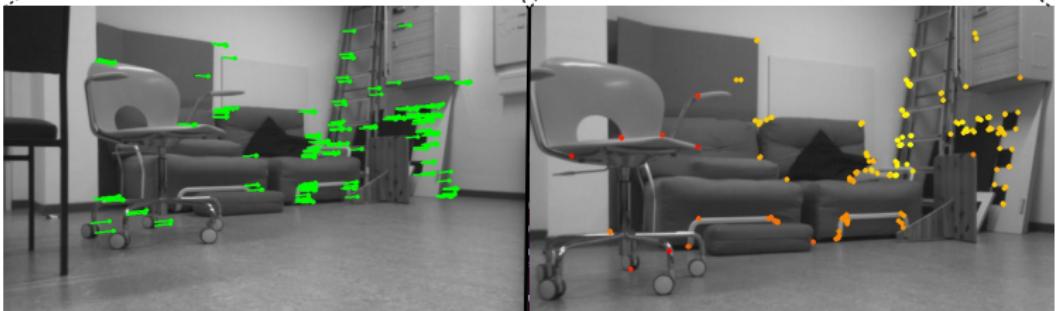


Localisation Using Adaptive Feature Selection Strategy



Overview

- ▶ Motivation

Overview

- ▶ Motivation
- ▶ Proposed Approach

Overview

- ▶ Motivation
- ▶ Proposed Approach
- ▶ Visual Odometry Algorithm Overview

Overview

- ▶ Motivation
- ▶ Proposed Approach
- ▶ Visual Odometry Algorithm Overview
- ▶ Adaptive Feature Search

Overview

- ▶ Motivation
- ▶ Proposed Approach
- ▶ Visual Odometry Algorithm Overview
- ▶ Adaptive Feature Search
- ▶ Implementation and Experimental Results

Overview

- ▶ Motivation
- ▶ Proposed Approach
- ▶ Visual Odometry Algorithm Overview
- ▶ Adaptive Feature Search
- ▶ Implementation and Experimental Results
- ▶ Conclusion and Outlook

Motivation

- ▶ Autonomous robots getting more relevant in science and our daily lives



¹<http://www.bayer.com.tr/>

Motivation

- ▶ Autonomous robots getting more relevant in science and our daily lives
- ▶ Location of mobile robots is required for most applications



¹<http://www.bayer.com.tr/>

Motivation

- ▶ Autonomous robots getting more relevant in science and our daily lives
- ▶ Location of mobile robots is required for most applications
- ▶ Common sensors (e.g. GPS) are not always feasible



¹<http://www.bayer.com.tr/>

Motivation

- ▶ Autonomous robots getting more relevant in science and our daily lives
- ▶ Location of mobile robots is required for most applications
- ▶ Common sensors (e.g. GPS) are not always feasible
- ▶ Humans are mostly orientating with vision



¹<http://www.bayer.com.tr/>

Motivation

- ▶ Autonomous robots getting more relevant in science and our daily lives
- ▶ Location of mobile robots is required for most applications
- ▶ Common sensors (e.g. GPS) are not always feasible
- ▶ Humans are mostly orientating with vision
- ▶ Modern cameras provide good quality for a reasonable price



¹<http://www.bayer.com.tr/>

Motivation

- ▶ Autonomous robots getting more relevant in science and our daily lives
 - ▶ Location of mobile robots is required for most applications
 - ▶ Common sensors (e.g. GPS) are not always feasible
 - ▶ Humans are mostly orientating with vision
 - ▶ Modern cameras provide good quality for a reasonable price
- ⇒ Using cameras for localisation task



¹<http://www.bayer.com.tr/>

Proposed Approach



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Stereo Visual Odometry: calculating the egomotion of a camera pair

Proposed Approach

- ▶ Stereo Visual Odometry: calculating the egomotion of a camera pair
- ▶ Goal: Find the rigid body motion ($R|t$) of the robot

Proposed Approach

- ▶ Stereo Visual Odometry: calculating the egomotion of a camera pair
- ▶ Goal: Find the rigid body motion ($R|t$) of the robot
- ▶ Problem: Operating in unknown environments with regions of few features

