Semantic Histogram Based Graph Matching for Real-Time Multi-Robot Global Localization in Large Scale Environment

Xiyue Guo Junjie Hu Junfeng Chen Fuqin Deng Tin Lun Lam

IEEE ROBOTICS AND AUTOMATION LETTERS (RA-L), 2021

https://github.com/gxytcrc/semantic-histogram-based-global-localization

Feb 24th, 2023

Outline

1. Introduction
1.1. Challenges
1.2. Contribution
2. Related Works
2.1. Appearance-Based Methods
2.2. Graph-Based Methods
3. Methodology
3.1. Graph Extraction
3.2. Semantic Histogram Based Descriptor
3.3. Graph Matching
3.4. Pose Estimation
4. Experiments
4.1. Performance Comparison
4.2. Global Localization for Multi-Robots
4.3. Generability on Real-World Scenarios

Challenges MR-SLAM is Facing

- **Global positioning accuracy problems caused by different viewpoints of each robot:** As shown in Fig. 1, there is a great difference in perspective between images shot by vehicles and drones;

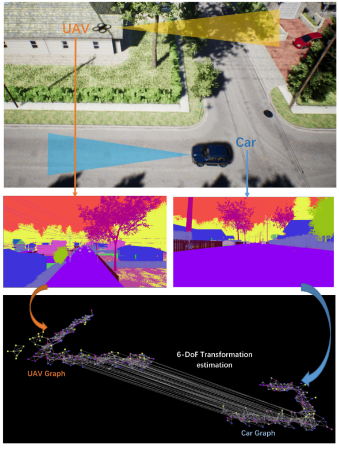


Fig. 1. An example of our semantic based graph matching method. The method is used for global localization in a large scale environment. The viewpoint between two robots (UAV and Car) is extremely large. We utilize semantic maps to build semantic graphs for two robots. Then, the transformation matrix between them can be simply estimated.

Challenges MR-SLAM is Facing

- **Global map must be calculated efficiently:** Traditional appearing-based methods use BoW, but for large perspective differences, local image features change significantly, resulting in failure of such methods.

Semantic-based method builds semantic-based graphs for different viewpoints, and then uses semantic information to match graphs, which has better performance when the viewpoint changes greatly.

In large-scale environment, only nodes' semantic labels used for graph matching will lead to mismatching. Therefore, for each node:

- a descriptor containing the surrounding information should be extracted;
- no matter how large the graph is, graph matching needs to be processed in real time.

Contributons of Paper

In this paper, we use semantic-based graph matching to propose a descriptor based on semantic histogram to achieve real-time matching under large viewpoint changes. (Figure -2 shows a diagram of the descriptor.) Based on the new descriptor, a global localization system based on semantic graph is further developed.

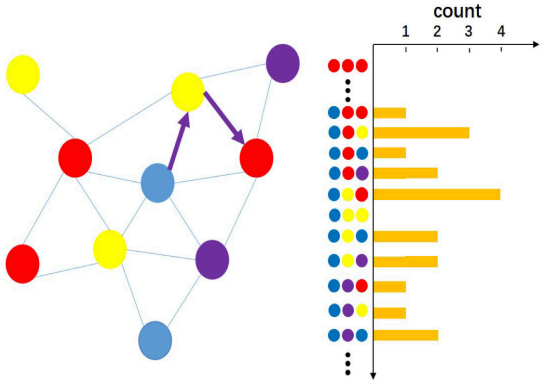


Fig. 2. An illustration of the semantic histogram based descriptor. Left is the semantic graph. The searched path is started from the start point (blue). The path information is recorded as a prearranged histogram on the right. The similarity score between two descriptors can be obtained through the normalized dot-product.

Outline

Appearance-Based Methods

- Using BoW to find associations of images, such as FAB-MAP. But when the viewpoint difference is large, the systems become less reliable.
- Overcome viewpoint change problem with landmarks created CNN, but the landmarks are not reliable when viewpoint changes significantly.
- Some methods, such as LoST, use semantic and appearance information to recognize the places in opposite viewpoint, but they do not consider localization.

Graph-Based Methods

Graph-based methods formulate the global localization prob- lem as a graph registration problem. The associations between different graphs are found by extracting the correspondences between nodes across the graphs. Then, the relative pose between graphs can be calculated.

- In some traditional methods, each node is labeled by local features based visual word, but these features are not reliable when viewpoint change significantly.
- Some recent methods to create labels, but BF(Brute Force) or Hungarian is used for matching, which is not suitable for complex and large environment.
- Using Random Walk to generate descriptor for each semantic node, but it will be high high computational complexity when matching graphs are large.

