Masterarbeit

Technische Hochschule Deggendorf

Fakultt Angewandte Naturwissenschaften und Wirtschaftsingenieurwesen

Studiengang Mechatronische und cyber-physische Systeme

Implementierung und Evaluierung kamerabasierter Odometriemethoden auf einer mobilen Plattform

Implementation and Evaluation of Camera-based Odometry Methods on a Mobile Platform

Masterarbeit zur Erlangung des akademischen Grades:

Master of Engineering (M.Eng.)

vogelegt von: **Gautam Dobariya**

Matrikelnummer: **777723**

prfer: **Prof. Dr. Ing. Wolfgang Aumer**

Alexander

Cham 26th September 2020

Declaration of Integrity

I hereby confirm, that I have written the Masters thesis at hand independently,that I have not used any sources or materials other than those stated, nor availed myself of any unauthorized resources, and that I have not submitted this Masters thesis in any form as an examination paper before, neither in this country, nor abroad, and hat the electronic copy of this Masters thesis and the printed versions are identical.

Cham, ____________ ____________________

Date Signature

Abstract

In industrial automation, the market needs reliable and scalable solutions for autonomous transportation in production and logistic processes. To address this need, SICK AG e.g. offers reliable LiDaR localization solutions and a big portfolio of LiDaR sensors. However, the increased usage of small mobile platforms in swarm applications introduces additional requirements compared to historically bigger autonomous vehicles: The cost factor regarding number and type of used sensor systems increases, the performance of the hardware is limited and the environment changes.The main target of this work is the investigation, adaption and evaluation of visual Odometry algorithms for application in mobile swarm robotics.

Keywords:

Standard Standard Standard Standard Standard Standard Standard Standard Standard Standard Standard Standard Standard Standard Standard

Acknowledgment

I would like to thanks

Contents

Acronyms

SFM Structure From Motion. 2

VO Visual Odometry. 2

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In industrial automation, the market needs reliable and scalable solutions for autonomous transportation in production and logistic processes. To address this need, SICK AG e.g. offers reliable LiDaR localization solutions and a big portfolio of LiDaR sensors. However, the increased usage of small mobile platforms in swarm applications introduces additional requirements compared to historically bigger autonomous vehicles: The cost factor regarding number and type of used sensor systems increases, the performance of the hardware is limited and the environment changes.The main target of this work is the investigation, adaption and evaluation of visual Odometry algorithms for application in mobile swarm robotics.

Localization of a robot is a fundamental challenge and one of the most important tasks. For autonomous navigation, motion tracking, and obstacle detection and avoidance, a robot must know of its position in real time. Vision-based Odometry is a novel and robust solution utilized for this purpose.[6] It allows a robot to localize itself accurately by using only a stream of images captured by a camera attached to the vehicle.

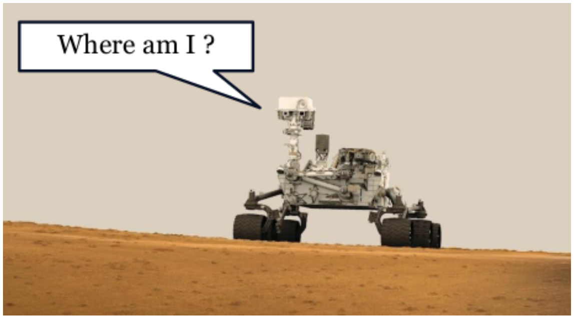

Figure 1.1: NASA Path finder robot (source:www.jpl.nasa.gov)]

1.2 Thesis Structure

The thesis is structured into seven chapters. The Chapter 1 which serves as an introduction.

Chapter 2 covers the basics of Visual Odometry. It explains the different methods of VO and a general process of VO. Furthermore it describes current state-of-the-Art VO algorithms and selection criteria for chosen ones for further work.

Chapter 3 explains in depth the 3 chosen algorithms LDSO, SVO, ORB-SLAM2. Code flowcharts are explained.

