Visual SLAM

Helen Oleynikova

F. W. Olin College of Engineering

August 18th, 2010
What is Visual SLAM?

- **SLAM**
  - Simultaneous localization and mapping
  - Create a map of environment while localizing within that map

- **VSLAM vs. GSLAM**
  - 3D instead of 2D
  - Images instead of laser scans
Components

- Feature detection/matching
  - Find keypoints in image
  - Match to keypoints in other images
- Visual odometry
  - Pose estimation from matched features between images
- Sparse bundle adjustment (SBA)
  - Large-scale optimization on 3D point positions and camera poses
- Place recognition
  - Finding matches between current frame and a set of previous frames
  - Loop closure
System Overview

- **New image(s)**
  - Feature Matching
  - **Visual Odometry**
    - Pose Estimation
    - Short-term SBA
  - **VSLAM**
    - Place Recognition
    - Long-term SBA

- **Previous image(s)**
  - Camera pose estimate
  - Point position estimates

- **keyframe**
What I Did This Summer

• Learned about SBA, VSLAM, ROS, etc.
• API Revisions
  • Library-level design
  • Indexing structure
  • Representation of projections within SBA
• Documentation
• Released vslam stack to 0.1
What I Did This Summer

• ROS Integration
  • Standalone nodes
    • SBA
    • Stereo VSLAM
    • Visual Odometry
  • Integration with navigation stack
    • Publish odometry to robot_pose_ekf
    • Publish TF
• Pointcloud data integration
Results
Mapping
Mapping
Integrating Pointcloud Data

- Different sensors
  - LIDAR
  - SwissRanger
  - PrimeSense
- Multi-sensor fusion
  - Use pose estimates and constraints from different sensors
Point-to-point
- Matches between keypoints in images
  - Exact correspondence

Point-to-plane
- Matches between pointclouds
  - Points aren’t exact matches
Implementation

- **Point-to-point matches**
  - Cost measurement is reprojection error

- **Point-to-plane matches**
  - Consider only error in direction of the point’s normal
  - Allows the points to move freely within the plane
  - Requires that point matches be locally planar
    - Filter matches based on curvature

- **Use thin covariance matrix with error calculation**
  - Rotate covariance matrix
  - Easy implementation within existing system
Applications

• 3D odometry, localization, and mapping

• Model building
  • For small objects, use stereo views from different angles
  • For large objects, like rooms, use image and laser scan data
    • Pointcloud matching
Future Work

• Persistent map
• Pose-to-pose constraints
  • Use odometry estimates
• Whole-room registration
• Monocular pose estimation
• Release vslam stack to 0.2
Thank you!