Report: ClusterSLAM

Zishuo Zhao (with Sheng Yang, Jiahui Huang and Shi-Min Hu)

Abstract

Simultaneous Localization and Mapping(SLAM) is a technology to construct the 3D scene via observations of a moving agent in it, and current methods of SLAM utilizes extracted feature points (as landmarks) to make best use of geometric tools, currently outperforming Deep Learning methods on raw video data. However, traditional algorithms for SLAM mostly works for static scenes, while in practice, there are almost always moving objects in a scene. Currently, most papers on SLAM in dynamic scenes are based on the fact that most landmarks are static, thus use robust kernels to rule out dynamic objects and do SLAM based on filtered static objects. [1]

In this work, we presented with an innovative perspective to utilize and process dynamic objects by the *clustering* paradigm. By using geometric properties of the rigid body that pointwise distances within it keep constant, we proposed an alternating algorithm to partition landmarks into rigid clusters from 3D position data from stereo camera observations, and further optimize the positions based on the clustering information. In this paper, we reached near-perfect accuracy for indoor scenes with our Cluster-Optimize alternating algorithm, and discussed about the theoretical limitation of detection of rigid bodies in outdoor scenes with the stereo camera.

1 Introduction

1.1 Background

Perceiving and modeling the surrounding environment is the foundation of navigating modern autonomous things, which is achieved through a simultaneous estimation of the robot state and construction of the map through on-board sensors. With the booming demand of service robots and self-driving cars, Simultaneous Localization and Mapping (SLAM) technology is currently facing more challenging scenarios, e.g., limited sensing devices and computing resources to be operated in cluttered and dynamic scenes.

The current *de-facto* standard formulation of SLAM approaches, is based on the Maximum-A-Posteriori (MAP) estimation: Variables including the trajectory of the robot and the position of landmarks, are solved under the constraint of noisy measurements observed by sensors, as a factor graph optimization. Such a framework can effectively handle static scenes, or a certain amount of dynamic scenes with additional numerical approaches to alleviate or eliminate their interference. [1] While essentially, dynamic components in the scene are the cause of the drifting of these landmarks. To achieve a deeper perception of the environment, we not only should robustly estimate the robot motion, but also require to parse the movement of these dynamic components.

Recent advanced SLAM systems that address dynamic environments are mainly based on two categories to detect these components: semantic information and motion consistency. For the first category, semantic information is extracted from frames through deep learning technologies, which segment objects in these frames into *moveable* and *static* parts. Although *moveable* is not equivalent to *moving*, such a priori knowledge facilitates the partition of scene components. [3] For the latter category, conflicts caused by relative motions are used to separate moving objects, which can be detected through visual differences, scene flow, or non-rigid deformation between these scene components. [2]

In this paper, we revisit the standard factor graph optimization, and extend it for handling dynamic environments, i.e., partition dynamic components and parse their motions. Specifically, we present a clustering approach for grouping landmarks that share a consistent rigid 3D motion in a temporally sliding window, and use an iterative two-stage manner for simultaneously refining the discrete partition and these estimated 3D motion(as shown in Figure 1).

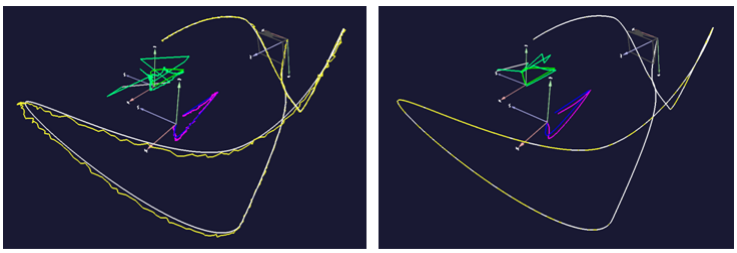


Figure 1: Overview: detection of motion groups and refinement of trajectories

We perform experiments on both publicly available real-world scans and high-fidelity simulated scans to verify our approach quantitatively. Besides, we also present derived applications to demonstrate the capability of our method in comparison to previous works.

