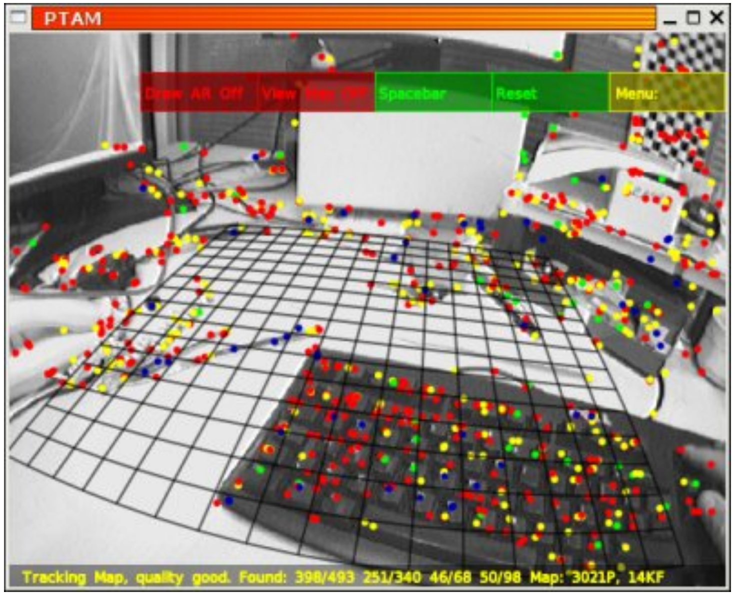
Report of Modifications to Parallel Tracking and Mapping (PTAM) for Outdoor Localization

September 3, 2013

Daniel DeTone
Yu Xiang
Silvio Savarese

Original PTAM



```
Tracking Map, quality good. Found: 398/493 251/340 46/68 50/98 Map: 3021P, 14KF
```

- Camera tracking system for augmented reality
- Designed for use in small, indoor workspaces
- Algorithm is split into two parts: Tracking and Mapping, for efficiency
- Real time operation, robust to jerky camera motion

Original PTAM Overview

Tracker

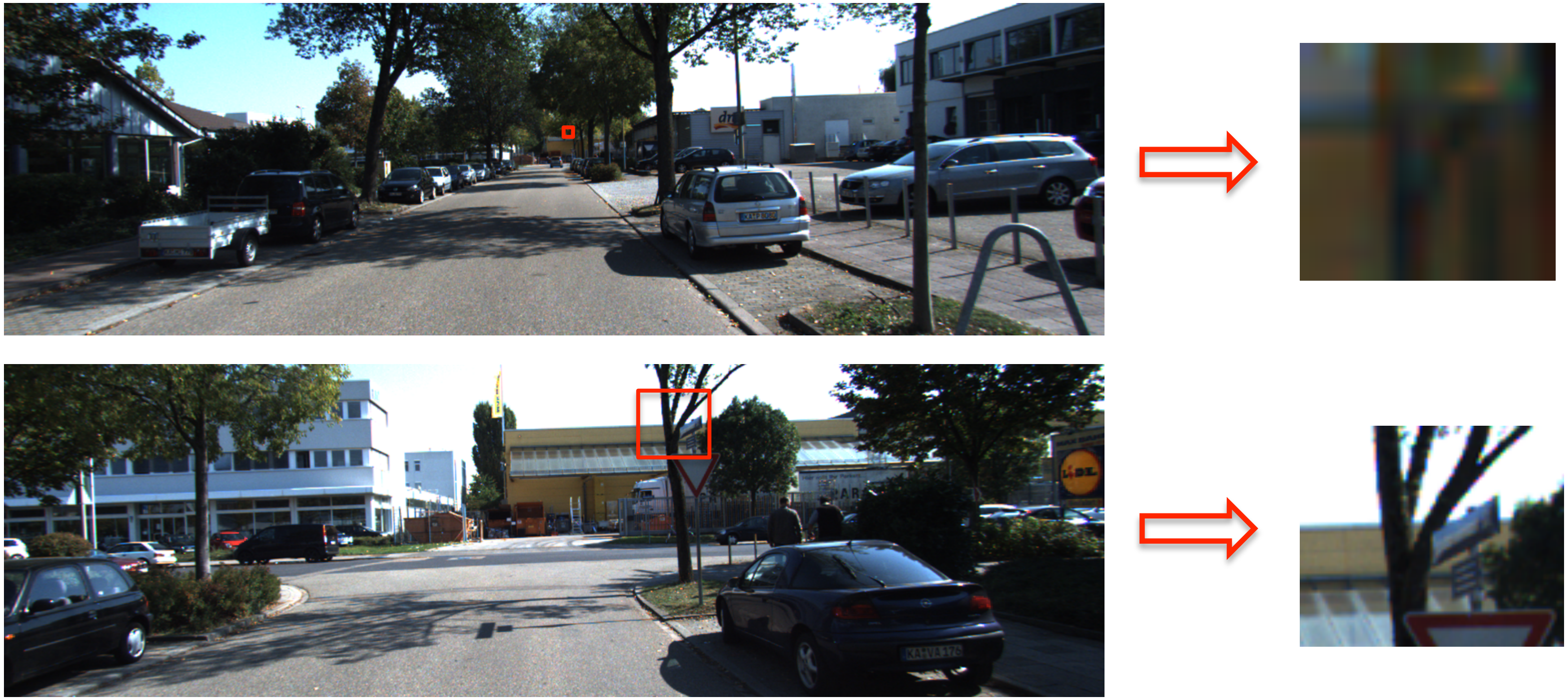
1. A pose estimate is generated from the motion model for a new input frame
2. Using this estimate, a small number of points (50 coarse features) are projected onto the image and searched for
3. The pose estimate is updated
4. A large number of features at varying scales (1000 points, sampled randomly from entire set) are project onto the image and searched for (if a point is projected to be off frame, it will be skipped)
5. The pose estimate is updated

