A Hybrid Kalman Filter Based Algorithm for Real-time Visual Obstacle Detection

Erik Einhorn
Ch. Schröter, H.-J. Böhme, H.M. Gross

Ilmenau Technical University (Germany)
Neuroinformatics and Cognitive Robotics Lab

Outline

- **Introduction**: challenges of monocular scene reconstruction on mobile robots
- **Overview** of system architecture and selected components
- **Results of experiments** in monocular scene reconstruction for obstacle detection
- **Summary**
A Hybrid Kalman Filter Based Algorithm for Real-time Visual Obstacle Detection

Shape-from-motion

Obstacle avoidance

reconstructed 3D model (point cloud)
Kalman filter based scene reconstruction

- separate Kalman filter for each model point / image feature
- unknown 3D position $X$ is chosen as state vector of each Kalman filter
- the image feature is tracked in consecutive frames
- with each iteration the 3D position will be estimated more precisely
Initialisation of the Kalman filter

- „classical“ multi-baseline stereo approach

- search for correspondences along the epipolar lines in previously recorded images

- explicit triangulation is not necessary if a parametric description of the epipolar line is used
Initialisation of the Kalman filter

- an initial estimate for the 3D position of the model point is computed using the estimated depth of the image feature
- while the feature is tracked in the next frames the Kalman filter will correct and refine the estimated 3D position of the model point
A Hybrid Kalman Filter Based Algorithm for Real-time Visual Obstacle Detection

Architecture

- Image sequence
- Feature tracking
- Select new features
- Track features
- Scene reconstruction
- "Classical" depth-estimation
- Shape-from-motion using Kalman filters
- Feature pool
- Inactive features
- Active features
- New f.
- Tracked f.
- Lost f.
Convergence of the scene reconstruction

![Graph showing convergence of mean model error with iterations for different approaches. The graph includes lines for hybrid approach (with initial depth estimation), Kalman filter with constant initialisation, with random initialisation, with initialisation on the ground plane, and plain depth estimation without Kalman filtering. The y-axis represents mean model error (in m) and the x-axis represents iterations.]
Feature selection and tracking

Feature selection:

- for the selection of appropriate image features the „FAST corner detector“ [1] is applied (FAST = Features from Accelerated Segment Test)
- it is the most computationally efficient feature detector available

Feature tracking:

- correspondent features are tracked using a guided feature matching approach

Guided feature matching

1. features are selected independently in each frame

2. possible matching hypotheses $H$ are selected

3. evaluation of each hypothesis using a cost function

4. best hypotheses are chosen using a greedy algorithm
Cost function

\[ \bar{x}_t^* = P_t \hat{X}_t^* \]

\[ \text{cost}(x_t^{(i)}, x_t^{(j)}) = w_1 \left\| x_t^* - x_t \right\|_2^2 \]
Cost function

\[ \text{cost}(x_t^{(i)}, x_t^{(j)}) = w_1 \left\| x_t^{* (i)} - x_t^{(j)} \right\|^2 \]

1. estimated 3D position
Cost function

\[ C(x_t, x_{t-1}) = d(x_t^*, x_t) + w_2 \cdot SAD_W(x_{t-1}^{(i)}, x_t^{(j)}) \]
Cost function

\[ C(x_t) = \alpha \cdot \text{estimated 3D position} + \beta \cdot \text{similarity (SAD)} + \gamma \cdot w_2 \cdot \text{SAD}_W(x_{t-1}, x_{t}^{(j)}) \]
Cost function

\[ d_{epi} = \frac{\tilde{x}_t^{(j)\top} F \tilde{x}_{t-1}^{(i)}}{\sqrt{(F \tilde{x}_{t-1}^{(i)})_1^2 + (F \tilde{x}_{t-1}^{(i)})_2^2}} + w_3 \cdot d_{epi}(\tilde{x}_t^{(i)}, \tilde{x}_t^{(j)}) \]
Cost function

\[ \text{Cost function} = \sum_{i} \left( x_{t}^* - d_{xt} \right)^2 \]

1. estimated 3D position \( x_{t-1} \)
2. similarity (SAD) \( x_{t}^* \)
3. epipolar distance

\[ + w_3 \cdot d_{epi}(x_{t-1}^{(i)}, x_{t}^{(j)}) \]
Cost function

1. estimated 3D position

2. similarity (SAD)

3. epipolar distance

\[
\text{cost}(x_t^{(i)}, x_t^{(j)}) = w_1 \left\| x_t^{*(i)} - x_t^{(j)} \right\|^2_2 \\
+ w_2 \cdot \text{SAD}_W(x_{t-1}^{(i)}, x_t^{(j)}) \\
+ w_3 \cdot d_{epi}(x_{t-1}^{(i)}, x_t^{(j)})
\]
Comparison with KLT tracker

http://www.ces.clemson.edu/~stb/klt/
Demonstration

E. Einhorn: A Hybrid Kalman Filter Based Algorithm for Real-time Visual Obstacle Detection
Map building: laser vs. visual
Summary

- hybrid shape-from-motion approach, that combines "classical depth estimation" and scene reconstruction based on Kalman filters

- can be used for obstacle detection / avoidance

- combining both approaches compensates the disadvantages of each method

- feature tracker based on guided feature matching
Thanks for your attention!
Problems and Challenges

– camera is moving along its optical axis
  – scene reconstruction must be performed over a long base distance in order to achieve reliable estimates

most obstacles are only visible for a few frames of the image sequence
  – reconstruction algorithm must be able to provide reliable estimates using a few image measurements only

proper initialization of the Kalman filter state vectors
Comparison with KLT tracker

![Graph showing comparison]
Runtime of the scene reconstruction approach

measured using an Intel Pentium III 3.4 GHz