1.2 Basic setting

The ground truth is a dynamic scene with some rigid static or moving objects, with a moving stereo camera in it, and features of those objects are labeled in the 3D space. In each frame, the camera takes two stereo photos with coordinates of visible landmarks given as input with some noise, of which the same ID corresponds to the same landmark in all frames. From those coordinates, we desire to partition landmarks into clusters, each expected to be a rigid body.

2 Target and My Contribution

The target for the main algorithm is to partition landmarks into different clusters in which each is a rigid body, by which different objects are detected by relative motion, and then find the trace of each object, by a video of stereo camera observations.

If the observations are absolutely accurate, then by the stereo observation of each frame, we can accurately decide the spatial position of each landmark relative to the camera. Then if we can detect rigid bodies from the landmarks and cluster them accordingly, we can track the motion of each object. However, due to errors and noises of the observations, the 3D coordinates of landmarks are not accurate and need some denoising. Therefore, we do the two phases alternatingly to get a satisfiable result:

- Clustering: to divide landmarks to multiple motion groups which are recognized as rigid bodies with some noise tolerance.
- Optimization: to optimize the positions of landmarks based on properties of real rigid bodies.

In this project, my contributions are mainly in:

- Designed a geometric model that reflects the properties of rigid bodies based on pairwise distances;
- Designed the minimax-cluster algorithm that partitions landmarks into different rigid groups, which is theoretically guaranteed to work for noise small enough;
- Improved its time efficiency and robustness of minimax-cluster algorithm by choosing representative landmarks on practical data.

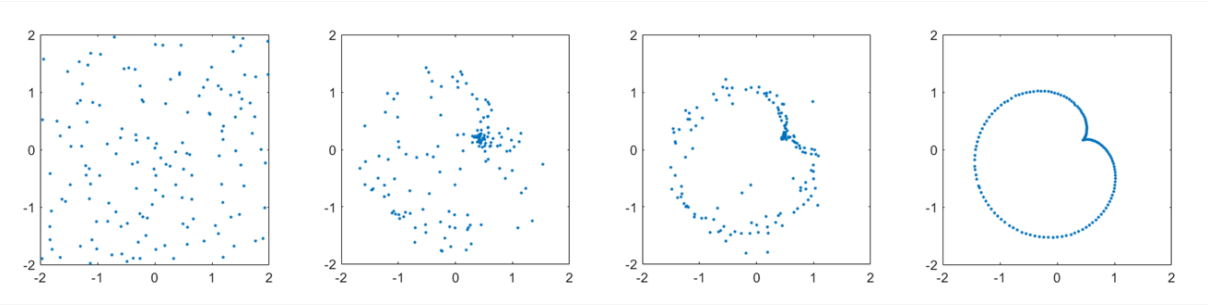


Figure 2: Reconstruction of the cardioid shape based on pointwise distance information

3 Method

3.1 Criterion for rigid bodies

3.1.1 The idea of pairwise distances

To properly partition landmarks into rigid bodies, we must first find some identities of rigid bodies which can translate and rotate but cannot deform. Rather than identify the complicated motions of an object which has not even been detected, I utilized a simple but effective criterion for a rigid body: the scalar distance between each pair of points on it is constant regardless of all kinds of rigid motions.

On the other hand, for two different rigid bodies, as long as there is relative motion between them, the distance between some corresponding points must vary in time (otherwise we can add imaginary rigid "sticks" between each pair of points on them and regard them as a whole rigid body). However, the distance between certain pair of corresponding points can remain the same: for example, the hour hand and minute hand are not a rigid body, but the end of one and the base of the other has constant distance. Therefore, we must be careful in the clustering algorithm, as the *constant-distance* property is NOT transitive, thus we must guarantee that "more than a spanning tree of" pairwise distances are constant for the inference of a rigid body.

