Computer Vision for Mobile Robots in GPS Denied Areas

Michael Berli, 28th of April 2015
Supervisor: Tobias Nägeli
Robots can work in places we as humans can't reach

and they can do jobs we are unable or unwilling to do.
Autonomous mobile robots

- How do we make robots navigate autonomously?

Robots should be able to explore an unknown environment and navigate inside this environment without active human control.
Autonomous mobile robots

- Using computer vision for autonomous navigation
Robots
Focus in this talk

Type of robot
- Autonomous Ground Vehicles

Environment
- Indoor environments (rooms, tunnels, warehouses)

Sensors
- Cameras, wheel sensors
Robot scenarios: Industrial-Automation
Robot scenarios: Inspection & Discovery
Robot scenarios: Space operations
The three navigation classes

- Mapless
- Map-Based
- Map-Building
Mapless Navigation

Walk through Paris without colliding
Collision Avoidance
Optical Flow

- Describe the **motion of patterns** in successive images

![Diagram](image-url)
Optical Flow

- Get an understanding of **depth** in images
- **Time-To-Contact** between a camera and an object
Optical Flow: Time-To-Contact
Optical Flow: Time-To-Contact
Optical Flow: Time-To-Contact

Focus of Expansion
Where the camera points at
Optical Flow: Time-To-Contact

- Left Flow
- Central Flow
- Right Flow

$TTC_l$, $TTC_c$, $TTC_r$
Obstacle Avoidance FSM
Inspired by biology
Inspired by biology
Inspired by biology

Maximum of optical flow
Optical Flow: Further applications

- Applications for visually impaired
- Image Stabilization
- Video Compression (MPEG)

Drawbacks
- Hard if no textures
- Dynamic scenes?
The three navigation classes

Mapless

Map-Based

Map-Building
Map-Based Navigation

Use a map of Paris to navigate to champs élysée
Map-Based Navigation: Robot Scenario
Map-Based Navigation: Map Representation

**Topological Map**

*Graph-based representation of features and their relations, often associated with actions.*

- simple and compact
- no absolute distances
- obstacle avoidance needed

**Metric Map**

*Two-Dimensional space in which objects and paths are placed.*

- very precise
- hard to obtain and to maintain
Map-Based Navigation Example

Build a topological map of the floor

Use the topological map to navigate
Feature Extraction

- Features should be
  - Easily re-observable and distinguishable
  - Plentiful in the environment
  - Stationary

Elements which can easily be re-observed and distinguished from the environment
Room Identification

Signature Room F
Topological Map
Room Searching
Drawbacks and Extensions

- Learning and maintenance is expensive

- Use scanner tags or artificial beacons?
The three navigation classes

Mapless

Map-Based

Map-Building
Map-Building Navigation

Leave your hotel in Paris, explore the environment and return to the hotel afterwards
Map-Building Navigation

- Goal: in an unknown environment the robot can build a map and localize itself in the map

- Two application categories
  - Structure from Motion (Offline)
  - Simultaneous Localization and Mapping (SLAM) ← Real-Time!
Structure from Motion (Offline)

Robot moves around and captures video frames

Frame-To-Frame feature detection

3D Map and trajectory reconstruction

Pros

- Well studied
- Very accurate and robust solution

Cons

- Offline approach
- Changing environment requires new learning phase
Simultaneous Localisation and Mapping (SLAM)

- Build a map using **dead reckoning** and **camera readings**
- We focus on EKF-SLAM (Extended Kalman Filter)
A map built with SLAM
Dead Reckoning

- **Motion estimation** with data from odometry and heading sensors
Six steps of map-building (1/2)

(1) Initialise feature A.

(2) Drive forward

(3) Initialise B. and C.
Six steps of map-building (2/2)

(4) Drive back

(5) Re-measure A.

