Online Point cloud Compression mapping for visual odometry and SLAM

8/24/2017

Offline Compression Technique

- [2] Goal: produce a reduced map version good for visualization.
- [3] Goal: improve localization through relocalization by conducting memory-based learning to achieve visibility prediction.
- [4] reduce the search space by evaluating features
- [5] key-frame selection
- [6] Bag of words
- [7]

Online Technique

- Step 1: constructing submap with k camera and its associated features.
- Step 2: feed the submap constructed in step 1 into compression process
  - 2.1: interpolate trajectory to reduce uncertainty from noise and mislocation.
  - 2.2: use NURBS(non-uniform rational b-splines) to model the curve through a series of control points.
  - 2.3 for each control point, take all the trajectory cameras in the segment.
  - 2.4 for each control point, take all the associated features in the segment.

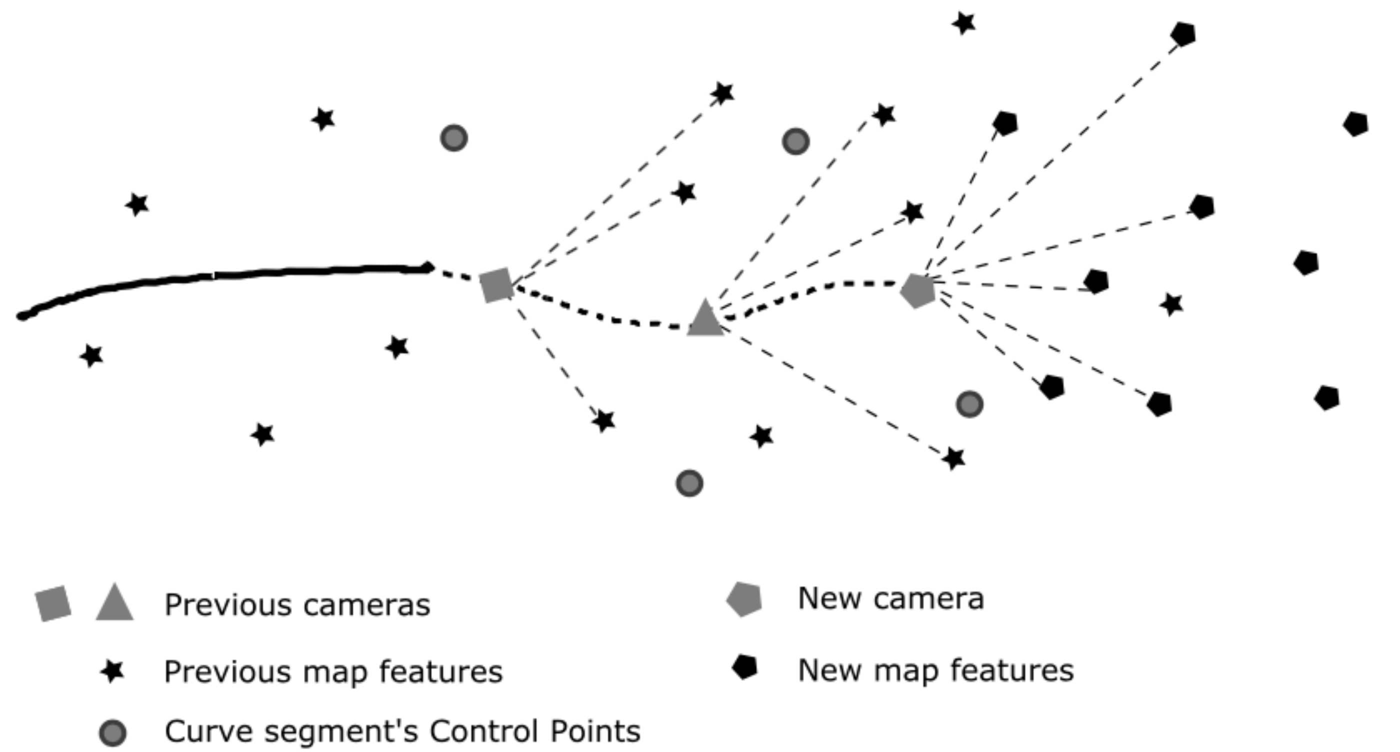


Fig. 2: O-POCO mapping process. We generate a sub-map of new features and those features in the global map which are visible from from the previous n cameras. We model the camera trajectory using the last n cameras. We aim to compress this sub-map and add the remaining points to the global map.