3.1.2 Can pairwise distances decide a rigid body?

It is straightforward to see that if a set of points form a rigid body, their pairwise distances are fixed even after rigid motions, but can fixed pairwise distances uniquely decide a rigid body? For this project, we only need a weaker property, as motions between neighboring frames in a video is expected to be small:

In small non-rigid deformations, some pairwise distance between pairs of corresponding points will not remain constant.

Rather than a rigorous mathematical proof, here we can understand it in a physical way: if an object is deformed, then some part of it must have been stretched or compressed, resulting in changes of pairwise distances.

Here is also an experiment that shows that, given pairwise distances between a discrete set of points, we can recover the structure of them by optimizing their positions by gradient descent with the loss as the square-sum of pairwise distance errors:

3.2 The Minimax-Cluster algorithm

3.2.1 The perfect-observation case

In the model, we assume that all landmarks are perfectly aligned: the landmark with the same ID (called as "the same" landmark in following texts) corresponds to the same point on the same object in each frame, and every landmark appears in most, if not all, frames of the video. For any pair (l_i, l_j) of landmarks, we define $D(l_i, l_j)$ the *dissimilarity* of them, as the variance of their distances in all frames they both appear. If every

landmark appears in at least $(1-h)$ fraction of all frames, then l_i, l_j appear in at least $(1-2h)$ fraction of all frames, making the variance of their distance a valid measure to judge if there is rigid connection between them for small h.

Then, we do the clustering by the following algorithm, with a threshold λ as the tolerance of a dissimilarity within a cluster, which is roughly the allowed noise of pairwise distances:

1. Assign each point into an isolated cluster c_i, and define the dissimilarity $d(c_i, c_j) = D(l_i, l_j)$.
2. Find the least $d_* = d(c_{i_*}, c_{j_*})$ s.t. $i_* \neq j_*$.
3. If $d_* > \lambda$, output all existing clusters and the algorithm ends.
4. Merge clusters c_i and c_j into a new cluster c_k distinct from all existing ones.
5. For all other existing c_l, let $d(c_l, c_k) = \max\{d(c_l, c_i), d(c_l, c_j)\}$.
6. Delete c_i and c_j and goto 2.

Notice that if we change the max in the Step 5 into min, the algorithm becomes the Kruskal algorithm, which yields the minimum number of clusters with all inter-cluster pairwise dissimilarity greater than λ, i.e.

$$\min_{c_i \neq c_j} \min_{l_s \in c_i, l_t \in c_j} d(l_s, l_t) > \lambda. \tag{1}$$

However, this might lead to under-segmentation as argued in 3.1.1. Actually, the condition that two clusters cannot be merged should be:

$$\max_{l_s \in c_i, l_t \in c_j} d(l_s, l_t) > \lambda,$$

therefore, as our algorithm yields a *local* minimum number of clusters with

$$\min_{c_i \neq c_j} \max_{l_s \in c_i, l_t \in c_j} d(l_s, l_t) > \lambda, \tag{2}$$

for input data accurate enough, it yields a desired clustering for this problem.

Time complexity analysis

For totally n points, there are $O(n)$ iterations, and in each iteration, if we use the brute-force search for Step 2, Step 2 takes $O(n^2)$ time, Step 4 takes $O(\alpha(n))$ time and Step 5 takes $O(n)$ time, so the total time complexity is $O(n^3)$.

If we use the heap structure to store the matrix of dissimilarity while maintaining the minimum value, then Step 2 takes $O(1)$ time, Step 4 takes $O(\alpha(n))$ time and Step 5 takes $O(n \log n)$ time, so the total time complexity is $O(n^2 \log n)$.

We can also use a sparse matrix with m entries rather than a dense matrix, in which setting the min and max are computed on applicable elements. In this case, the time complexity is $O(m \log n)$.