Chapter 4 provides information regarding the Mobile robot and cameras used for the implementation and required parameters like Intrinsic and Photometric Camera Calibration.

Chapter 5 describes the procedure of experiments and data collection. All algorithms are then evaluated. The results are then discussed.

Chapter 6 gives suggestions how to choose the best algorithm and to improve results furthermore.

Chapter 7, concludes the thesis by summarizing the results and suggestions for future works.

Chapter 2

Basics

This Chapter gives an overview of Visual Odometry, its working and different types of approaches and state-of-the-art of Visual Odometry.

2.1 What is Visual Odometry ?

VO is defined as the process of estimating the egomotion (translation and rotation with respect to a reference frame) of an Agent(e.g. vehicle, human and robot) by observing a sequence of images using single or multiple cameras attached to it.[4]] VO is a particular case of a technique known as Structure From Motion (SFM) in Computer Vision that tackles the problem of 3D reconstruction of environment and camera poses from set of images[7]. VO mainly focuses on 3-D motion of the camera sequentially in real time (sequential SFM).VO mainly differs with SLAM in terms of global mapping. VO focuses on local consistency and incrementally estimate the path of camera/robot pose, and some local optimization whereas SLAM performs both localization and global mapping.

2.2 Problem Formulation

A camera is attached rigidly to a moving Robot in an unknown environment and provides images at some constant time instants k. For a monocular case images taken are given by $I_{0:n} = \{I_0, ..., I_n\}$. For stereo case there would be two sets of images for left and right camera. The camera positions between two consecutive time instants $k-1$ and k can be described as rigid body transformation $T_{k,k-1} \in R^{4\times 4}$ in the following form:

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

where $R_{k,k-1}/inSO(3)$ is the rotation matrix, and $t_{k,k-1}$ the translation. The set $T_{1:n} = \{T_{1,0}, ..., T_{n,n-1}\}$ contains all relative transformations. As shown in Figure 2.1 given the initial pose, the camera pose trajectory at given time can be computed by concatenating all transformations up-to that time with initial pose[7].

VO computes the path trajectory pose by pose. To improve the accuracy of trajectory an iterative refinement called as Bundle Adjustment can be applied. It minimizes the sum of squared error of reprojection of over last n image frames.

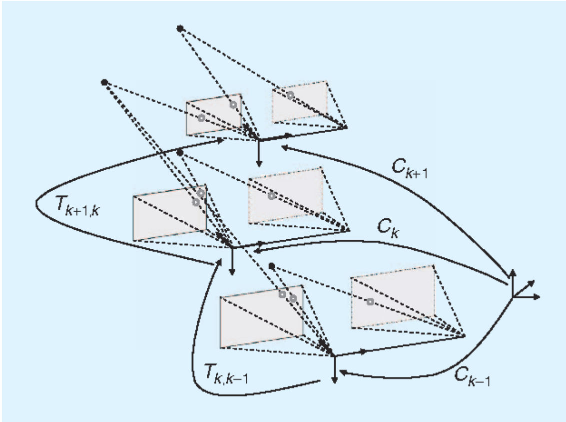

Figure 2.1: An illustration of the visual odometry problem.[7]

2.3 VO Pipeline

The VO pipeline is summarized in Figure 2.2. For every new image I(or image pair for stereo case), the first two steps consist of detecting and matching 2-D features with those from the previous frames. 2-D features that are the reprojection of the same 3-D feature across different frames are called image correspondences. The feature detection consists of detecting features independently in all the images and then then feature matching will find the same features in sequence of images and then tracks them using a local search technique, such as correlation. The next step consists of computing the relative motion(translation and rotation) between the two consecutive time instants. There are three different approaches for motion estimation depending on the correspondences specified in 3-D or 2-D. Current camera pose is then computed by concatenation of the previous pose. Finally, an iterative local optimization known as BA can be done over the last m frames to obtain a more accurate estimate of the local trajectory. Each steps are discussed further in next section.

