Object detection and tracking aided SLAM in image sequences for dynamic environment.

Catherine Waithera Wangari
Applied Informatics Major, Graduate School of Science and Engineering,
Hosei University,
Supervisor: Prof. Jinjia Zhou.

Abstract— **Object detection in a dynamic environment is important for accurate tracking and mapping in Simultaneous Localization and Mapping (SLAM). Dynamic feature points from people or vehicles are the main cause of unreliable SLAM performance. Previous researchers have used varied techniques to solve this problem, such as semantic segmentation, optical flow, and moving consistency check algorithm. In this proposal, Object Detection and Tracking SLAM (ODTS), we define a weighted grid-based attention model for a feature tracking module to track landmarks and objects. ODTS system tracks landmarks, such as buildings in the background, and objects, such as vehicles, in the foreground. For optimizing performance, a robust self-attention module is integrated. For evaluation, the trajectory of the robot is tracked, and the root mean square error (RMSE) is recorded. Additionally, the number of background and foreground feature points were observed for landmarks and objects. ODTS significantly minimizes the tracking lost problem and produces more accurate maps and tracking of feature points.**
Keywords: Keywords: SLAM, weighted grid-based, feature points, objects, landmarks.

I. INTRODUCTION

SLAM performs two major tasks concurrently: localization and mapping. Previous research have described the problems affecting SLAM performance and the impact of tracking and mapping problems in SLAM includes, surveys by Cadena and Taketomi [1], [2]. Therefore, we propose ODTS SLAM system that minimises the problems of tracking and mapping in SLAM.

Previous work on SLAM in dynamic environments include: DS-SLAM [3] that uses semantic segmentation and moving consistency check for dynamic object detection and exclusion to solve the tracking and mapping problem. Their framework is robust and able to capture a priori data effectively, but it fails to detect the dynamic object that is not a priori. [4] use RANdon SAmpling Consensus (RANSAC) algorithm in moving consistency check resource-intensive, which requires a lot of iterations to compute an optimal solution and requires problem-specific thresholds to be set.

DynaSLAM system [5] uses semantic segmentation and a multi-view geometry algorithm to detect a priori dynamic objects as well as non a priori dynamic objects in the environment. The disadvantage of DynaSlam is that it does not perform well in very populated areas.

Li *et al.* propose attention SLAM [6], a system that combines visual saliency semantic model and visual SLAM. Their approach is from an Information Theory point, and the attention SLAM reduced the pose estimate error. Dynamic object culling(Doc) SLAM [7] system obtains object culling by using panatopic segmentation, optical flow, moving consistency check, and key point supplement, to prevent the tracking lost problem. The disadvantage of this approach is that it significantly increases the system overhead and computational cost.

Lai et al. [8] system is a robust visual SLAM for dynamic environments. They handle dynamic objects by combining instance segmentation networks, optical tracking, and epipolar constraints to eliminate the influences of dynamic objects.

When performing localization and mapping concurrently, the presence of objects introduces a trade-off between creating accurate maps using landmarks and utilizing important information coming from objects in localization to avoid a collision. Our approach optimizes the input images sequence to improve the mapping and tracking, thereby, increasing the system's robustness and producing more accurate results.

II. METHODS

A. Proposed framework

Our system Fig.1 is inspired by the original Murtado et al. Oriented Fast and Rotated Brief (ORB-SLAM2) framework [9]. ODTS Slam proposes: (1) contrast enhancement and (2) weighted grid-based feature tracking module. These changes improve the accuracy of SLAM for feature points tracking and mapping module.

B. Contrast enhancement

For the pre-processing step Fig.2, RGB images are converted into the *L*a*b** color space and apply the S curve to the Luminous layer (L*) only, keeping the a* and b* layers unchanged. Using the sigmoid function equation 1 according to [10] below and defining the optimal c and b values for our target, the brightness layer, L* is fed to the function to obtain an enhanced image.

$$\alpha L = \frac{1}{1 + e^{(L \times (c - l_{i_j})b + \beta)}} \tag{1}$$

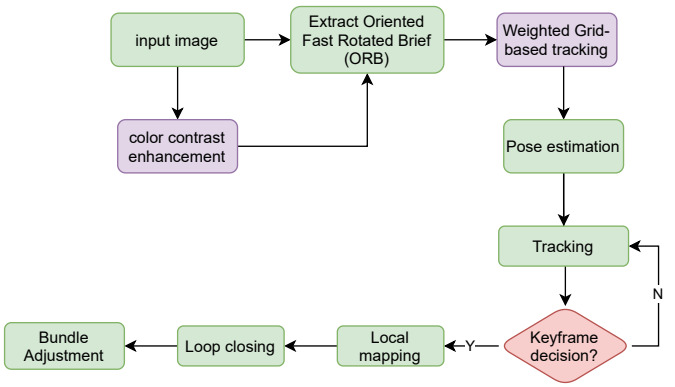

Fig. 1. Object detection and tracking SLAM (ODTS) framework for objects in a dynamic environment.