3.2.2 The idea of representative landmarks

The $O(n^2 \log n)$ time complexity shows that the running time grows rapidly with the number of landmarks, because we must deal with all $\Theta(n^2)$ pairwise distances. However, for a rigid body in the scene, if there is not noise, it is sufficient to describe its motion by tracking only 4 landmarks on it. It suggest that: to detect all rigid motion groups, we do not actually need all n landmarks, so tracking only some of them may be sufficient. After deciding on local coordinate systems for all motion groups with the ICP algorithm, we can assign other points into the group in which they are the most static.

The decision on the representative points can be technical, but there is naturally a criterion for practical scenes. As some landmarks can be obscured or out of sight, the distance information concerning them can be

less sufficient than those appear more constantly in the video. Therefore, we can choose points with relatively more occurence in the video, and their information can be more accurate than others.

After clustering those representative points, we use ICP algorithm to generate the local coordinate system for each cluster (which is Jiahui's contribution), and then assign each other landmark into the cluster in which it moves the least with variance of position. Then, we finish with the clustering of all landmarks.

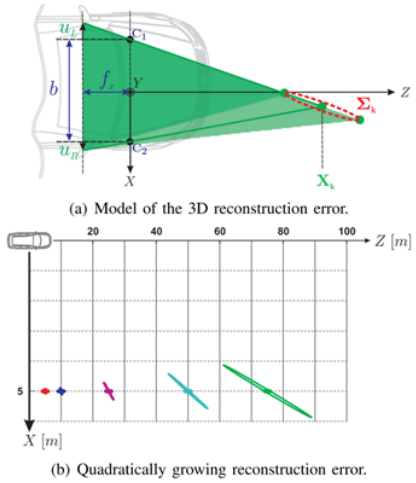
3.3 Theoretical limitations

3.3.1 Robustness against noise

The clustering algorithm is based on the paradigm of computing the 3D position for landmarks in stereo observations. In this way, the depth information is mainly contained in the parallex, i.e. the difference in the observation in two lenses. Intuitively, the parallex gets smaller for objects farther away, so for fixed noise level in observation, the error in estimated position grows rapidly with depth. Therefore, it expects to work better for indoor scenes than outdoor scenes.

For a stereo camera with baseline length l, the parallex for an object with depth d is approximately

$$p \approx \frac{l}{d}. \tag{3}$$



(a) Model of the 3D reconstruction error.

(b) Quadratically growing reconstruction error.

Figure 3: Rapidly growing triangulation error with depth

Therefore, if the observed parallex has uncertainty δp, the uncertainty of d is approximately

$$\begin{aligned} \delta d &= \delta\left(\frac{l}{p}\right) \\ &= \left|\frac{\partial}{\partial p}\frac{l}{p}\right|\delta p \\ &= \frac{l}{p^2}\delta p \\ &\approx \frac{d^2}{l}\delta p. \end{aligned} \tag{4}$$

Then, if δd is greater than the distance of relative motions between objects, then it is difficult to distinguish if two landmarks are rigidly connected or not, as we cannot decide whether the observed relative motion is due

to actual relative motion or observation noise by its scale. Therefore, (4) suggests an information theoretical limitation to the accuracy of stereo-camera observations.

Particularly, for the driving case (in next section) with 1080×720 resolution and $FOV = 90°$, $l = 0.5m$, if the uncertainty of both observations is 1 pixel, then the uncertainty of the parallex is 2 pixels, thus for $d = 30m$, the uncertainty $\delta d \approx 7m$, making it hard to do the task.

3.3.2 Consistence of the dissimilarity measure

The dissimilarity of two landmarks is defined as the variance of their distance in all frames they appear simultaneously. It can be expected that if they both appear in most frames in the video, the computed value will not deviate from the ground truth. However, if the number of samples is small, then the dissimilarity measure can be defective.