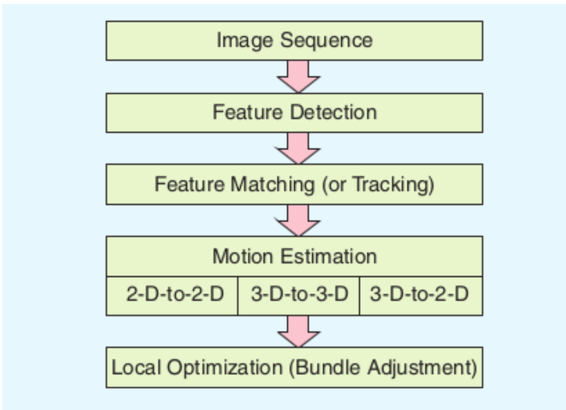

Figure 2.2: Flowchart of VO
[7]

2.4 VO Approaches

VO mainly classified based on types of approaches named as Indirect(Feature-based), Direct (Appearance based) and Hybrid Approach. Another classification can be done on camera used for Such as stereo, monocular, omnidirectional, and RGB-D cameras. Monocular VO suffers from scale ambiguity because of unknown depth information of images. Stereo VO solves this scaling problem by retrieving depth information using two cameras at little on distance known as baseline. Stereo case can be degraded to monocular if the baseline is much smaller than distances to the scene from camera. The Analysis of these approaches is discussed further.

2.4.1 Indirect Approach

This is a classical approach for VO and SFM. The Indirect or Feature-based method involves extraction of some features such as corners, edges etc. from the images frames. See Figure2.3 These features are then matched and tracked among two consecutive image frames. Based on the feature tracking motion of camera is estimated. This approach can typically divided into two steps: 1) Feature detection and matching, 2) geometric optimization on the computed point correspondences. In first step an image is matched with a previous one by comparing each feature in both images and calculating the Euclidean distance of feature vectors to find the candidate matching features.[6] In second step using these match correspondences the camera motion and surrounding 3D geometry can be estimated. In case of stereo VO the features are first compared with each image pair and thus depth information of feature can be estimated. In this approach the reprojection error is minimized using Bundle Adjustment because keypoints positions(geometric quantities) are used to compute camera pose. The Bundle adjustment problem is described as below.

$$T_{k,k-1} = argmin_T \sum_i \|u_i^{'} - u_i\|_\Sigma^2$$

where $u_i^{'} = \pi(P_i, T_{k,k-1})$, u_i is i^{th} pixel 2D positions and $u_i^{'}$ is reprojected 2D pixel position using 3D projection (π).

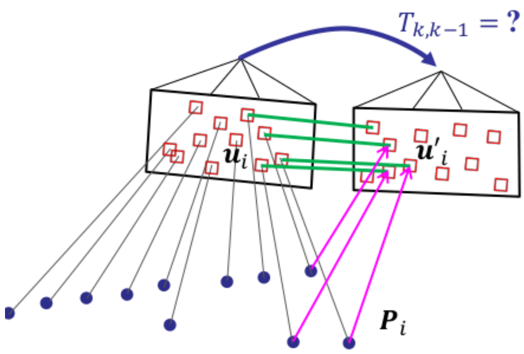

Figure 2.3: Indirect method Optimizes Reprojection Error

2.4.2 Direct Approach

Direct method uses directly the pixel intensity as an information instead of extracting features and tracking them for motion estimation. Direct methods are based on assumption that Brightness remains constant in all image frames.[3] Direct methods are also known as Appearance based approach as it monitors the appearance of image in consecutive frames. The camera motion then can be estimated by Optical-flow algorithms which determines the displacement of brightness patterns of a group of pixels using intensity values from one image to another.[6] There are two types of such algorithms based on selection of number of image pixels for calculation called as Dense and Sparse Optical-flow methods. Dense algorithms are less robust to noise as compared to Sparse based. Sparse algorithms select only those features which have more variance than others in particular image region. One of the most used sparse based algorithms for tracking is Lucas-Kanade method. [5]. As There is no feature extraction step is involved direct approach minimizes directly the photometric error formulated as below.

