Evaluation of stand-of-the-art monocular visual simultanious and mapping approaches

Masterarbeit

zur Erlangung des akademischen Grades

Master of Science in Engineering (M.Sc.)

Eingereicht bei:

Fachhochschule Kufstein Tirol Bildungs GmbH

Data Science & Intelligent Analytics

Verfasser/in:

Julian Bialas, BSc

1910837917

Erstgutachter	:	Prof. (FH) PD Dr. Mario Döller
Zweitgutachter	:	Robert Kathrein, MSc

Abgabedatum:

06. July 2020

Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich die vorliegende Masterarbeit selbständig und ohne fremde Hilfe verfasst und in der Bearbeitung und Abfassung keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe. Die vorliegende Masterarbeit wurde noch nicht anderweitig für Prüfungszwecke vorgelegt.

Kufstein, 06. July 2020

Julian Bialas, BSc

Sperrvermerk

Ich habe die Sperrung meiner Masterarbeit beantragt, welche von der Studiengangsleitung genehmigt wurde.

Kufstein, 06. July 2020

Julian Bialas, BSc

Contents

1 Introduction **1**

1.1 Related Work 1

2 Methods **2**

2.1 vSLAM Algorithms 2

2.1.1 Definitions 2

2.1.2 ORB-SLAM 3

2.1.3 DSM-SLAM 5

2.1.4 DSO-SLAM 5

2.2 Calculations 5

2.2.1 Trajectory Alignment 5

2.3 Datasets 6

2.3.1 EuRoC Dataset 6

2.4 Setup and Environment 7

2.4.1 Evaluation 7

2.4.2 Flight Path Planning 8

3 Results **9**

3.1 Trajectory Evaluation 9

3.2 Pointcloud Evaluation 9

3.3 Calculation Time 9

4 Discussion **14**

4.1 Conclusion of SLAM-Algorithm Evaluation 14

4.2 Handlungsempfehlung 14

4.2.1 Framework for Trajectory Automatation 14

4.2.2 Trajectory Automatation using Reinforement Learning . 14

List of Figures

1 Overview of the system components extracted from [3] 3

2 Pointcloud ground truth of sequence V1_01_easy visualized with python package pptk 7

3 Ground truth flight path and evaluated flight path of each algorithm after alignment with the method of Umeyama in the x and y axis. Left the sequence MH01, middle the sequence V102 and right the sequence V203 is displyed. 10

4 Boxplot of all euclideans distances between the ground truth position of the keyframe and the evaluated position after alignment with the method of Umeyama. Outliers greater than 1.5 are not displayed. 12

5 The groundtruth of the Pointcloud from Sequence V101 (white points) and the evaluated points by each algorithm (red points). The points in Figure (a) are four times as large for better visability (ORB-SLAM generates only few points). 12

6 Boxplot of the eucleadian distances between an evaluated point and the closest point of the ground truth point cloud. For computational feasibility, for each sequence and algorithm, 500 points for evaluation are sampled randomly 13

7 Influence of downsizing of the images on the trajectory error (a) and the computation time (b) for the sequence V101. 15

List of Tables

1 Overview of the sequences included in the EuRoC Dataset . . . 8

2 Number and accuracy of evaluated points of each algorithm . . 11

3 Computation Time (excluded time needed for initialization) of each Sequence and Algorithm 11

List of Listings

List of Acronyms

HTML HyperText Markup Language

JS JavaScript

FH Kufstein Tirol
Data Science & Intelligent Analytics

Abstract of the thesis: **Evaluation of stand-of-the-art monocular visual simultanious and mapping approaches**

Author: Julian Bialas, BSc
First reviewer: Prof. (FH) PD Dr. Mario Döller
Second reviewer: Robert Kathrein, MSc

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis.

Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

06. July 2020

FH Kufstein Tirol

Data Science & Intelligent Analytics

Kurzfassung der Masterarbeit: **Evaluation of stand-of-the-art monocular visual simultanious and mapping approaches**

Verfasser: Julian Bialas, BSc

Erstgutachter: Prof. (FH) PD Dr. Mario Döller

Zweitgutachter: Robert Kathrein, MSc

06. July 2020

1. Introduction

1.1 Related Work

2. Methods

2.1 vSLAM Algorithms

2.1.1 Definitions

Keyframe

Direct and Feature based methods

Group of Rigid Transformations in 3D

SE(3) is the group of rigid transformations in 3D space [2]. Each Matrix $T \in \mathbb{R}^{4x4}$ with

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

and $R \in \mathbb{R}^{3x3}$ beiing a rotation matrix and $t \in \mathbb{R}^3$ a translational vector, is an element of SE(3).

