
- [10] Contemplas GmbH, *Vicon Motus 10*. URL: <https://www.motus10.com/> (2023.02.26.)
- [11] Brown D., Christian W., Hanson R. M. *Tracker: Video Analysis and Modelling Tool*. URL: <https://physlets.org/tracker/> (2023.02.26.)
- [12] Harmer A., Thomas D. B. *pathtrackr: An r package for video tracking and analysing animal movement*. Methods in Ecology and Evolution 10.8. 2019. 1196–1202. old.
- [13] Sridhar V. H., Roche D. G., Giggins S. *Tracktor: Image-based automated tracking of animal movement and behaviour*. Methods in Ecology and Evolution 10.6. 2019. 815–820. old.
- [14] Riverbank Computing Limited. *PyQt5 Reference Guide*. 2016. URL: <https://www.riverbankcomputing.com/static/Docs/PyQt5/>. (2023.02.26)
- [15] Bradski G. *The OpenCV Library*. Dr. Dobb’s Journal of Software Tools. 2000.
- [16] van Breugel F., Liu Y., Brunton B. W., Kutz J. N. *PyNumDiff: A Python package for numerical differentiation of noisy time-series data*. Journal of Open Source Software 7.71. 2022. 4078. old.
- [17] The pandas development team. *pandas-dev/pandas: Pandas*. 1.4.2. verzió. 2022. URL: <https://doi.org/10.5281/zenodo.3509134>
- [18] Hunter J. D. *Matplotlib: A 2D graphics environment*. Computing in Science & Engineering 9.3. 2007. 90–95. old.
- [19] Floch K., Kossa A., *Motion Tracker Beta: A GUI Based Open Source Motion Tracking Application*. Submitted to SoftwareX. Available at SSRN: <https://dx.doi.org/10.2139/ssrn.4348616> ,2023