For example, if the landmarks A, B satisfy $\|A - B\| = 2 + 0.1t$, then for two sample sets $t \in \{0, 1, 2, 3, 4\}$ and $t \in \{0, 10, 20, 30, 40\}$, although they both contain 5 frames, the computed dissimilarities of A and B are significantly different.

This issue appears more in outdoor scenes than indoor scenes too. One solution can be modify the dissimilarity by a factor determined by the distribution of co-appearance frames, but it may not be perfect either.

4 Evaluation

As the project has not been finished, the heap-structure speedup is not applied in the experiment yet. Therefore, the time complexity of the clustering algorithm is $O(n^3)$, in which n is the number of all landmarks or representative landmarks.

4.1 Benchmarks

4.1.1 Indoor case

For the Two-Chair scene without representative landmark technique, the average clustering time is 4s, and the algorithm converges to 98.3% accuracy after 2 iterations.

For the Two-Chair scene with representative landmark technique, even for larger noise (Q means rounding), the average clustering time is reduced to 0.7s and it converges to 100% accuracy on representative landmarks and 99.5% accuracy on all landmarks after 2 iterations, improved in both time efficiency and accuracy.

4.1.2 Outdoor case

For the Driving scene without representative landmark technique, the average clustering time is 25s. The initial accuracy is 61%, but the optimization decreased it to 57%.

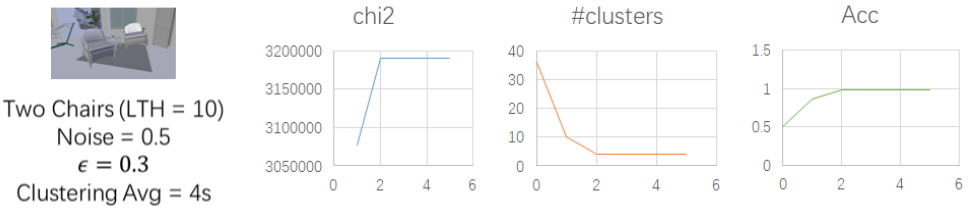


Figure 4: Experiment result: indoor, without representative landmarks

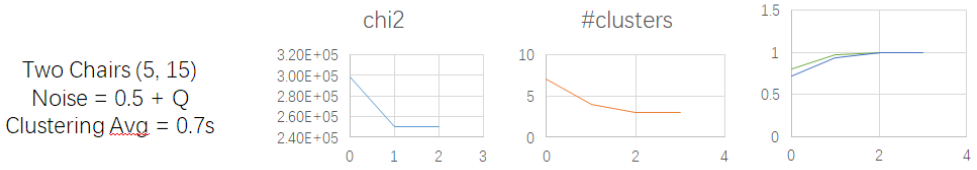


Figure 5: Experiment result: indoor, with representative landmarks

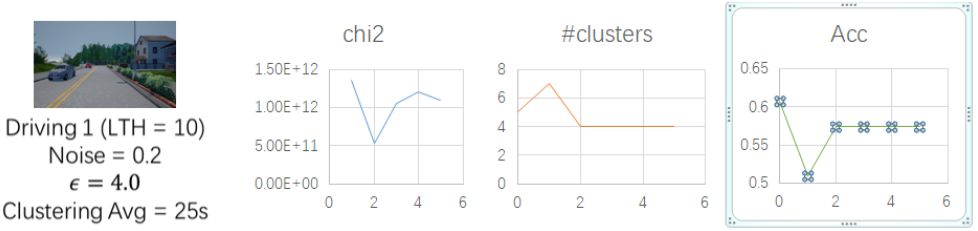


Figure 6: Experiment result: outdoor, without representative landmarks

For the Driving scene with representative landmark technique, the average clustering time is 14s, and the algorithm converges after 2 iterations to 71% accuracy for representative landmarks and 67% for all landmarks. Although the accuracy is not very high due to issues in Section 3.3.1, the representative landmark technique does improve both the efficiency and accuracy.