2.1.2 ORB-SLAM

ORB SLAM is a feature based, state of the art slam method. The first version was published in 2015 [3]. Here, an overview of the functionality of ORB SLAM is provided. The Algorithms runs on three threats simultanously. Each thread performes one of the following tasks: Tracking, Local Mapping and Loop Closing. An overview over the tasks can be found in figure 1. The explaination of these system components are described in the following subsections.

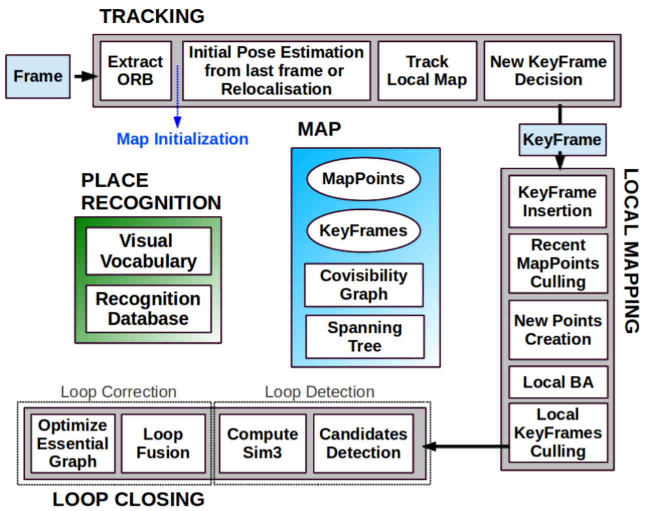


Figure 1: Overview of the system components extracted from [3]

Tracking

The tracking component determines the localization of the camera and decides, when a new keyframe is beeing inserted. As it is shown in figure 1, the tracking is performed in four steps.

1. Feature Extracting

   Features are extracted using Oriented FAST and Rotated BRIEF [4]. This method starts by searching for FAST (Features from Accelerated and Segments Test). Herefor, for each pixel x in the image, a circle of 16 pixels around that pixels is considered and checked if at least eight of these 16 pixels have major brightness differences. If so, the pixel x is considered as a keypoint, since it is likely to be an edge or corner. This is repeated again and again after downsizing the image up to a scale of eight. To extract features evenly distributed over the image, it is devided into a grid, trying to extract five features per cell. Extracting features this way, makes the algorithm more stable to scale invariance. Next the orientation of the extracted feature is calculated using a intensity centroid. Finally the features are converted into a binary vectors (ORB descriptor) using a modified version, which is more robust to rotation, of BRIEF descriptors (Binary robust independent elementary feature).

2. Initial Pose Estimation
3. Track Local Map
4. New Keyframe Decision

Local Mapping

Loop Closing

Bundle Adjustment

The keyframe poses $T_i \in \mathrm{SE}(3)$ and Map Points $X_j \in \mathbb{R}^3$ are optimized by minizing the reprojection error to the matched keypoints $x_{i,j} \in \mathbb{R}^2$. The error is computed by the following term:

$$e_{i,j} = x_{i,j} - \pi_i \left(T_i, X_j \right)$$

. i is the respective Keyframe and j the index of the map Point. π_i is a projection function, calculation a transformation to project all keypoints on mappoints by minimizing a cost function, that can be found in [5]. In case of full BA (used in the map initialization) we optimize all points and keyframes, by the exception of the first keyframe which remain fixed as the origin. In local BA all points included in the local area are optimized, while a subset of keyframes is fixed. In pose optimization, or motion-only BA, all points are fixed and only the camera pose is optimized.

2.1.3 DSM-SLAM

2.1.4 DSO-SLAM

2.2 Calculations

2.2.1 Trajectory Alignment

In order to compare the evaluated position of the camera at a given time with the ground truth of the position, the trajectories need to be aligned. This is because most SLAM Algorithms innitilize the origin of their coordinate system with the camera position from the first frame. Whereas the ground truth of the trajectory uses a different origin. As a consequence, evaluated points $\{\widehat{p_i}\}_{i=0}^{N-1}$ can not be compared to the ground truth points $\{p_i\}_{i=0}^{N-1}$ Also, as described in the vSLAM Algorithms section,

the minority of the existing vSLAM algorithms are recognizing the true scale of the coordinate system. For those two reasons, the target is to find $S = \{R, t, s\}$, while R being a rotation matrix, t a translation vector and s a scaling factor, such that

$$S = \underset{S'=\{R',t',s'\}}{\arg\min} \sum_{i=0}^{N-1} \left\| p_i - s'R'\widehat{p_i} - t' \right\|^2$$

.