$$T_{k,k-1} = argmin_T \sum_i \|I_k(u_i^{'}) - I_{k-1}(u_i)\|_\sigma^2$$

where $u_i^{'} = \pi(P_i, T_{k,k-1})$ and I_k is k_{th} image. see Figure 2.4

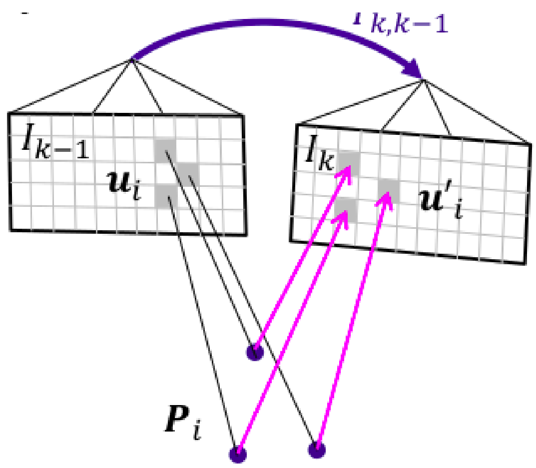

Figure 2.4: Direct Method Optimizes Photometric Error

Depending upon the number of feature selection for calculating 3D geometry Direct methods can be divided into three types such as Dense, Semi-dense and Sparse methods.A graphical Overview of these methods can be seen in Figure 2.5. Dense approaches use every pixel in the image, where as semi-dense use just the pixels with high intensity gradient, and the proposed and sparse approach uses selected pixels at corners or along intensity gradient edges.[1]

As Direct methods minimize the photometric error (intensity difference) for tracking between two images they required a well calibrated camera as compared to Indirect methods because they minimized the image pixel positions on images. A simple process comparison and properties are describes in th Fig.2.6 and

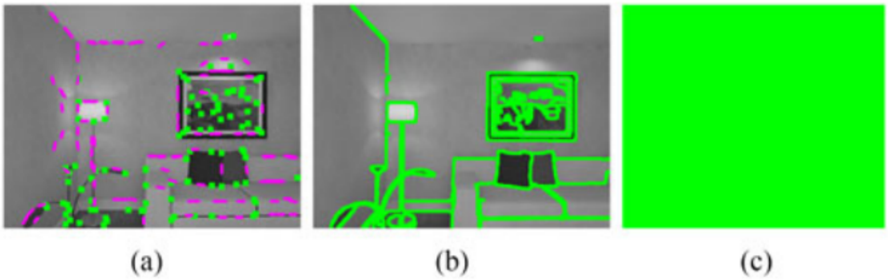

Figure 2.5: Image-to-model alignment (marked in green for corners and magenta for edgelets) for sparse, semi-dense, and dense methods. [2]

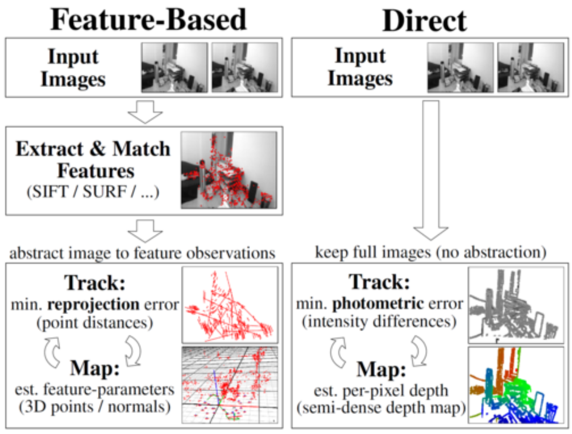

Figure 2.6: Process comparison between Direct and Indirect methods source:[1]