4.2 Comparison with existing methods

Versus RANSAC

In the RANSAC algorithm, each time we randomly draw 3 points and assume they are in the same rigid body, thus construct a local coordinate based on the 3 points. If they are truly from the same rigid body, then other points from the same rigid body will probably be nearly static in that local coordinate system.

However, as only 3 points are drawn for a coordinate system, it will be subject to noise more than Minimax Cluster algorithm, and it is hard to decide on the threshold for "nearly static" as it has no direct geometric interpretation. Additionally, even if we draw 3 points not from one rigid body, some points may still be "static" in the coordinate system, adding to uncertainty to the result.

Versus Affinity Propagation

Affinity Propagation is a clustering algorithm that applies to non-metric dissimilarity matrices, which has $O(kn^2)$ $(k \ll n)$ time complexity, better than brute-force Minimax Cluster algorithm and comparable to heap-structured Minimax Cluster. However, both the algorithm itself and its parameters has no geometric interpretation either, therefore even for absolutely accurate input, it does not guarantee to converge in the right clustering, while our Minimax Cluster algorithm guarantees it.

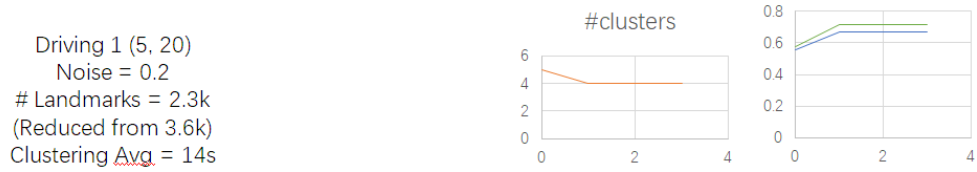


Figure 7: Experiment result: outdoor, with representative landmarks

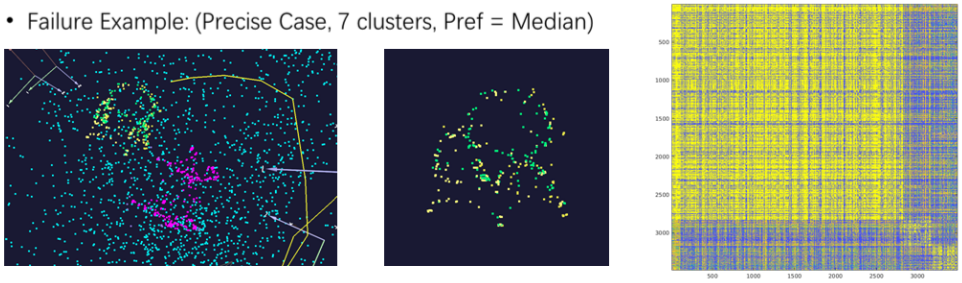


Figure 8: Affinity Propagation on Two-Chair scene: not accurate even for precise input.

5 Acknowledgements

Additional acknowledgement for Shiyin Wang (in IIIS 60) for discussion about the idea of the clustering algorithm and a wide range of additional topics.

Additional acknowledgement for Junxiong Cai (in Tsinghua CG Lab) for discussion about the heap structure for speedup of the Minimax-Cluster algorithm.

References

[1] Matthew C. Graham, Jonathan P. How, and Donald E. Gustafson. Robust incremental slam with consistency-checking. In *IEEE/RSJ International Conference on Intelligent Robots Systems*, 2015.

[2] P. Lenz, J. Ziegler, A. Geiger, and M. Roser. Sparse scene flow segmentation for moving object detection in urban environments. In *Intelligent Vehicles Symposium*, 2011.

[3] Liang Zhang, Leqi Wei, Peiyi Shen, Wei Wei, and Juan Song. Semantic slam based on object detection and improved octomap. *IEEE Access*, PP(99):1–1, 2018.