Motion Tracker Beta

User Guide

Version: *0.1.5*

Kristóf Floch

Department of Applied Mechanics,
Faculty of Mechanical Engineering,
Budapest University of Technology and Economics

MŰEGYETEM 1782

Contents

1 Introduction

Motion Tacker Beta is a open-source motion-tracking software developed for researchers and engineers who conduct experiments utilizing high-speed video cameras. This application assists them with the processing and analysis of the recorded footage. From mechanical experiments such as tensile testing or vibration analysis to biomechanical tests the goal was to develop an easy to use program that also meets all the demands and needs of engineers. Right now if somebody decides to use motion-tracking in his work, the following choices are available:

1. **License a professional video processing and editing software and use its built in tracking method.** However, this solution is not ideal since these applications usually come with a hefty price-tag. An another disadvantage could be the fact that these software are usually made for video editors, not engineers. Utilizing these software can be complicated and the results may not be satisfactory.

2. **Use a free motion tracking application**.The problem here is that these applications are usually extremely limited and cannot achieve acceptable results.

3. **Write custom code for a specific tracking job**. This solution can lead to sufficient results. On the other hand this method is excessively time consuming and hardly generalizable for different video inputs.

For this reasons we developed Motion Tracker Beta, a dedicated motion-tracking software available to anyone interested. By providing an easy-to-use graphical user interface, the goal was to simplify the tracking process as much as possible while maintaining a high quality of precision and speed.
The main features of this software include:

- The handling of the most common video file formats
- The possibility of selecting multiple objects to track
- The option to calculate the angular rotation between selected objects
- The utilization of multiple tracking algorithm from the OpenCV library
- The calibration of the video frames to get the results in SI units
- The selection of the starting and ending position of the tracking to analyze only certain parts of the video
- The option to calculate velocity and acceleration data from the recorded position using a diverse set of numerical differentiation algorithms
- The option to plot and export the results into different file formats
- The option to play back and export the video with the overlay of the trajectory of the tracked objects

The software was developed in Python but for the specific tasks many third party tools were adopted. The graphical user interface was created with the PyQt5 framework. For the handling of video files and to do the actual tracking the OpenCV library was used with its built in tracking algorithms. Numerical differentiations are carried out using the PyNumDiff package. Plots and figures are generated by matplotlib. For the complete list check the software's GitHub page.
At the end of this documentation we also provide some use-cases as well as potential areas where the application of this software can be useful.

2 Installation

2.1 PyPI installation – *Platform independent*

1. Install Motion Tracker Beta via pip:

   ```
   $ pip install motiontrackerbeta
   ```

2. Start the application from terminal with:

   ```
   $ MotionTrackerBeta
   ```

To use Chebyshev filters in the post processing, install the `pychebfun` package:

```
$ pip install pychebfun
```

2.2 Download and install with wheel file – *Platform independent*

1. Make sure you have Python 3.8.3 or later installed
2. Download the wheel file from the latest release
3. Install the wheel file with pip:

   ```
   $ pip install <path-to-wheel-file>
   ```

4. Start the application from terminal with:

   ```
   $ MotionTrackerBeta
   ```

2.3 Build from source – *Platform independent*

1. Make sure you have Python 3.8.3 or newer installed
2. Download source code from the latest release, or clone the GitHub repository:

   ```
   $ git clone https://github.com/flochkristof/motiontracker.git
   ```

3. Build the package using `poetry`:

   ```
   $ poetry build
   $ pip install dist/motiontrackerbeta-<version>-py3-none-any.whl
   ```

4. Start the application:

   ```
   $ MotionTrackerBeta
   ```

2.4 Download prebuilt binaries[1] – *Windows only*

1. Download the latest release
2. Unzip the downloaded folders
3. Run the software with `Motion Tracker Beta.exe`

2.5 Use the installer [1] – *Windows only*

1. Download the installer from the latest release
2. Run the installer, follow the instructions
3. After the installation has successfully ran, the software is ready to use

.

