SeqSLAM\textsuperscript{[1]}

Final Presentation and Live Demo

Tim Wiese, Mathias Kanzler, Maxi Weber

TU München: Applied Computer Vision for Robotics SS 2013
The Problem

Recognition of recurring sceneries under changing environmental conditions:

- Camera input
- Saving compressed frames
- Applying algorithm to search for suitable matches
- Several improvements to fulfill given requirements
Preprocessing

Store compressed information without losing relevant information:

- Convert to grayscale
- Scale down to $64 \times 36$
- Contrast enhancement: stretch histogram, patch based
Matching

- Calculate image difference value for each template
- Local Contrast Enhancement
- Localized Sequence Recognition
Matching

- Calculate image difference value for each template
- Local Contrast Enhancement
- Localized Sequence Recognition
Matching

- Calculate image difference value for each template
- Local Contrast Enhancement
- Localized Sequence Recognition

Image difference matrix
Enhancements - Multithreading

Thread 1: Decoding and Preprocessing
Ring Buffer
Thread 2: Matching
Thread 3: Output
Enhancements - Stationary Frame Detection
Results

- Successful implementation of SeqSLAM
- Robust matching under changing conditions
- Multiple times faster than realtime performance
Outlook

- SeqSLAM could be used as independent visual positioning system
- Further improvements by intelligent template learning
  - Learning rate coupled with vehicle speed
  - Online aggregation of subsequent runs with changing environment
## Schedule

| Week 1 | ● Read paper  
  ● Understand topic  
  ● Create schedule | Done!  
  Done!  
  Done! |
|---|---|---|
| Week 2 | ● Start implementation  
  ● Preprocess datasets and create bagfiles  
  ● Generate first basic output (matches side by side) | Done!  
  Done!  
  Done! |
| Week 3 | ● Implementation: final stage  
  ● Testing  
  ● Bug fixing  
  ● Evaluation of matching quality and performance | Done!  
  Done!  
  Done!  
  Done! |
| Week 4 | ● Parameter tuning  
  ● Improvements (quality / performance)  
  ● Enhancements  
  ● Final presentation | Done!  
  Done!  
  Done!  
  Done! |
# Schedule

| Week 1 | ● Read paper  
  ● Understand topic  
  ● Create schedule | Done!  
  Done!  
  Done! |
|---|---|---|
| Week 2 | ● Start implementation  
  ● Preprocess datasets and create bagfiles  
  ● Generate first basic output (matches side by side) | Done!  
  Done!  
  Done! |
| Week 3 | ● Implementation: final stage  
  ● Testing  
  ● Bug fixing  
  ● Evaluation of matching quality and performance | Done!  
  Done!  
  Done!  
  Done! |
| Week 4 | ● Parameter tuning  
  ● Improvements (quality / performance)  
  ● Enhancements  
  ● Final presentation | Done!  
  Done!  
  Done!  
  Done! |
The End