In other words, the evaluated points are rotated, translated and scaled in a way, that the sum squared error over the point distances is minimized. The upper expresssion is calculated by using the method of Umeyama [6].

Similar to principal component analysis, Umeyama uses the singular value decomposition of the covarianve matrix Σ of p and $\widehat{p}$. Thus, $\Sigma = UDV^T$ is yielded. Umeyama proves, that R, t and s can be calculated as followed:

$$\begin{aligned} R &= UWV^T \\ s &= \frac{1}{\sigma_p^2} \operatorname{tr}(DW) \\ t &= \mu_{\widehat{p}} - sR\mu_p \end{aligned}$$

with

$$W = \begin{cases} I, & \text{if } \det(U)\det(V) = 0 \\ \operatorname{diag}(1, 1, -1), & \text{otherwise} \end{cases}$$

σ_p beeing the standard deviation of p, μ the mean and tr the trace of a matrix.

2.3 Datasets

2.3.1 EuRoC Dataset

For the evaluation of the vSLAM Algorithms, the EuRoC dataset [1] was used. The dataset contains eleven video sequences, recorded with a micro aerial vehicle at 20 frames per second. The sequences have a resultion of 752x480 pixels. For each Sequence, RGB images from two cameras exist. However, since the evaluation focuses on monocular SLAM methods, only the left camera was considered. Also the available inertial and camera pose data was not taken in consideration. The first five sequences were recorded in the machine hall at ETH Zürich, and the other six were recorded in a room, that was provided with additional obsticals. For the latter six sequences, the groundtruth of the environment exists as a dense pointcloud, as can be seen in figure 2.

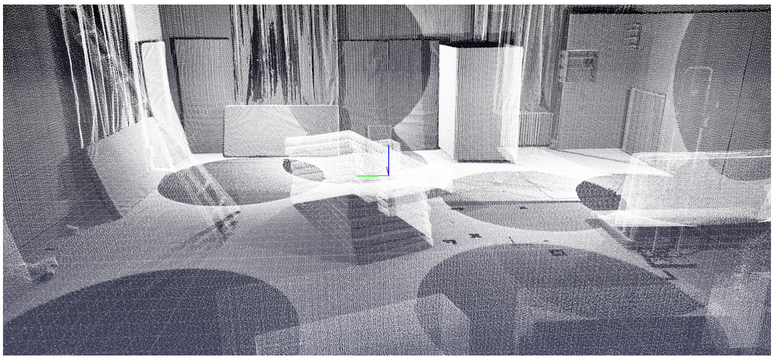


Figure 2: Pointcloud ground truth of sequence V1_01_easy visualized with python package pptk

Finally the true position of the camera is known at a high frequency of over 200 points per second. An overview of the sequences is shown in table 1.

2.4 Setup and Environment

2.4.1 Evaluation

The entire evaluation is run on a virtual machine. The host system is a lenovo yoga with eight GB of RAM and the basic model (8250U CPU @1.6 GHz 1.80GHz) of an eight core i5. The operating system of the host machine is Windows 10 Home. The virtual machine is given 5 GB of Ram and 4 cores.

Table 1: Overview of the sequences included in the EuRoC Dataset

Sequence Name	Duration in s	Average Velocity in ms^{-1}	Pointcloud available
MH_01_easy	182	0.44	No
MH_02_easy	150	0.49	No
MH_03_medium	132	0.99	No
MH_04_difficult	99	0.93	No
MH_05_difficult	111	0.88	No
V1_01_easy	144	0.41	Yes
V1_02_medium	83.5	0.91	Yes
V1_03_difficult	105	0.75	Yes
V2_01_easy	112	0.33	Yes
V2_02_medium	115	0.72	Yes
V2_03_difficult	115	0.75	Yes

The operating system of the virtual machine is Ubuntu 18.04. All further setup information can be extracted from the github repository.