Outline

Summary

The framework proposed is inspired by X-view[1]:

1. Given two odometries, related depth maps, and semantic maps.
2. Generate semantic graphs first, and then extract the semantic histogram based descriptors.
3. Two graphs are matched with the extracted descriptors.
4. Calculate 6-DoF transformation matrix.

[1] *A. Gawel, C. D. Don, R. Siegwart, J. Nieto, and C. Cadena, "X-view: Graph-based semantic multi-view localization," IEEE Robot. Automat. Lett., vol. 3, no. 3, pp. 1687–1694, Jul. 2018.*

Diagram of the System

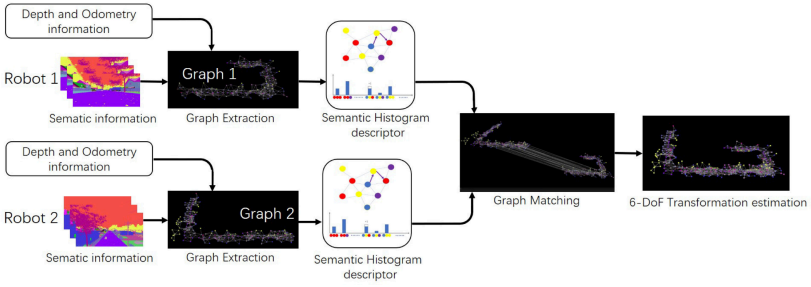


Fig. 3. The diagram of our global localization system. The system takes semantic maps, depth maps, and odometries as inputs. The 3D semantic graph for each robot is first built from the inputs. Then, the descriptor of each node is extracted with the semantic histogram method. Next, the two graphs are matched by comparing the descriptors across the graphs. Finally, the matched correspondences are used to estimate the 6-DoF transformation between the coordinate systems of two robots.

Graph Extraction

To build the graph, we need to extract nodes from images:

- Employ the seed filling[2] method to segment objects frome images.
- Using 3D coordinates to avoid failed segmentation between two neighboring objects with the same semantics during the segmentation process.
- The 3D geometry center of each object is extracted as a node (A node contains 3D coordinate and semantic label).
- Nodes with the same semantic label are merged if they are close to each other.
- Undirected edges are formed if the distances between nodes are smaller than threshold.

[2] *M. C. Codrea and O. S. Nevalainen, "Note: Analgorithm for contour-based region filling," Comput. Graph., vol. 29, no. 3, pp. 441–450, 2005.*

Histogram Based Descriptor

The surrounding information of the node needs to be recorded by extracting the node's descriptor. Histogram based descriptors are simple, and the matching procedure is very fast.

The simplest descriptor is Neighbor Vector[3], but its matching performance is low due to the lack of topology information (information of neighbors).

[3] *E. Stumm, C. Mei, S. Lacroix, J. Nieto, M. Hutter, and R. Siegwart, "Robust visual place recognition with graph kernels," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4535–4544.*

Descriptor Proposed by Author

Goal: Include more neighbors information for each node.

- For each node, descriptor stores all possible paths started from it.
- Set the length of the path as 3, each path records three steps' semantic labels.
- The time complexity of one descriptor extraction is $O(MN)$, where M and N are the numbers of first-order and second-order neighbors.

Algorithm 1: Descriptor Extraction.

Input: G: Semantic Graph;
Output: V: Histogram of path descriptors for G;

for i-th node in G **do**
  Initialize the histogram vector V_i;
  Record the node's label l_i;
  for m in neighbor nodes of i **do**
    Record the first neighbor node's label l_m;
    forn in neighbor nodes of m **do**
      Record the second neighbor node's label l_n;
      The Histogram cell $V_i(l_i\text{-}l_m\text{-}l_n)$ plus one;
    end for
  end for
  Add V_i into V;
end for

Matching Score

In the matching process, only the nodes that have the same labels will be compared. The similarity score is:

$$\text{Score}(A, B) = \frac{\sum_{d=1}^{n_d} A_d \times B_d}{\sqrt{\sum_{d=1}^{n_d} (A_d)^2} \times \sqrt{\sum_{d=1}^{n_d} (B_d)^2}} \tag{1}$$

A and B denote descriptors of nodes from two graphs; n_d is descriptor dimension, which is equal to the cubic of the lable number n_l; the size of n_d is typically on the order of hundreds.

- The similarity score between the two nodes is between 0 and 1, the higher score means higher similarity.
- The correspondences whose similarity scores are higher than the threshold T_s are stored as the matching candidates.

Matching Algorithm

Transformation between two graphs is rigid, and transformation values between the correct pairs of correspondences should be similar.