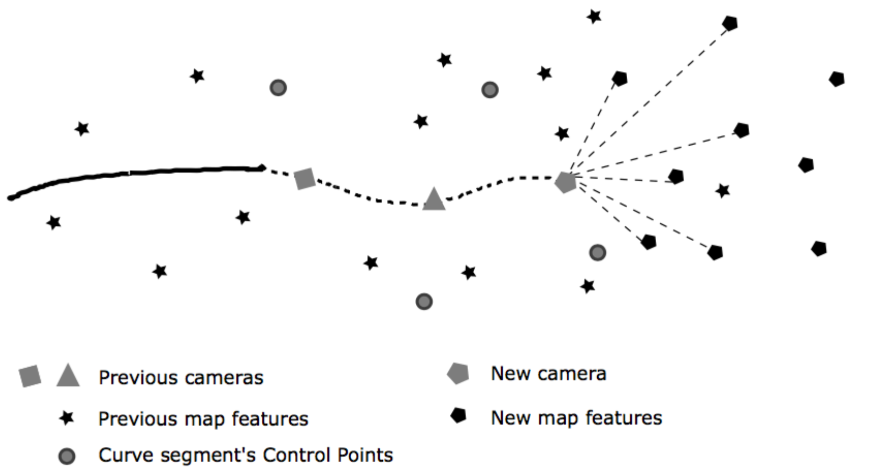


Fig. 3: Alternative key-frame mapping. We generate a sub-map by taking only the latest features and, as in the previous case, and model the camera trajectory from the last n cameras. Similarly, we compress this sub-map and add to the global map the selected points.

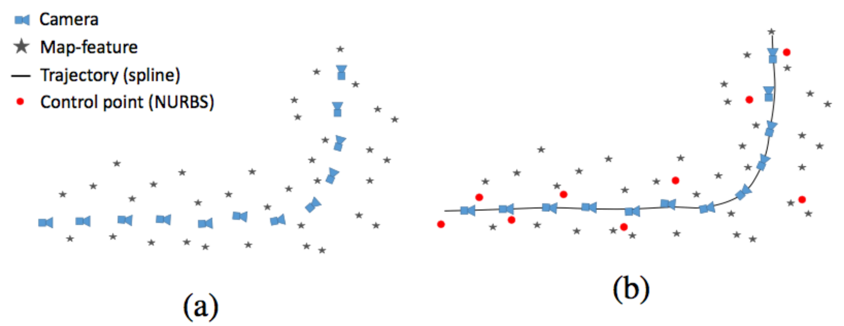
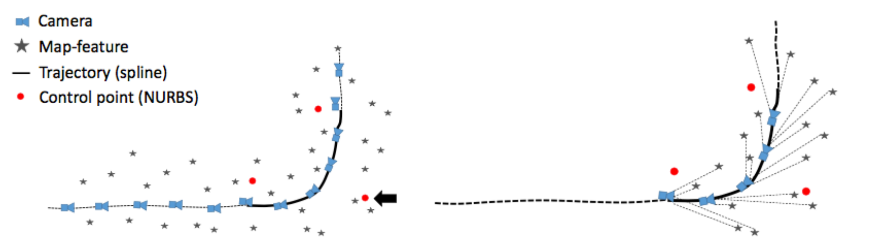


Fig. 4: We first model the camera trajectory from a series of cameras in (a) by interpolating the curve that best fits all n cameras in the window – reducing the uncertainty from noisy positions and mislocated cameras – and then modeling this curve through a series of control points using NURBS, as shown in (b).



(a) (b)

Fig. 5: In the subsetting process, for each control point in the curve model, e.g. the point indicated by the black arrow in a), we take all the cameras on the segment of the trajectory they represent – indicated as a solid line – and select all the map features those cameras can see, as shown in b).

Require: cameras $cams$, point cloud pc, number of words NoW
Ensure: point cloud subsets $subset$

$poly$ = spline($cams$)
cp = NURBS($poly$)
for all $cams$ **do**
  $segment$ = getsegment($cp(i-1)$, $cp(i+1)$)
  min_c = getmincam($segment$)
  max_c = getmaxcam($segment$)
  for $cam = min_c{:}max_c$ **do**
    $subset_i = subset_i \cup$ visible(pc, cam)
  end for
end for

Fig. 6: The Trajectory based Subsetting generates a series of point cloud subsets, based on the control points from the traveled trajectory.