[1]Some function of the application are unavailable using the prebuilt binaries or the installer. To unlock the full potential of the software run it in a Python environment detailed in Section 2, 2.2 and 2.3

3 Usage

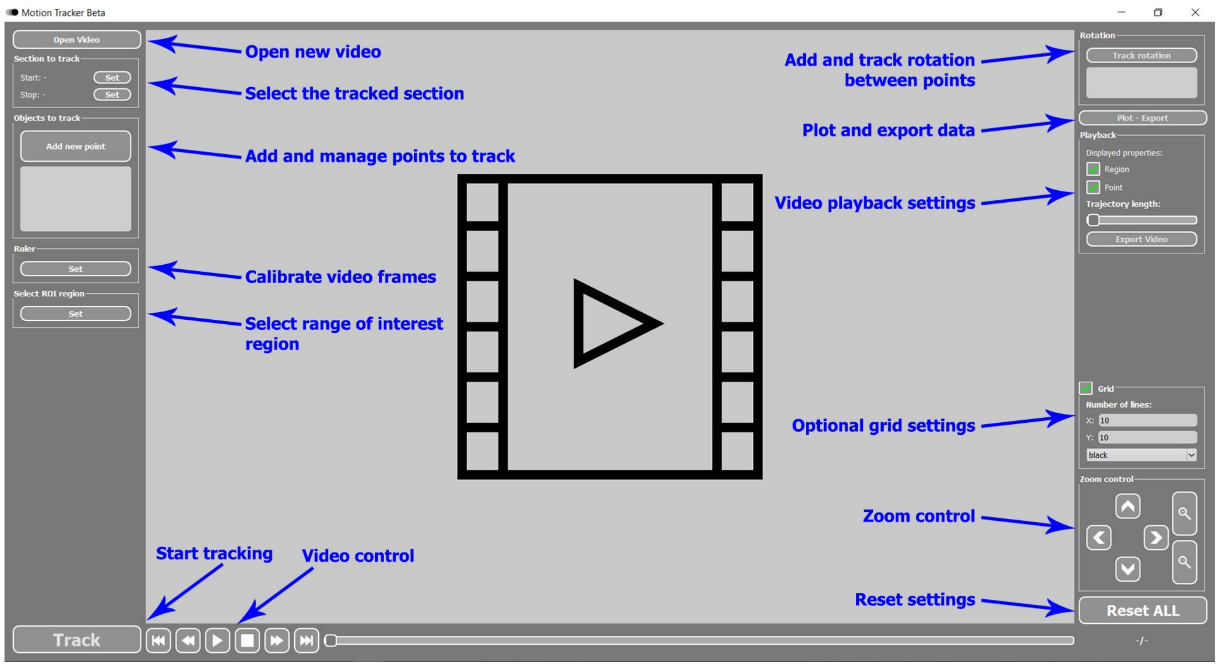


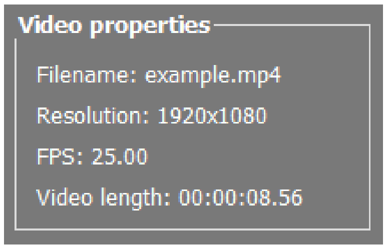
Figure 1: Graphical user interface

3.1 Opening and playing video files

Motion Tracker Beta is capable of handling the following file formats:

- MP4
- AVI
- MKV
- MOV

To start the analysis, users should click the Open Video button which then presents them with a file browser dialog where they can select the footage to be analyzed. After the video is opened, the most important properties of it are displayed in the application.



Video properties

Filename: example.mp4

Resolution: 1920x1080

FPS: 25.00

Video length: 00:00:08.56

Figure 2: Video properties

3.2 Navigation through the video

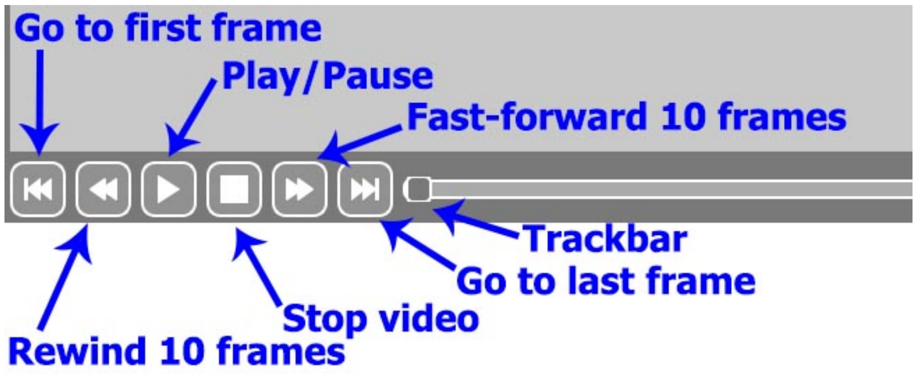


Figure 3: Navigation

The trackbar and the buttons at the bottom of the window can be used to start, pause, fast-forward and stop the video.