Proposed Approach

- ▶ Stereo Visual Odometry: calculating the egomotion of a camera pair
 - ▶ Goal: Find the rigid body motion ($R|t$) of the robot
 - ▶ Problem: Operating in unknown environments with regions of few features
- ⇒ Applying an adaptive feature selection strategy

Proposed Approach

- ▶ Stereo Visual Odometry: calculating the egomotion of a camera pair
 - ▶ Goal: Find the rigid body motion ($R|t$) of the robot
 - ▶ Problem: Operating in unknown environments with regions of few features
- ⇒ Applying an adaptive feature selection strategy
- ▶ Implementation on a real robot system: Robotino, a mobile robot with omnidirectional wheels

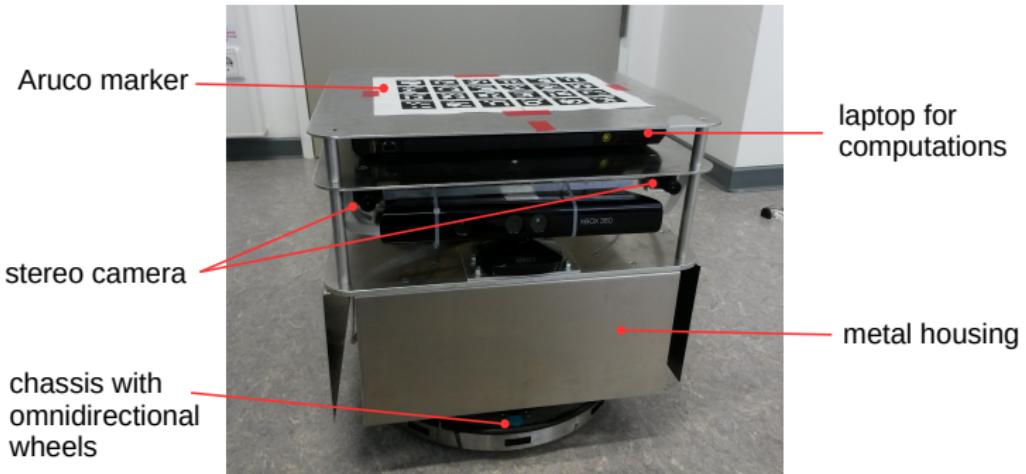
Proposed Approach

- ▶ Stereo Visual Odometry: calculating the egomotion of a camera pair
- ▶ Goal: Find the rigid body motion ($R|t$) of the robot
- ▶ Problem: Operating in unknown environments with regions of few features
- ⇒ Applying an adaptive feature selection strategy
- ▶ Implementation on a real robot system: Robotino, a mobile robot with omnidirectional wheels
- ▶ VO algorithm programmed in C++ with OpenCV

Proposed Approach

- ▶ Stereo Visual Odometry: calculating the egomotion of a camera pair
- ▶ Goal: Find the rigid body motion ($R|t$) of the robot
- ▶ Problem: Operating in unknown environments with regions of few features
- ⇒ Applying an adaptive feature selection strategy
- ▶ Implementation on a real robot system: Robotino, a mobile robot with omnidirectional wheels
- ▶ VO algorithm programmed in C++ with OpenCV
- ▶ ROS for managing the whole system and robot control

Robotino



Visual Odometry Algorithm I



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Image preprocessing (undistortion and rectification)

Visual Odometry Algorithm I

- ▶ Image preprocessing (undistortion and rectification)
- ▶ Feature detection in the left and right image of the stereo camera

Visual Odometry Algorithm I

- ▶ Image preprocessing (undistortion and rectification)
- ▶ Feature detection in the left and right image of the stereo camera
- ▶ Extracting and matching descriptors of corresponding points

Visual Odometry Algorithm I

- ▶ Image preprocessing (undistortion and rectification)
- ▶ Feature detection in the left and right image of the stereo camera
- ▶ Extracting and matching descriptors of corresponding points
- ▶ ORB (Oriented Fast and Rotated Brief)