MapMaker

1. Creates new map points using high-scoring FAST corners that are found in previous frames (epipolar search is used for depth information)
2. Bundle adjustment (BA) is performed. BA minimizes positional error of keyframes and mapoints. Two types of BA are performed:
    1. *Local BA* – input to BA is the nearest keyframes (KF) (default is four nearest neighbors)
    2. *Global BA* – input to BA is all keyframes (lower priority than local BA)

Outdoors versus Indoors

- A couple hypothesis were used to drive code modifications, in outdoor exploration:
  1. Once observed features leave the field-of-view, they are unlikely to be re-observed
  2. Features are likely to change appearance after many new KFs due to scale changes (i.e. a branch may appear very different from 50 m away versus 1 m away, see below)



Modification Summary

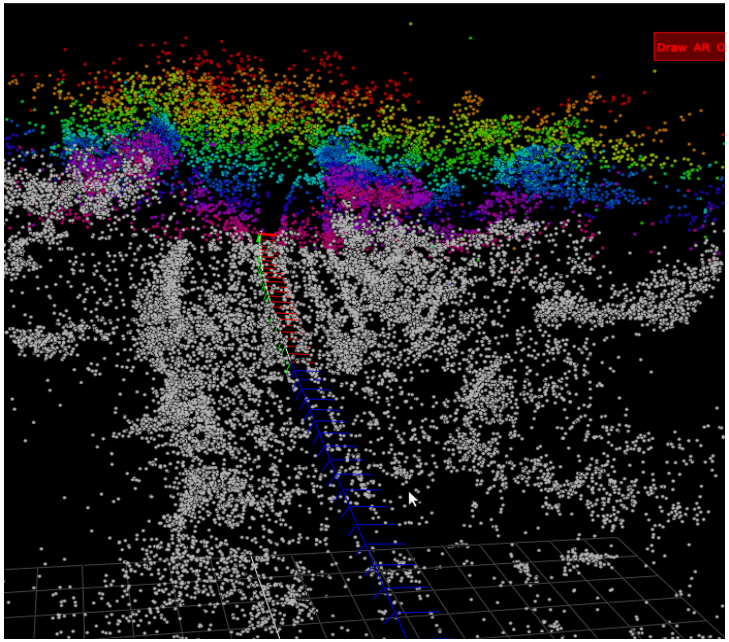
- Functionality
  1. Input to program is stored images, not real-time video
  2. Delay added in between frames [5 sec, use ‘p’ for pause]
- Algorithm
  3. Limit map point lifetime to being very short [4 KFs]
  4. Flag non-recent KFs as “fixed” in the global adjustment process [25 KFs away or more]
  5. Limit BA iterations in initialization [20 iterations]
- Aesthetic
  6. Rescale the dimensions [from 10cm to 1m]
  7. Camera follow [use ‘f’ key]
  8. Colored depths

Functionality Changes

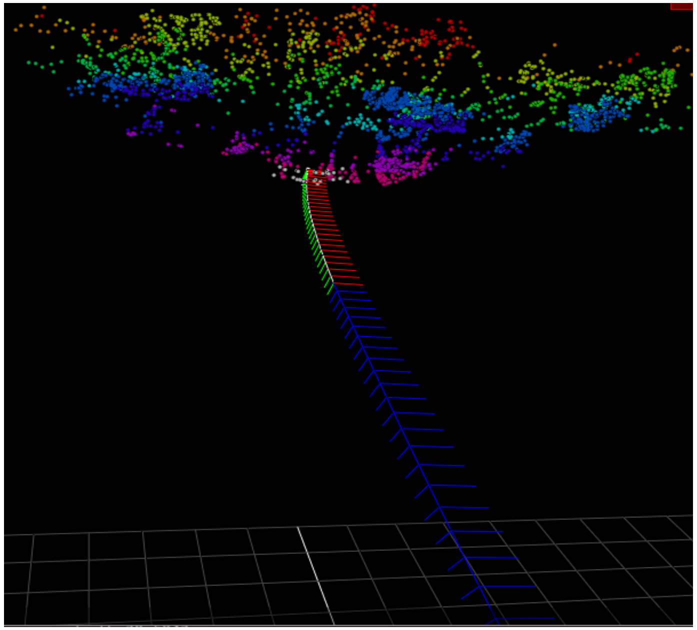
1. Input is stored images
   - Allows for testing with datasets
   - *VideoSource_File.cpp, VideoSource.h*
2. Delay added between frames
   - This gives time for the MapMaker thread to iterate multiple times which leads to more accurate results
   - Experimentally set delay to 5 sec
   - Use space for initialization, ‘p’ for pause
   - *Tracker.cpp [lines ~190-210]*