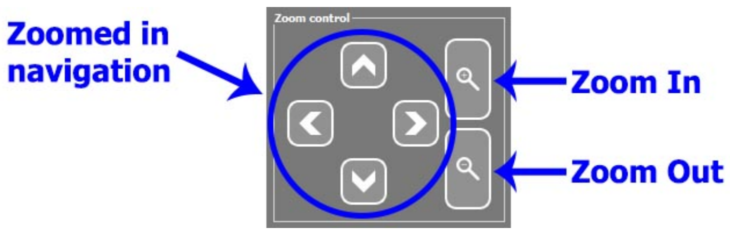


Figure 4: Zoom control

Users can also zoom in and out of the video any time with the help of the mouse wheel or the provided control buttons. For zoomed in navigation the W-A-S-D keys or the dedicated buttons can be utilized.

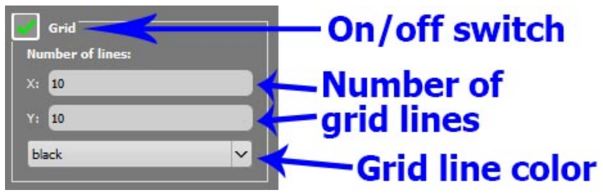


Figure 5: Grid settings

Furthermore, for more convenient navigation and point selection there is an optional grid overlay that can be turned on or off as needed. To be always clearly visible there are five different colors to choose from and users can also modify the density of the vertical and horizontal lines independently.

3.3 Select the starting and ending position of the tracking

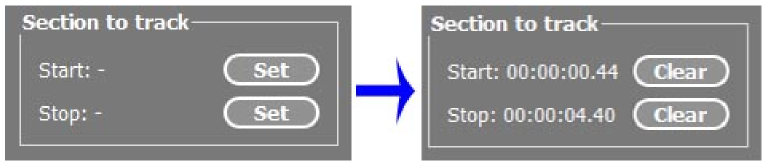


Figure 6: Selecting tracked section

This is a necessary step. The analysis cannot move forward until the starting and the ending frame are not selected. Use the trackbar and the control buttons to navigate to the desired time in the video and press the Set button. After the boundaries are successfully selected, users can no longer play back the full video, only the selected section.

3.4 Adding and removing Objects

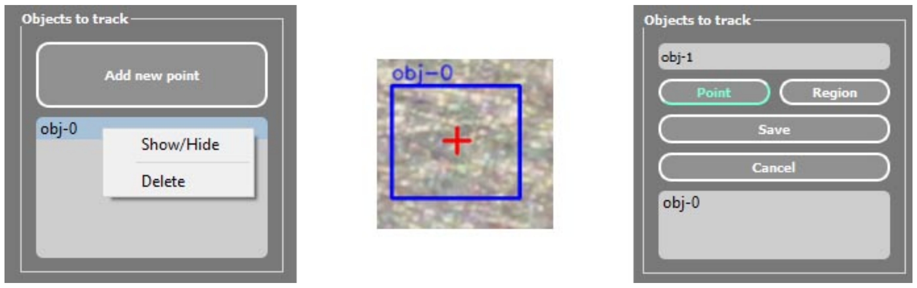


Figure 7: Managing tracked objects

The next step is to pick objects to track. Each object consists 3 important properties: name, point and region. The tracking algorithms are only capable of following regions but we are usually only interested in the movement of a certain point. Therefore users should pick a region to track but they are also required to select a specific point from that block. The position of this point will be the tracked property. Providing a name for the object is not necessary, but it could be convenient to use some custom identifier rather than the automatically generated ones.
After adding objects users can easily hide or delete them by clicking on the object name in the list. Note that a hidden object only disappears from the playback screen, the tracking and post processing algorithms will not ignore them. So, if an object is longer needed, its generally a better idea to delete it.