Visual Odometry Algorithm I

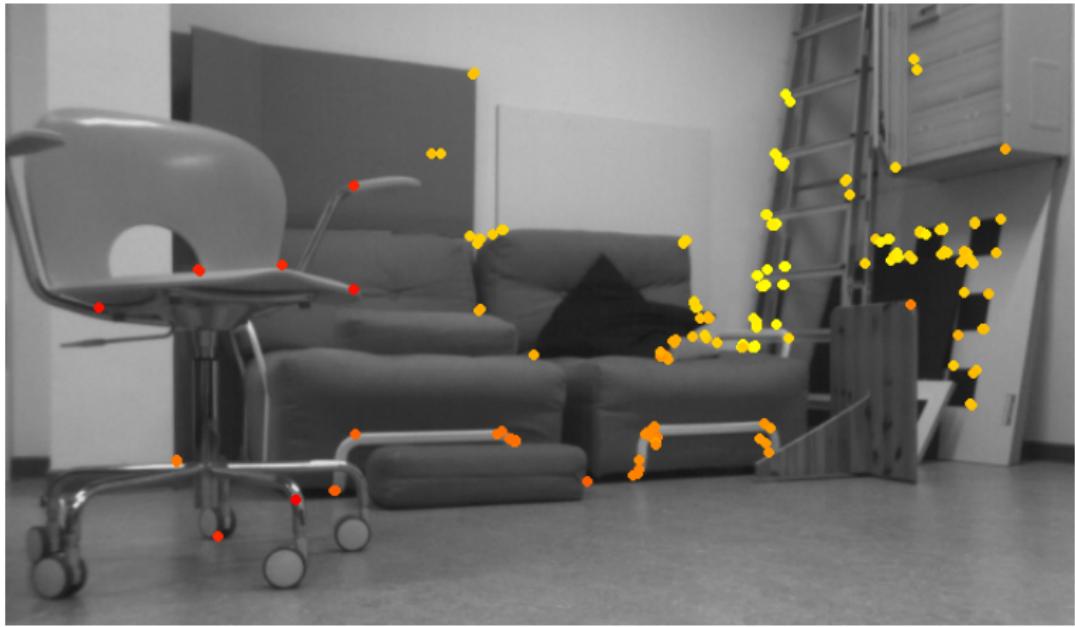
- ▶ Image preprocessing (undistortion and rectification)
- ▶ Feature detection in the left and right image of the stereo camera
- ▶ Extracting and matching descriptors of corresponding points
- ▶ ORB (Oriented Fast and Rotated Brief)
- ▶ Generating a set of 3D points with triangulation (linear method)

Visual Odometry Algorithm I

- ▶ Image preprocessing (undistortion and rectification)
- ▶ Feature detection in the left and right image of the stereo camera
- ▶ Extracting and matching descriptors of corresponding points
- ▶ ORB (Oriented Fast and Rotated Brief)
- ▶ Generating a set of 3D points with triangulation (linear method)

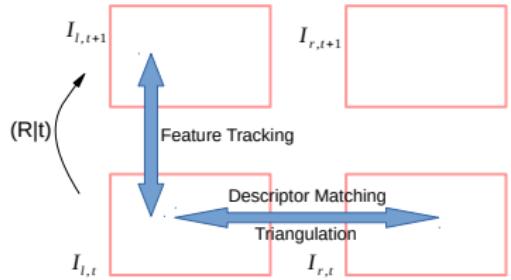


Visual Odometry Algorithm I



Visual Odometry Algorithm II

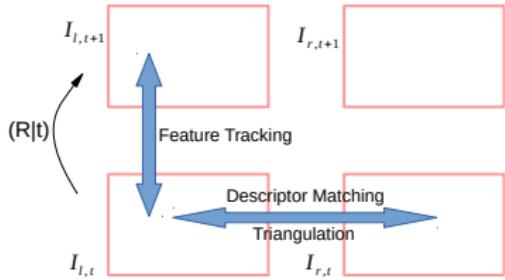
- ▶ Tracking of features in the next image with the Lucas-Kanade tracker



Visual Odometry Algorithm II

- ▶ Tracking of features in the next image with the Lucas-Kanade tracker
- ▶ Estimating the Pose ($R|t$) with the P3P algorithm

$$\min_{(R|t)} \sum_{i=0}^n \| \mathbf{x}_{k+1,i} - \mathbf{K} \cdot (R|t) \cdot \mathbf{x}_{k,i} \|^2$$

