Stereo SLAM
(Simultaneous localization and Mapping)

Group: A-Team

- Goal: accurate dense point cloud representation of the world

**Step 1**
- Grid-FAST Detector
  - User specified
    - grid size
  - points per cell
  - Decrease Workload for Descriptor

**Step 2:** 3D Feature Map of the environment
- Not only current time step n but also time step n-1.
- All previous time steps, up to now.
- Concatenate currently detected 3D features (Keyframe)
- Keyframe = description with all previous ones.
- Complete and robust.
- Many past points are not visible.
- New data structures: Keyframe

**Step 3:** map
- We use our digital world.
- A combination of point cloud.
- Our data set contains all keyframes even location.
- Map of our world is complete.
- Robust solution.
- Collect data contribution.
- Test the absolute error of each worldframe.
Stereo SLAM
(Simultaneous localization and Mapping)

Group: A-Team

Step 1
- Grid-FAST Detector
- User specified
  - grid size
  - points per cell
- Decrease Workload for Descriptor

Step 2 - 3D Feature Map of the environment
- Not only current time step but also timestamps before
- All previous time steps up to now
- Compare currently detected 3D features (keypoints)
  - keypoint descriptions with all previous ones
  - Compute similarity w.r.t. position
- Many past points are not visible
- New 3D structure: Keyframe
• Grid-FAST Detector

Step 1

• Goal: accurate dense point cloud representation of the world

Group: A-Team
Step 1

- Grid-FAST Detector
- User specified
  - grid size
  - points per cell
- Decrease Workload for Descriptor
Step 2 - 3D Feature Map of the environment

- Not only current time step \( t[n] \) and last timestamp \( t[n-1] \)
- All previous time steps \( t \) up to now
- Compare currently detected 3D features (Point3d + keypoints + descriptors) with all previous ones
  - computationally expensive
  - many past points are not visible
- New data structure: Keyframe
• Keyframe
  • Aggregate 3D features from multiple views
  • Stores transformation
    • world to current Keyframe $K_c$
• Match currently detected 3D features
• only against closest Keyframe $K_c$
• New Keyframe
  • Ratio: detected points with / without matches
  • After certain distance (e.g. 1m)
  • Every x frames
  • All points from last Keyframe visible at the current position
  • Set new Keyframe transformation
Step 3 - dense map

- At this stage: resulting map very sparse
  - Represented as point cloud

- Use dense point cloud at every Keyframe (exercise 4)
  - Integrate resulting dense point clouds into the map
    - Result: dense map

- Dense (stereo) reconstruction
  - Run in a separate thread for each new Keyframe
Step 4 (optional)

- Improve performance via
  - Sliding Window Optimization
  - Bundle adjustment

- Several opportunities are included in GTSAM and g2o
Stereo SLAM (Simultaneous localization and Mapping)

Group: A-Team

- Goal: accurate dense point cloud representation of the world

Step 1
- Grid-FAST Detector
- User specified
  - grid size
  - points per cell
- Decrease Workload for Descriptor

Step 2: 3D Feature Map of the environment
- Not only current time stamp but and past timestamp
- All previous time steps up to now
- Compress (previously detected) 50 features (Image + Keyframe + description) with all previous ones
- Computationally expensive
- Many pixel points are not visible
- New data structure: Keyframe

Step 3: 3D Map
- All we have collected features are stored
- The point cloud is basically the 3D map
- Use point cloud at every keyframe even though it is noisy
- New realization: common in dynamic environment
- Several errors correction
- Not the exact distance of each feature/feature