

Three dimensional trajectory fitting to discrete noisy camera poses by Bézier interpolation: Code Guide

Johannes Hauck, Adam Kalisz and Jörn Thielecke

I. INTRODUCTION

This document describes the code usage published with the paper "Three dimensional trajectory fitting to discrete noisy camera poses by Bézier interpolation". The code is available at <https://github.com/GSORF/VisualOdometry2Bezier>.

II. RUN THE CODE

The code was developed with the python 3.9 interpreter. Necessary additional packages to run the code are:

- matplotlib version 3.3.2 or higher
- numpy version 1.19.4 or higher

The folder `\measurementData` contains example measurement files, which can be used in this demonstration. To run the example execute the python file `plot3D.py`.

III. PLOT3D.PY

This is the main file of the code demonstration, in which a measurement file is loaded, the algorithm for optimized control point estimation called and the results are plotted, as in Fig. 1 and Fig. 2. In the file the default folder path of the measurements is defined and the desired file can be selected during the run of the code. Alternatively or if an error occurs, the file path can be setted directly e.g. `measurementData/straightLine3D_3D_WNOS_P_10_NoiseS_5_001.txt`. In the array `plotBool` at the end of the file the plot can be configured. By setting the boolean values at the specific positions to true, the following objects are contained in the plot:

- 0: camera translations
- 1: control polygon
- 2: control points
- 3: Bézier points
- 4: ground truth
- 5: camera coordinate systems
- 6: optimized control points
- 7: optimized curve

IV. ALGORITHM FOR OPTIMIZED CONTROL POINT ESTIMATION

The algorithm for optimized control point estimation is called by the function `estOptControlPoints`. The input data is:

- non-noisy translation of the first measurement point
- non-noisy translation of the last measurement point
- all camera poses
- all absolute velocities

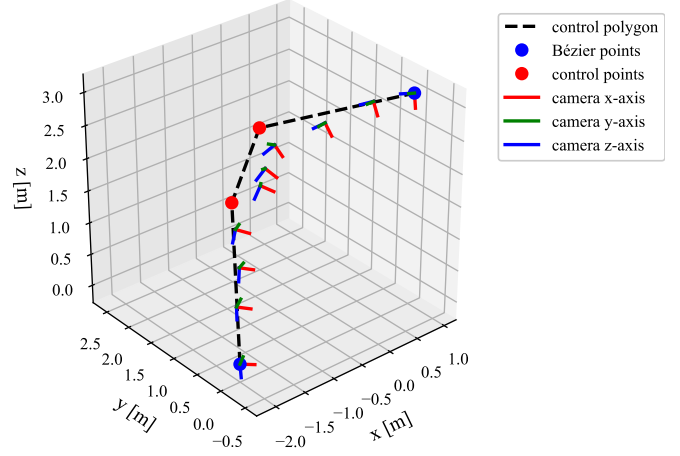


Fig. 1. Generated plot including control polygon, control points, Bézier points and camera coordinate systems.

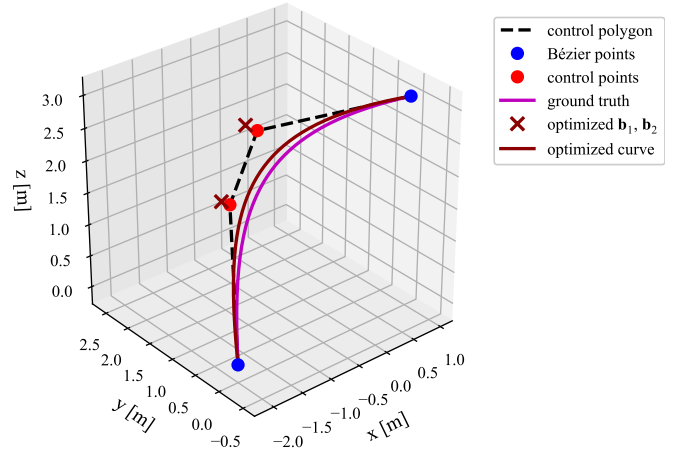


Fig. 2. Generated plot including control polygon, control points, Bézier points, ground truth, optimized control points and optimized curve.

- all measurement time stamps

The function returns next to the optimized control points the median absolute deviation in every component of both control points. It is recommended to use the algorithm in the debugger to follow its functionality step by step together with in line comments and the description of the paper.

V. MEASUREMENT DATA

The example measurement files in the folder `\measurementData`, like `parable3D_WNOTRS_3D_P_10_NoiseT_0.1_NoiseR_5_NoiseS_1_001`.

txt, are named according to the following conventions: The first substring shows the trajectories shape e.g. `parable3D`. The second part symbolises the kind of noise on the measurement data, like `WNOTRS` stands for white noise on translation, rotation and velocity. In the next substring the dimension of noise is expressed. In case of `3D` the translation noise is on every component of the translation and the rotation matrix is rotated around all three axis. In `2D` the translation noise is only added on the x any y component and the camera coordinate frame is only rotated around the up vector. `P_10` shows that in the measurement ten points are included. The numbers after the single noise strings `NoiseT`, `NoiseR` and `NoiseS` represent the standard deviation of noise on translation, rotation and velocity. The last part of the file name is a simple index.

The content of the measurement files is structured as follows. In the first N lines the measurement values for every point are listed. The first number is the current time, whereby every measurement starts at 0s. The second value is the parameter u . The absolute velocity is noted next. At last the camera pose is listed comma separated line by line. From file line $N + 1$ until the end the true Bézier and control points are stored. With this information the ground truth can be calculated and plotted.