Map compression

Require: frame sequence $frame$, compression rate q, window size win
Ensure: compressed map map

```
if i < win then
    (map_i, cam_i) = SfM( frame_i )
    map = map ∪ map_i
end if
if i == win then
    (map_i, cam_i) = SfM( frame_i )
    map = map ∪ map_i
    subset = TbS( map, cam_i )
    n = count(subset), size = q * n
    centroids = clustering(subset[position], size)
    map = knn(subset[position], centroids)
end if
if i > win then
    case key-frame
        (map_tmp, cam_i) = SfM( frame_i )
    case windowing
        (map_i, cam_i) = SfM( frame_i )
        map_tmp = map ∪ map_i
    (map_sub, cam_sub) = SUBMAP(map_tmp, cam_{i-win}:cam_i)
    subset = TbS( map_sub, cam_sub )
    n = count(subset), size = q * n
    centroids = clustering(subset[position], size)
    map_i = knn(subset[position], centroids)
    map = map ∪ map_i
end if
```

Experiments and Results

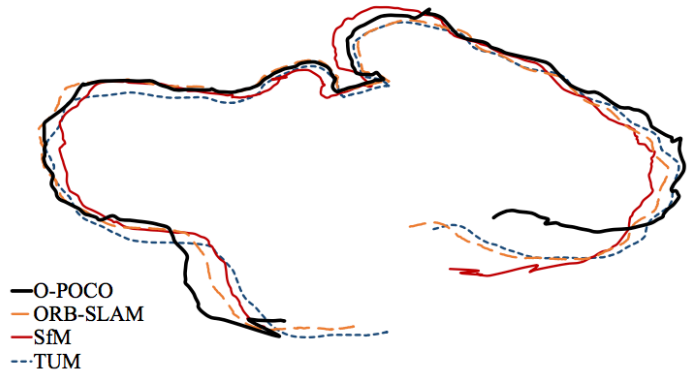


Fig. 10: SfM and ORB-SLAM vs O-POCO in the TUM [14] long office sequence. We see some drift introduced in SfM and O-POCO due to the standard mapping process.

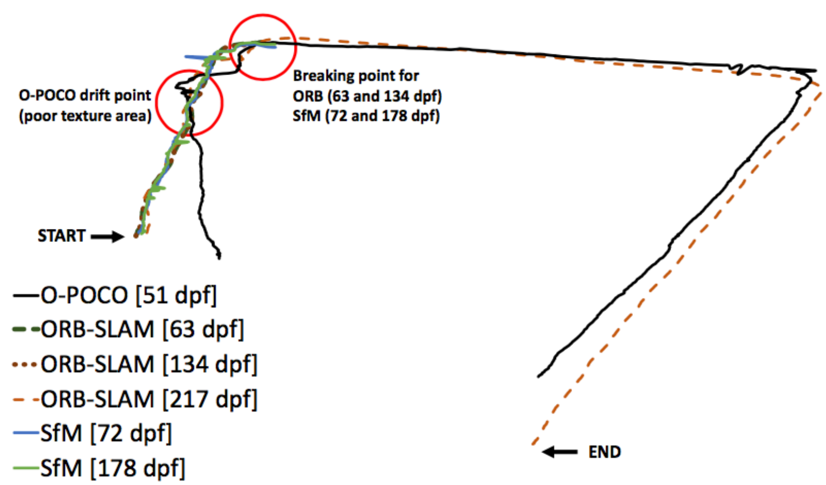


Fig. 11: SfM and ORB-SLAM vs O-POCO. Long corridor test II. Different ORB-SLAM configurations were tested, where only offline ORB-SLAM seems to performs well; our online algorithm suffers from some drift in the poor textured areas but keeps working – we believe that an optimization process will help to correct some of this drift.

Experiments and Results

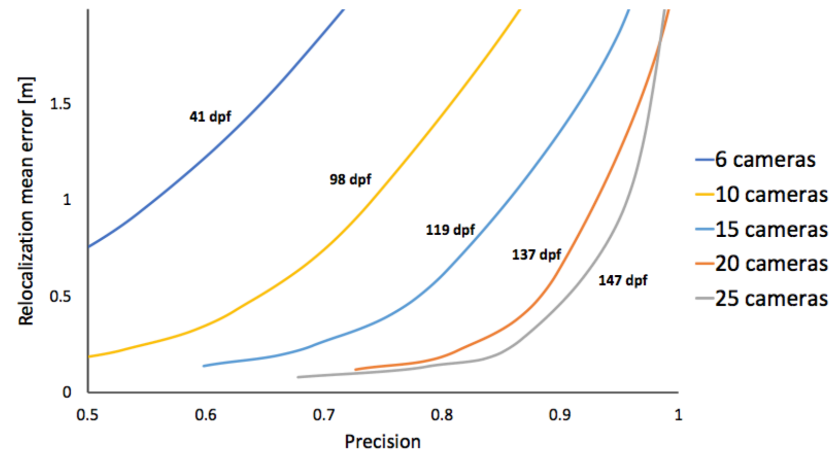


Fig. 13: O-POCO's relocalisation and compression performance versus precision rate – i.e. proportion of correctly relocated cameras – at different sub-map window sizes.

Conclusion

- traditional SfM and ICP
- low relocalisation mean error: 20cm
- outperming than ?? in case of four time less information (in terms of what?)