- ICP-RANSAC is used to reject the outliers.
- Remained inlier correspondences are kept for pose estimation.

Algorithm 2: Graph Matching and ICP-RANSAC Rejection.

Input: V, V': descriptor sets of two graphs; N_i: iteration number for RANSAC; M_0: initial matches set;
Output: M_1: final matches set;
Initialize M_0;
for i in V **do**
  for j in V' **do**
    scores = Score(V_i, V'_j);
    if scores > score threshold T_s; **then**
      Add the Correspondence C_{ij} to M_0;
    end if
  end for
end for
Initialize M_1;
Initialize the Maximum Inlier number A^*;
let $A^* = 0$;
for o = 1 to N_i **do**
  Select 4 correspondences M_{four} Randomly;
  R_o, t_o = ICP(M_{four});
  for k in Matches set M_0 **do**
    Obtain the correspondence C_k;
    Error = Evaluation(C_k, R_o, t_o);
    if Error< Threshold T_R **then**
      Add C_k to the Inlier set M_o;
    end if
  end for
  Inlier number A = Count(M_o);
  if $A > A^*$ **then**
    $M_1 = M_o$;
    $A^* = A$;
  end if
end for

Pose Estimation

The final transformation matrix is computed with ICP algorithm. The inlier correspondences obtained by RANSAC method are used for registration. Rotation matrix R and translation vector t is obtained by minimizing the sum of squared error:

$$E(R,t) = \frac{1}{N_p} \sum_{k=1}^{N_p} W_k \| q_k - R p_k - t \|^2 \tag{2}$$

N_p is the correspondences number after RANSAC rejection; p_k and q_k are the correspondent nodes in two graphs; W_k is the weight.

Outline

1. 1. Introduction
1.1. Challenges
1.2. Contribution
2. 2. Related Works
2.1. Appearance-Based Methods
2.2. Graph-Based Methods
3. 3. Methodology
3.1. Graph Extraction
3.2. Semantic Histogram Based Descriptor
3.3. Graph Matching
3.4. Pose Estimation
4. 4. Experiments
4.1. Performance Comparison
4.2. Global Localization for Multi-Robots
4.3. Generability on Real-World Scenarios

Experiment Summary

Dataset: SYNTHIA, KITTI(only use RGB images)
Hardware: Intel Core i7-8565 U @ 1.80 GHz

- The method in this paper is stable and accurate for large viewpoint differences in a large-scale environment;
- Much faster than state of the art semantic-based methods [1]. This method has good performance for both homogeneous and heterogeneous robot systems.
- On the KITTI dataset, the DNN is used to predict the depth map and semantic map, which proves the good performance of the proposed method in map fusion.

Data in SYNTHIA

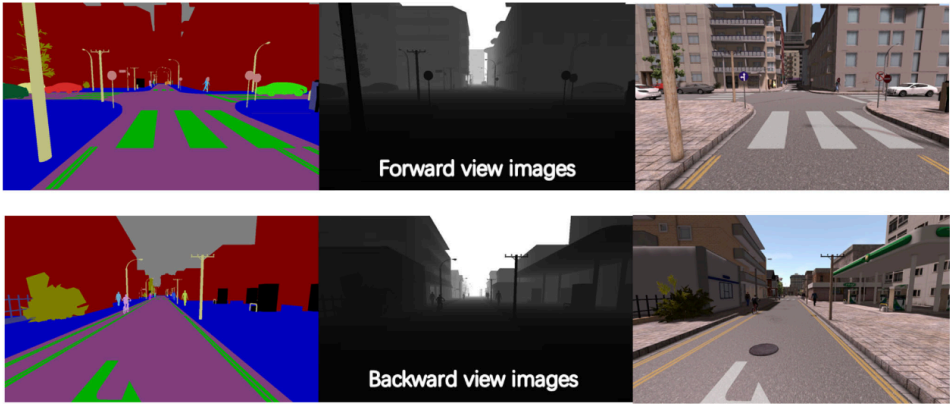


Fig. 4. Samples of SYNTHIA dataset. images in top row are the forward view images, including semantics, depths, and RGB images. The images in bottom row are the backward view images collected at the same time.

Experimental Result on SYNTHIA (Single Robot)

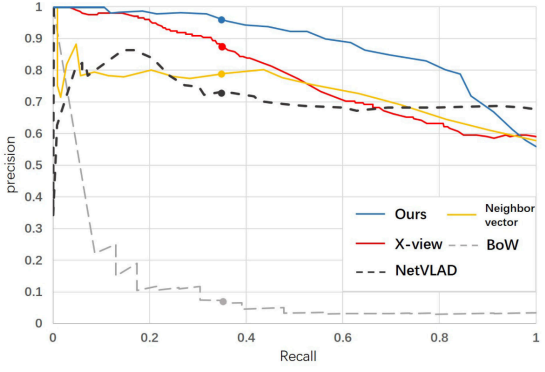


Fig. 5. Precision-Recall curve of different global localization methods. The operation points are shown as dots.