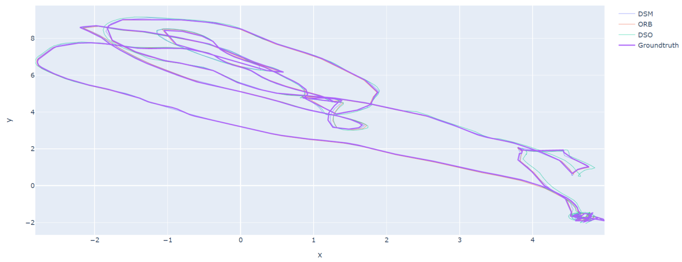
2.4.2 Flight Path Planning

3. Results

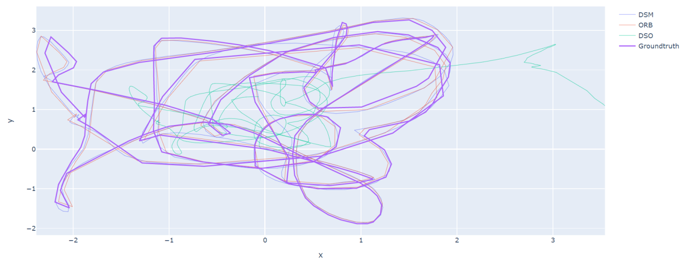
3.1 Trajectory Evaluation

3.2 Pointcloud Evaluation

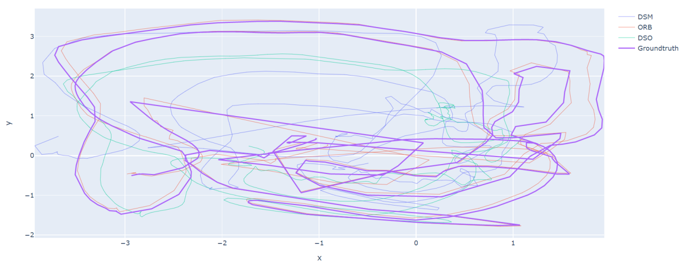
3.3 Calculation Time



(a) MH01



(b) V102



(c) V203

Figure 3: Ground truth flight path and evaluated flight path of each algorithm after alignment with the method of Umeyama in the x and y axis. Left the sequence MH01, middle the sequence V102 and right the sequence V203 is displyed.

Table 2: Number and accuracy of evaluated points of each algorithm

Sequence Name	ORB	DSM	DSO
MH_01_easy	8958 (/)	675720 (/)	361633 (/)
MH_02_easy	8692 (/)	700920 (/)	343804 (/)
MH_03_medium	/ (/)	614264 (/)	371752 (/)
MH_04_difficult	7943 (/)	495752 (/)	208445 (/)
MH_05_difficult	8373 (/)	517712 (/)	232415 (/)
V1_01_easy	7075 (0.049)	6108440 (0.066)	374257 (0.066)
V1_02_medium	6517 (0.042)	648440 (0.187)	366513 (1.458)
V1_03_difficult	/ (/)	775080 (0.092)	448212 (0.459)
V2_01_easy	/ (/)	584552 (0.58)	247905 (0.086)
V2_02_medium	/ (/)	733992 (0.078)	490608 (0.104)
V2_03_difficult	/ (/)	921312 (0.645)	465396 (0.677)

Table 3: Computation Time (excluded time needed for initialization) of each Sequence and Algorithm

Sequence Name	Computation Time in s ORB	Computation Time in s DSM	Computation Time in s DSO
MH_01_easy	257	1098	749
MH_02_easy	209	984	690
MH_03_medium	198	1369	707
MH_04_difficult	165	896	504
MH_05_difficult	193	825	633
V1_01_easy	253	1383	905
V1_02_medium	150	1550	820
V1_03_difficult	186	2262	1134
V2_01_easy	187	1045	612
V2_02_medium	162	1675	1522
V2_03_difficult	143	1600	793