3.5 Calibrating with the Ruler

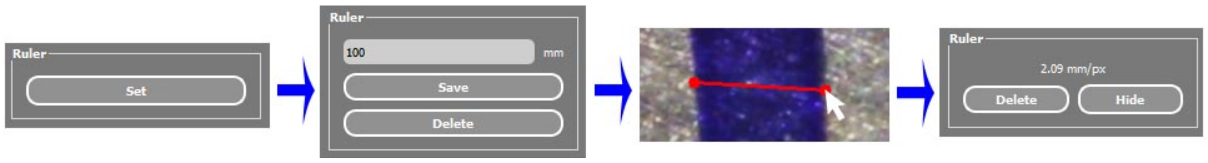


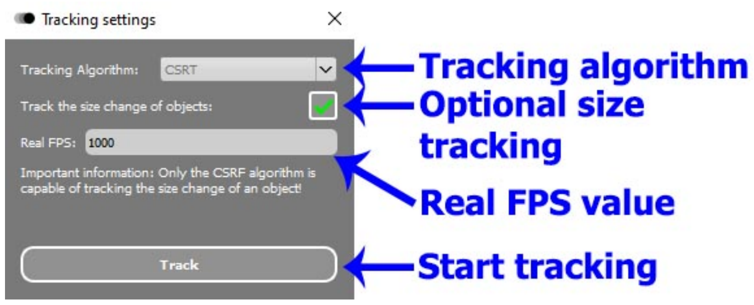
Figure 8: Managing tracked objects

Optionally, if users want to get the results not only in pixels but also in SI units like [m] or [mm] the video can be calibrated with the built in Ruler functionality. After clicking on the Set button users can use the left mouse to draw a line on the displayed video frame. Then they can provide the length of this line in the corresponding input field. After saving the Ruler the program will automatically convert the results into SI units.

3.6 Selecting the range of interest

With large, high-resolution video files the tracking algorithms can run significantly slower. By selecting a range of interest, the video frames can be cropped to speed up the tracking process. The algorithms will only scan the the outlined region which can help reducing the run time.

3.7 Tracking


Tracking settings

Tracking Algorithm: CSRT

Track the size change of objects: ✓

Real FPS: 1000

Important information: Only the CSRF algorithm is capable of tracking the size change of an object!

Figure 9: Tracking settings

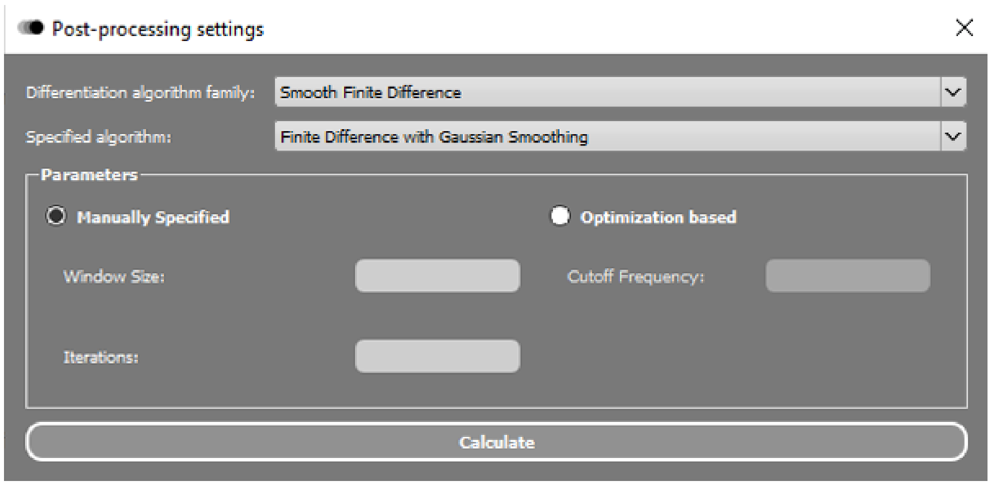
After the user clicks on the Track button a new window opens where they can choose from the implemented tracking algorithms:

- CSRT
- MOSSE
- BOOSTING
- MEDIANFLOW

- KCF
- TLD
- MIL

However, if the user wishes to track the size change of the objects only the CSRT algorithm is available. For further information on the tracking algorithms please refer to the OpenCV documentation.
The real frame per second value can also be provided. This is particularly important to reconstruct the actual time stamp values. If no data is given, the default FPS extracted from the video file will be used.

3.8 Post-processing


Post-processing settings

Differentiation algorithm family: Smooth Finite Difference

Specified algorithm: Finite Difference with Gaussian Smoothing

Parameters

[✓ Manually Specified] Optimization based

Window Size:

Cutoff Frequency:

Iterations:

Figure 10: Post processing settings

After the tracking has run successfully, the post-processing options will be displayed. Here users can select from a set of various numerical differentiation algorithms to calculate the velocity and acceleration of the tracked objects. The four families of methods implemented:

- Finite difference
- Smoothed finite difference using different smoothing techniques
- Linear models for local approximation
- Total variation regularization

Most of these methods require multiple input parameters that user can use to tune the performance and accuracy. However, the PyNumDiff library also implements a multi-objective optimization framework for selecting the proper values. Further information on the package can be found in the PyNumDiff documentation page. Here a detailed description about the optimization framework can also be found.