TABLE I
TIME-CONSUMING FOR EACH COMPONENT OF OUR METHOD

Module	Time(ms/frame)
Graph Extraction	114.23 ± 4.53
Descriptor Extraction	0.68 ± 0.03
Graph Matching	1.65 ± 0.49
Pose Estimation	0.63 ± 0.14
Total	117.19 ± 5.19

Collect data from the Neighborhood of AirSim.
Dataset: https://drive.google.com/file/d/106sPA48vFThLK0RB4WBcj-i8FZPQPmcV/view

Fig. 6. The illustration of three simulated trajectories generated from AirSim. We use them to evaluate the performance of our global localization method for both homogeneous and heterogeneous robot systems.

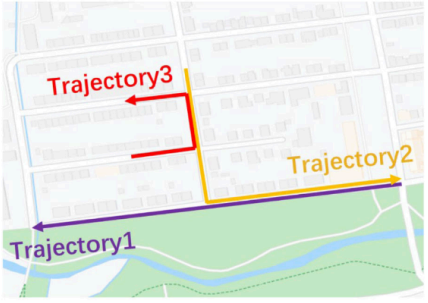
Experimental Result (Multi-Robots)

- Due to the less surrounding information, the matching performance of the Neighbor Vector is the worst.
- Time complexity: Random Walk $\gg$ Paper's $>$ Neighbor Vector

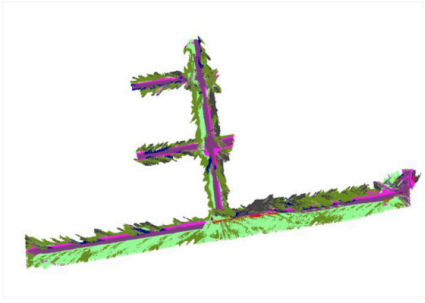
TABLE II
THE QUANTITATIVE COMPARISONS OF DIFFERENT DESCRIPTORS FOR GLOBAL LOCALIZATION OF MULTI-ROBOT SYSTEMS ON AIRSIM

Robot Type	Matching Graph Size	Descriptor Type	Matching Time (sec)	Good Matches	Good Matches Rate(%)	Processing Time (sec)	Translation Error (m)	Rotation Error (degree)
Car1 and Car2	317 points and 328 points	Random Walk	4.155	125	40.0	4.304	3.44 ± 1.39	0.92 ± 0.49
		Neighbor Vector	**0.013**	120	37.8	**0.057**	4.55 ± 1.32	0.48 ± 0.31
		Ours	0.132	**152**	**49.1**	0.184	$\mathbf{3.12 \pm 0.76}$	$\mathbf{0.30 \pm 0.29}$
Car1 and UAV	317 points and 486 points	Random Walk	6.637	136	43.6	6.859	2.23 ± 0.88	2.62 ± 0.46
		Neighbor Vector	**0.021**	100	31.7	**0.089**	3.23 ± 1.02	3.25 ± 0.54
		Ours	0.195	**142**	**45.1**	0.248	$\mathbf{2.12 \pm 0.47}$	$\mathbf{2.61 \pm 0.25}$

Mapping Visulization on KITTI



(a) The trajectories of KITTI 08 dataset



(b) The successful multi-robots map fusion

Fig. 7. Trajectories and reconstructed maps of sequence 08 from KITTI dataset. (a) shows the trajectories where each line denotes a trajectory. (b) shows the maps reconstructed and merged from these three trajectories.

Mapping Error on KITTI

TABLE III
THE TRANSLATION ERROR OF GLOBAL LOCALIZATION ON THE KITTI DATASET (IN METERS)

	Sequence 02	Sequence 08A	Sequence 08B	Sequence 19
Neighbor Vector	14.42±20.02	4.59±0.63	18.42±4.00	15.18±11.45
Random Walk	76.61±36.42	4.83±0.68	25.55±8.72	14.63±13.35
BoW	55.20 ± 42.01	74.12 ± 51.14	32.16 ± 20.79	108.83 ± 54.05
NetVLAD [17]	28.21 ± 19.35	35.02±21.04	24.52±14.41	55.11±20.96
Ours	**8.77±11.39**	**4.42±0.35**	**7.48±3.67**	**8.10±6.63**

Thank you!