Indirect	**Direct**
use only corners, edges for reconstruction	use whole image for reconstruction
flexible :uses outliers removal (e.g RANSAC)	Inflexible :outliers removal is difficult
robust to inconsistencies like rolling stutter	not robust such model
not requires good initialization	requires good initialization
only geometric calibration	requires geometric as well as photometric calibr
e.g.PTAM,RTAB,S-PTAM, ORB-SLAM	e.g. DTAM, LSD-SLAM, DSO

Chapter 3

Experimental Setup

3.1 Introduction

Some meaningless text to show how citation works [].

3.1.1 Random text

.

Random squared text

Chapter 4

Conclusion

In a non-fiction book, a Conclusion is an ending section which states the concluding ideas and concepts of the preceding writing. This generally follows the body or perhaps a Afterword, and the conclusion may be followed by an Epilogue, Outro, Postscript, Appendix/Addendum, Glossary, Bibliography, Index, Errata, or a Colophon.

References

[1] J. Engel, T. Schöps, and D. Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *European Conference on Computer Vision (ECCV)*. 2014.

[2] C. Forster et al. "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems". In: *IEEE Transactions on Robotics* 33.2 (2017), pp. 249–265.

[3] M. Irani and P. Anandan. "All About Direct Methods". In: *ICCV '99: Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*. 1999, pp. 267–277.

[4] Reza Hoseinnezhad Khalid Yousif Alireza Bab-Hadiashar. *An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics*. 2015. DOI: `10.1007/s40903-015-0032-7`.

[5] Bruce D. Lucas and Takeo Kanade. "An iterative image registration technique with an application to stereo vision". In: *In IJCAI81*. 1981, pp. 674–679.

[6] M.Iqbal Saripan Mohammad O.A.Aqel Mohammad H.Marhaban and Napsiah Bt.Ismail. *Review of visual odometry: types, approaches, challenges, and applications*. 2016. DOI: `10.1186/s40064-016-3573-7`.

[7] D. Scaramuzza and F. Fraundorfer. "Visual Odometry [Tutorial]". In: *IEEE Robotics Automation Magazine* 18.4 (2011), pp. 80–92.

Appendix A

Title of appendix A

A.1 Section title

Appendix B

List of installed software at FH Wiener Neustadt

Software Winter Semester 2013 / 2014

Wo: E=EDV (3,4,5,6,8)
L=Labor, C=CAD (1,2,7,9)

Als CAD Säle sind die Säle EDV1, EDV7 und EDV9 vorgesehen.

	Version	Sprache	Wo
MS Office 2010	2010	deutsch & englisch	E,C
MS Project	2010	deutsch & englisch	E,C
MS Visio	2010	deutsch & englisch	E,C
SPSS	21		E,C
MaxQDAPlus	11		E,C,L
Amos	21		E,C
R-Project			E,C
IBM SPSS Modeler	15		E,C
Matlab R2011a			E,L,C
GPSS	5.2.2		E,L,C
Labview		englisch	C
Ansys	14.5		C
Comsol			
Oslo			
Eagle	6.4.0		
Codelite	5.1.0		
MPLab	8.86		
Bloodshed	4.9.9.2		
Esacomp	4.4		C
Catia	V5 R22		E,C
Geomagic	12		E,C
Freshminder			
Magix			
Minitab	16		
EndNote	X5		E,C
Audacity	2.0.0(ANSI)		E,C
Oracle Developer Suite			
Oracle VirtualBox			E,L,C
Irfanview			E,L,C
Firefox			E,L,C

VideoLan (VLC)			E,C
Wiener Testsystem			E,C
SAP Gui			E,C
Visual Studio 2012			C
Kinovea	0815	englisch	
Longomatch	0.18.12	deutsch	
Xilinx	14.2	englisch	C
SkillSpector & SkillCapture		englisch	E
Octave		englisch	E
ÖWA-Plus		deutsch	E,C