(6) Re-measure B.
EKF-SLAM: The system

This system is represented by
- System state vector
- System covariance matrix
EKF-SLAM: The state vector

\[ \hat{x} = \begin{pmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \end{pmatrix} \]

\[ \begin{pmatrix} x_r \\ y_r \\ \theta_r \end{pmatrix} \]

\[ \hat{y}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \]

\[ \hat{y}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \]

\[ \hat{y}_3 = \begin{pmatrix} x_3 \\ y_3 \end{pmatrix} \]
EKF-SLAM: The covariance matrix

\[ P = \begin{bmatrix}
  P_{xx} & P_{xy1} & P_{xy2} & \cdots \\
  P_{y1x} & P_{y1y1} & P_{y1y2} & \cdots \\
  P_{y2x} & P_{y2y1} & P_{y2y2} & \cdots \\
  \vdots & \vdots & \vdots & \ddots 
\end{bmatrix} \]
SLAM Process

Robot moved

PREDICTION of Robot position

PREDICTION of observed features

ESTIMATION of updated robot position

EKF Fusion

Match predicted and observed features

Feature Extraction

Camera
Motion model

- Estimate robot’s new position after a movement

Motion model

\[ x_v = f_v(\hat{x}_v, u) \]

- Estimated robot position
- Old position
- Odometry

\[ f_v \]
SLAM Process

Robot moved

PREDICTION of robot position

PREDICTION of observed features

EKF Fusion

Match predicted and observed features

Feature Extraction

ESTIMATION of updated robot position

Camera
Measurement model

- Based on the predicted robot position and the map, use a measurement model to predict which features should be in view now
SLAM Process

Robot moved

PREDICTION of robot position

PREDICTION of observed features

ESTIMATION of updated robot position

EKF Fusion

Match predicted and observed features

Feature Extraction

Camera
Data matching

- Match predicted and observed features
SLAM Process

Robot moved

PREDICTION of robot position

PREDICTION of observed features

ESTIMATION of updated robot position

EKF Fusion

Match predicted and observed features

Feature Extraction

Camera
EKF Fusion

Prediction

Camera

Residual
EKF Fusion
EKF Update

\[ \hat{x} = \begin{pmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \end{pmatrix} \]

\[ P = \begin{bmatrix} P_{xx} & P_{xy_1} & P_{xy_2} & \cdots \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2} & \cdots \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2} & \cdots \\ \vdots & \vdots & \vdots & \cdots \end{bmatrix} \]
SLAM – Research topics

- Robustness in changing environments
- Multiple robot mapping
Motion estimation of agile cameras

- Real-Time SLAM with a Single Camera
  - Andrew J. Davison, University of Oxford, 2003

- Parallel Tracking and Mapping for Small AR Workspaces
  - Georg Klein, David Murray, University of Oxford, 2007
Motion estimation of agile cameras

- No odometry data, **fast and unpredictable** movements
- Use a constant velocity model instead of odometry

\[ x_v = (x \ y \ z \ \alpha \ \beta \ \delta \ v_x \ v_y \ v_z \ v_\alpha \ v_\beta \ v_\delta ) \]

- Position
- Orientation
- Velocity
Motion estimation of agile cameras

- Real-Time SLAM with a Single Camera
  - Andrew J. Davison, University of Oxford, 2003

- Parallel Tracking and Mapping for Small AR Workspaces
  - Georg Klein, David Murray, University of Oxford, 2007
Tracking and Mapping for AR Workspaces
What we have seen

- What autonomous mobile robots are used for

- How today's mobile robots navigate autonomously
  - mapless, map-based, map-building

- The potential and the challenges of SLAM


References

Papers


References

Images & Videos
1. https://www.youtube.com/watch?v=ISznqY3kESI
6. http://cnet4.cbsistatic.com/hub/i/r/2014/12/01/b1baf339-67d6-4004-bc66-7dd34c11a870/resize/770x578/3d17e8de0dbd6d26cbf13e53a6c0b655/amazon-kiva-robots-donna-7611.jpg
ref=Grey_Coast_Media&ref=Grey_Coast_Media&clickthrough_id=415192702&redirect_back=true
19. http://www.robots.ox.ac.uk/~ajd/Movies/realtime 30fps slam.mpg
22. https://www.bcgperspectives.com/content/articles/business_unit_strategy_innovation_rise_of_robotics/