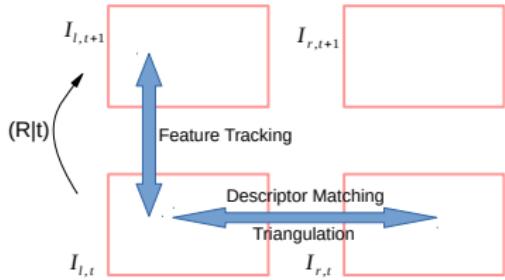


Visual Odometry Algorithm II

- ▶ Tracking of features in the next image with the Lucas-Kanade tracker
- ▶ Estimating the Pose ($R|t$) with the P3P algorithm

$$\min_{(R|t)} \sum_{i=0}^n \| \mathbf{x}_{k+1,i} - \mathbf{K} \cdot (R|t) \cdot \mathbf{X}_{k,i} \|^2$$

- ▶ Using RANSAC for outlier rejection

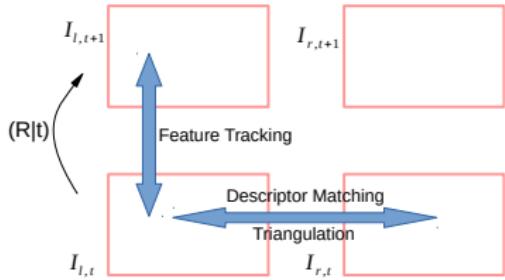


Visual Odometry Algorithm II

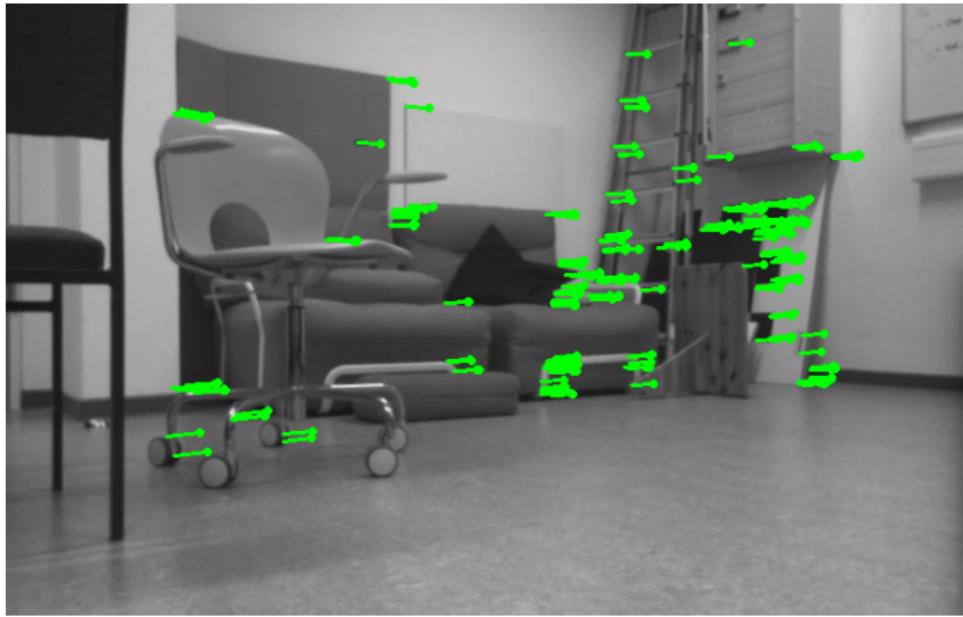
- ▶ Tracking of features in the next image with the Lucas-Kanade tracker
- ▶ Estimating the Pose ($R|t$) with the P3P algorithm

$$\min_{(R|t)} \sum_{i=0}^n \| \mathbf{x}_{k+1,i} - \mathbf{K} \cdot (R|t) \cdot \mathbf{X}_{k,i} \|^2$$

- ▶ Using RANSAC for outlier rejection
- ▶ Concatenating the current motion, to get the absolute position of the robot



Visual Odometry Algorithm II



Adaptive Feature Search

- ▶ VO algorithm is relying on features



Adaptive Feature Search

- ▶ VO algorithm is relying on features
- ▶ Unknown environments might always contain featureless areas



Adaptive Feature Search

- ▶ VO algorithm is relying on features
- ▶ Unknown environments might always contain featureless areas
- ▶ Dynamic feature search (rotate while moving)