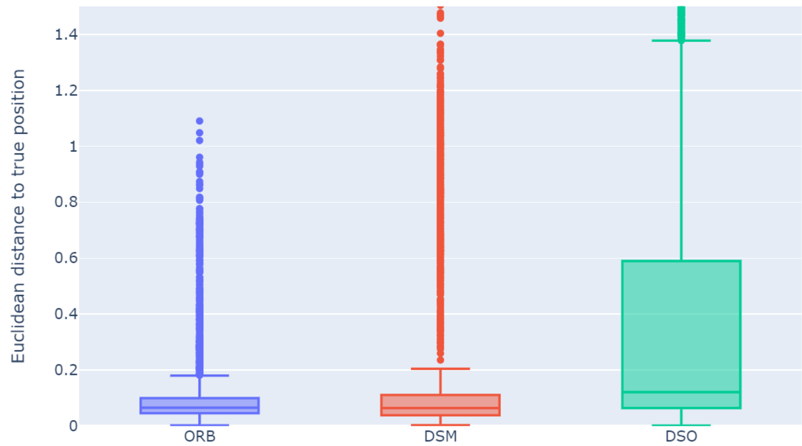
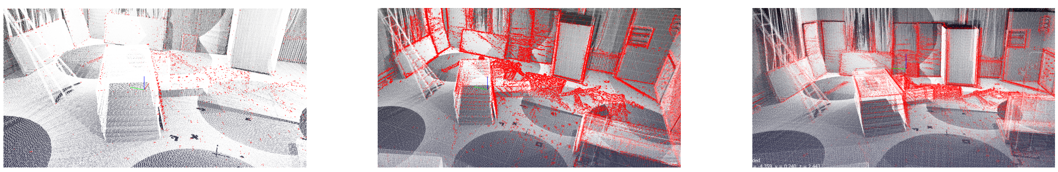


Figure 4: Boxplot of all euclideans distances between the ground truth position of the keyframe and the evaluated position after alignment with the method of Umeyama. Outliers greater than 1.5 are not displayed.



(a) ORB (b) DSM (c) DSO

Figure 5: The groundtruth of the Pointcloud from Sequence V101 (white points) and the evaluated points by each algorithm (red points). The points in Figure (a) are four times as large for better visability (ORB-SLAM generates only few points).

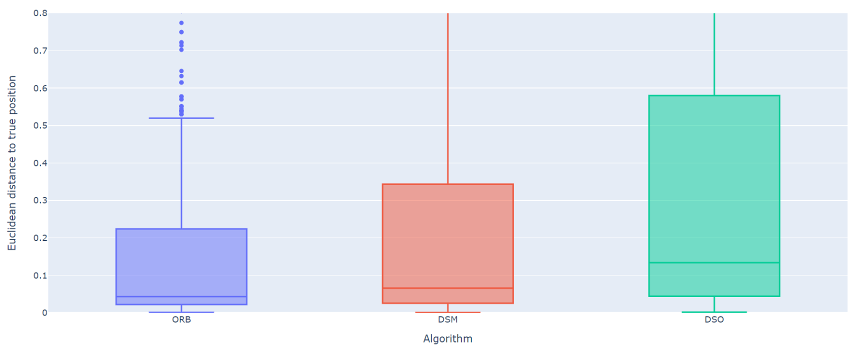


Figure 6: Boxplot of the eucleadian distances between an evaluated point and the closest point of the ground truth point cloud. For computational feasibility, for each sequence and algorithm, 500 points for evaluation are sampled randomly

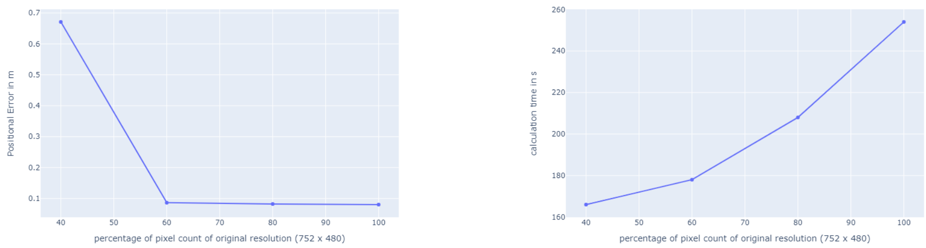
4. Discussion

4.1 Conclusion of SLAM-Algorithm Evaluation

4.2 Handlungsempfehlung

4.2.1 Framework for Trajectory Automatation

4.2.2 Trajectory Automatation using Reinforement Learning



(a) trajectory error

(b) computation time

Figure 7: Influence of downsizing of the images on the trajectory error (a) and the computation time (b) for the sequence V101.

Bibliography

[1] M. Burri. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016.

[2] E. Eade. Lie groups for computer vision. 2002.

[3] R. Mur-Artal. Orb-slam: a versatile and accurate monocular slam system. *IEEE TRANSACTIONS ON ROBOTICS*, 2015.

[4] E. Rublee. Orb: an efficient alternative to sift or surf. 2012.

[5] B. Triggs. Bundle adjustment – a modern synthesis. *International Workshop on Vision Algorithms*, 2000.

[6] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1991.