Note:
The optimization framework is only available if the application is running in a Python environment. In the complied versions this function cannot be accessed, users must tune the parameters manually.

The optimization is a computationally expensive task that can take several minutes to complete.

3.9 Rotation tracking

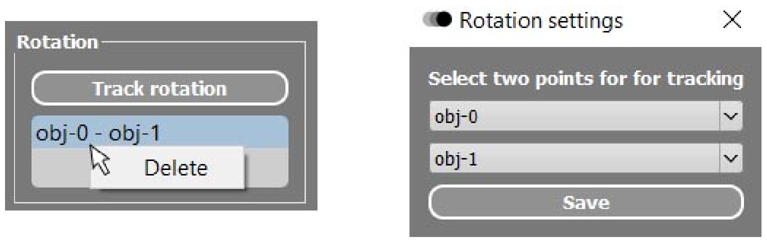


Figure 11: Track rotation

If more than two objects were selected and tracked, users could also track the rotation between two specified objects. To use this functionality the Track rotation menu needs to be used. Here the two endpoints for the rotation tracking can be selected, then the post-processing setting specified just like in the object tracking.

3.10 Plotting and exporting results

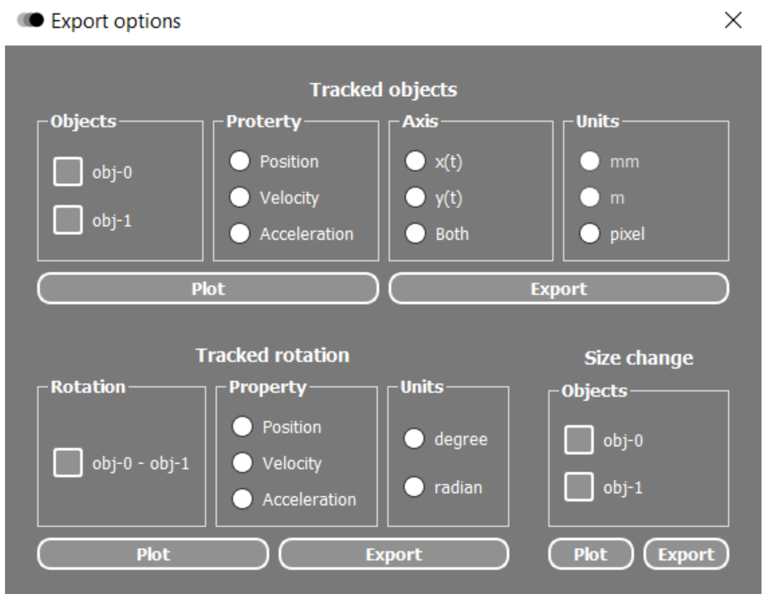


Figure 12: Plot and export options

After successful tracking and post-processing, the results can be visualized using the built in plotting options. The plotted figures can then be saved as images. If users need the calculated or the raw data for further analysis they can also export it into different supported file formats:

- CSV
- TXT
- XLSX

3.11 Playing back the results and exporting the video

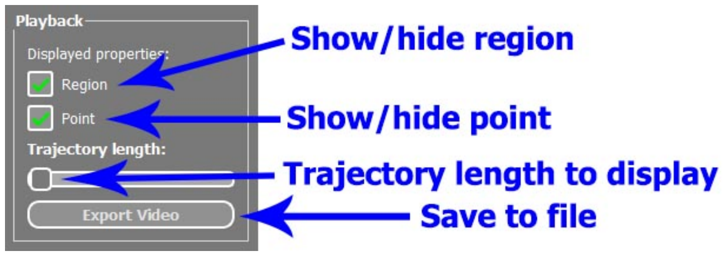


Figure 13: Playback settings

The results can be easily checked by playing back the video overlaying with the tracked objects. Here users can select which property of the objects they want to display. There is also a possibility for displaying the trajectory of the tracked points. The length of the plotted trajectory can also be modified by the trackbar. Note that for long videos it is not recommended to display the whole trajectory, only a small part of it as this can be computationally very expensive!
There is also a possibility to export the video with the overlaying objects into MP4 format.

3.12 Reset settings

If a new analysis is required on the same video but different section or objects the Reset ALL button is recommended. This method keeps the video open but deletes all previously specified settings and data such as Objects, Rotations, Ruler...
On the other hand, if users wish to open a new video recording they should use the stop button instead. It closes the opened recording and resets all the settings.

4 Source code

The source code of the software is available in the following GitHub repository:
https://github.com/flochkristof/motiontracker

5 License

Motion Tracker Beta is released under the GNU General Public License v3.0