Adaptive Feature Search

- ▶ VO algorithm is relying on features
- ▶ Unknown environments might always contain featureless areas
- ▶ Dynamic feature search (rotate while moving)
- ▶ Static feature search (initial feature search)



Adaptive Feature Search

- ▶ VO algorithm is relying on features
- ▶ Unknown environments might always contain featureless areas
- ▶ Dynamic feature search (rotate while moving)
- ▶ Static feature search (initial feature search)
- ▶ Robot control is independent from rotation angle



Implementation with ROS

- ▶ Robot Operating System (ROS): Flexible framework for developing robot software

Implementation with ROS

- ▶ Robot Operating System (ROS): Flexible framework for developing robot software
- ▶ Modules containing hardware drivers and many different robot functionalities

Implementation with ROS

- ▶ Robot Operating System (ROS): Flexible framework for developing robot software
- ▶ Modules containing hardware drivers and many different robot functionalities
- ▶ Executable programmes are called “Nodes”

Implementation with ROS

- ▶ Robot Operating System (ROS): Flexible framework for developing robot software
- ▶ Modules containing hardware drivers and many different robot functionalities
- ▶ Executable programmes are called “Nodes”
- ▶ Node communication via “Topics”

Implementation with ROS



Experimental Results



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Various experiments to test the performance of the adaptive VO algorithm

Experimental Results

- ▶ Various experiments to test the performance of the adaptive VO algorithm
- ▶ Accuracy in terms of relative rotational and translational error

Experimental Results

- ▶ Various experiments to test the performance of the adaptive VO algorithm
- ▶ Accuracy in terms of relative rotational and translational error
- ▶ KITTI Benchmark Suit for previous simulation

Experimental Results

- ▶ Various experiments to test the performance of the adaptive VO algorithm
- ▶ Accuracy in terms of relative rotational and translational error
- ▶ KITTI Benchmark Suit for previous simulation
- ▶ Robotino without feature search (in a feature rich environment)

Experimental Results

- ▶ Various experiments to test the performance of the adaptive VO algorithm
- ▶ Accuracy in terms of relative rotational and translational error
- ▶ KITTI Benchmark Suit for previous simulation
- ▶ Robotino without feature search (in a feature rich environment)
- ▶ Comparison of the dynamic and static feature search approach

Test Scenario	Rotational Error in °/m	Translational Error in %
KITTI data set 30m	0.35	0.9
KITTI data set 200m	0.03	2.5
Robotino no adaptive FS	~ 1	5
Robotino with dynamic FS	~ 2	~ 60
Robotino with static FS	~ 1.5	8

Conclusion and Outlook

- ▶ The developed VO algorithm yields reasonable results

Conclusion and Outlook

- ▶ The developed VO algorithm yields reasonable results
- ▶ Static adaptive feature search performs well, making the localisation more robust

Conclusion and Outlook

- ▶ The developed VO algorithm yields reasonable results
- ▶ Static adaptive feature search performs well, making the localisation more robust
- ▶ Additional error is introduced

Conclusion and Outlook

- ▶ The developed VO algorithm yields reasonable results
- ▶ Static adaptive feature search performs well, making the localisation more robust
- ▶ Additional error is introduced
- ⇒ Improvements for the future:

Conclusion and Outlook

- ▶ The developed VO algorithm yields reasonable results
- ▶ Static adaptive feature search performs well, making the localisation more robust
- ▶ Additional error is introduced
- ⇒ Improvements for the future:
 - ▶ Dealing with moving objects

Conclusion and Outlook

- ▶ The developed VO algorithm yields reasonable results
 - ▶ Static adaptive feature search performs well, making the localisation more robust
 - ▶ Additional error is introduced
- ⇒ Improvements for the future:
- ▶ Dealing with moving objects
 - ▶ Applying a filter (Kalman/Particle) to refine the results

Conclusion and Outlook

- ▶ The developed VO algorithm yields reasonable results
 - ▶ Static adaptive feature search performs well, making the localisation more robust
 - ▶ Additional error is introduced
- ⇒ Improvements for the future:
- ▶ Dealing with moving objects
 - ▶ Applying a filter (Kalman/Particle) to refine the results
 - ▶ Taking advantage of the 3D scene information

Thank you for your attention!