Algorithm Changes

3. Limit map point lifetime to being very short
   - This greatly improves performance for two reasons. First, a higher percentage of points in the map are projected into the field of view (i.e. fewer iterations of point projections are skipped, resulting in more efficient code). Second, map points are not subject to major appearance change as their lifetime is very short
   - Experimentally set lifetime as 4 KFs
   - *MapMaker.cc [lines ~190-195]*



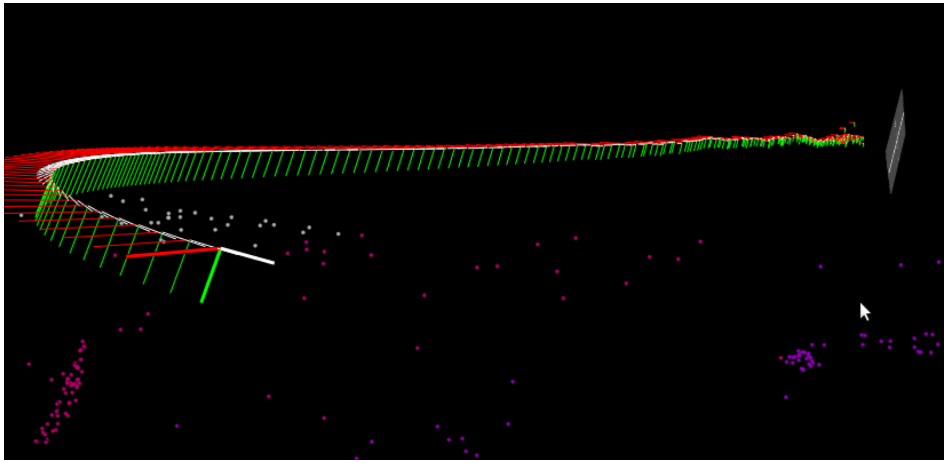
Original



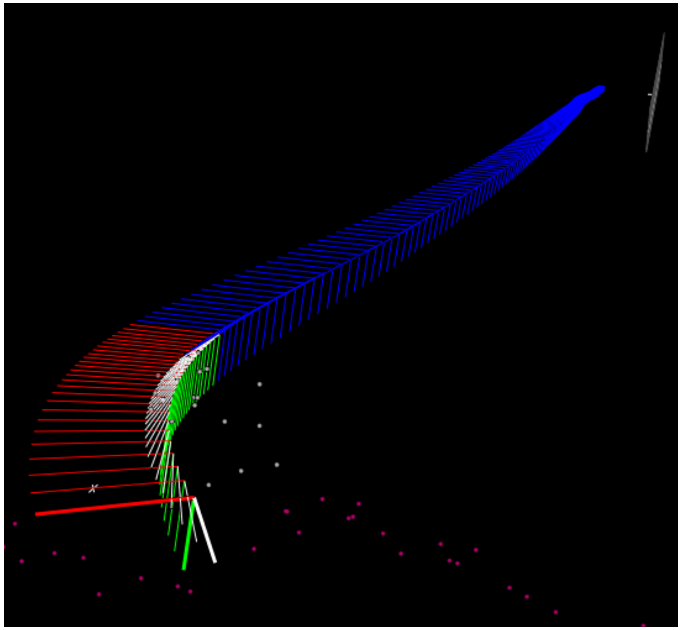
Modified

Algorithm Changes

4. Flag non-recent KFs as "fixed" in the global adjustment process
   - This prevents erroneous corrections to KFs in the distant past
   - Experimentally set to 25 KFs away or more (fixed KFs are blue)
   - *Tracker.cc [lines ~1040-1055]*



Original



Modified

Algorithm Changes

5. Limit BA iterations in initialization
   - In certain test cases, the bundle adjustment failed to converge, resulting in a failed initialization. Experimentally set to maximum of 20 iterations
   - *MapMaker.cc [lines ~400-405]*

Aesthetic Changes

6. Rescale the dimensions from 10cm to 1m
   - Displacement between initialization frames was originally assumed to be 10cm
   - Experimentally set to 1m
   - *MapMaker.cc [line ~41 (wigglescale)] and MapViewer.cc [line ~17]*
7. Camera follow
   - Use the ‘f’ to update GLcamera to follow estimated camera position
   - *MapViewer.cc [lines ~194-200]*
8. Colored depths
   - Better visualization of scene
   - *MapViewer.cc [lines ~40-60]*

Continued Work

- Modify initialization process to better estimate ground plane
  - Investigate poor depth estimation after initialization
- Reduce delay time and find bottlenecks
- Evaluate KITTI dataset with GPS ground truth

Notes

- All line numbers reference commit hash bbca60c8a817ccbc826f8b1bbe79d9def1d120b2 in master branch (tag “v1.3”)