Keyframe-Based SLAM with Bundle Adjustment

Team: gang-of-three
Video!!!
Structure

- **Thread 1 - Frontend**
  - Tracking
  - Camera pose estimation
  - KeyFrame Selection / Criteria

- **Thread 2 - Backend**
  - Map Management
  - Point Cloud Publishing

- **Thread 3 to n - Optimizer**
Frontend - Tracking

- BRISK / AGAST

- Dynamic Keypoint Threshold
  - keep the number of Keypoints in a constant range

- Window Matching
  - depends on Key Frame Criteria
Frontend

● Pose Estimation
  ○ RANSAC
  ○ Umeyama

● Key Frame Criteria
  ○ Only if enough Matches
  ○ Average Reprojection Error (between map and current frame)\cite{1}
    ■ Error Threshold: 0.5 Matching Window Size
  ○ Translation bigger than 20 cm
Backend

- **Map**
  - Observations
    - pixel position, feature descriptor
  - Shared pointers to relate KeyFrames, Observations and Points
  - Ensured thread save access
  - Points and KeyFrame poses in global frame
Backend

- Map Management
  - Add new KeyFrame
  - Compute local maps for Frontend and Optimizer
  - Notify when map changed
Optimizer

- Ceres Cost Function: reprojection error

- Challenge using Ceres:
  - Not thread-safe
  - Solution: Synchronisation wrapper classes
Conclusion

● **Achievement**
  ○ mapping and tracking working well with careful camera motion

● **To be done**
  ○ Code cleanup / refactoring
  ○ Local map for optimizer
  ○ Robust performance for different scenarios (BMW bagfile)
  ○ Loop closures
Reference

[1]: Hauke Strasdata, Andrew J. Davison, J.M. M. Montielb, Kurt Konoligecc. Double Window Optimisation for Constant Time Visual SLAM