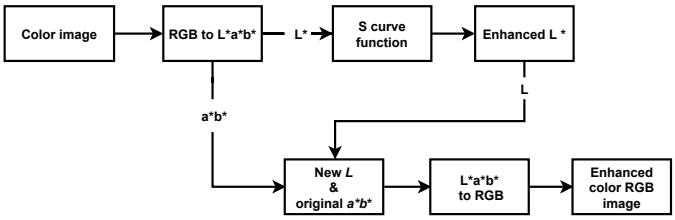

Fig. 2. Color contrast enhancement pipeline.

Where, L represents the Luminous channel that is modified, c and b are constants representing the contrast and brightness values respectively. l_{i_j} is the original image pixel at position x_{i_j} , and β is the constant threshold value. Modifying c and b levels inversely against a set threshold β value improves the image illumination and prevents over-illumination.

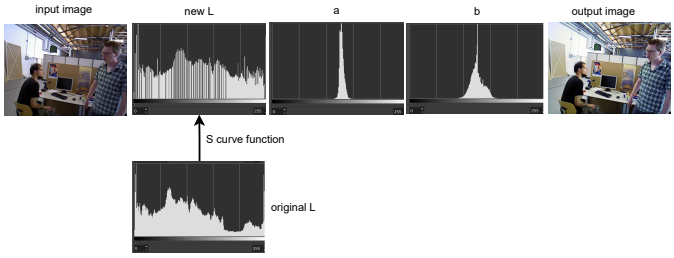

Fig. 3. Qualitative results from color contrast enhancement pipeline.

From Fig.3 the histogram of the original layer L has poor illumination, and the intensity values are concentrated in a small range. After applying the s curve, the new layer L has high contrast, and the corresponding histogram has the intensity levels distributed over range and leveled.

C. Weighted Attention model

We examine a dynamic environment based on the human gaze to simulate which parts of a scene are important to detect the position and orientation of an agent and, which parts are important to detect the pose of an agent in a scene.

Towards achieving true self-awareness, attention as described in Attention SLAM [6] needs to be robust. Our ODTS framework uses self-attention to assign the relevant scores for ORB feature descriptors, and to classify landmarks and objects.

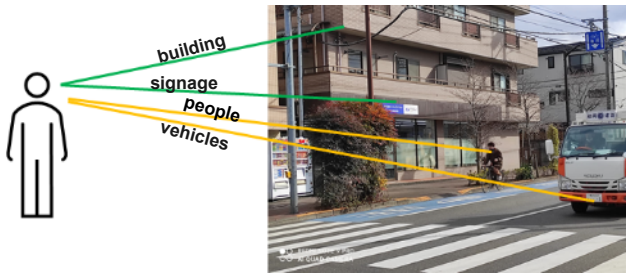

Fig. 4. The green line shows the higher lying features, and the yellow lines show the lower lying features. As the agent moves, the only the lower lying features are likely to be affected by occlusion or removal or new features to be inserted.

We are inspired by the grid system defined in [11], and implement a weighted grid-based system for ODTS framework. Input images from the dataset sequence are defined by a size of H x W, let x represent the input vector for the frames in the image sequence and m represents the number grids, using the Leece lattice approach [12] product quantization, we define the grids as below. The frame is divided into I x J grids, and each of the grids have size h x w. Each of the grids has feature points corresponding to the ORB feature points in the original frame.

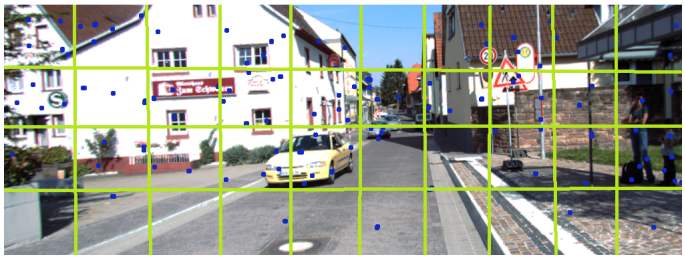

Fig. 5. The grid generated on the image with ORB feature points.

In Fig 5 The feature points are weighted using a grid-based system and feature vectors are generated. We use the sum of the weighted feature vectors for tracking and mapping in the ODTS SLAM system. Feature points for tracking are selected based on their position and the weighted attention scores.

III. RESULTS

Our experiments are performed on a powerful Intel Core i7 using Nvidia GeForce RTX 3060 GPU computer, running Ubuntu 20.04 focal. We ran image sequences from the KITTI dataset [13] odometry and [14] TUM RGB-D freigburg3 dynamic scene dataset, in our ODTS system and compared the absolute trajectory error RMSE values, observe the qualitative and quantitative results and discuss our results below.

In Table 1, we show the RMSE results of freiburg3 dataset sequences and the average (%) improvement in our system. We select the dataset from the TUM RGBD database because

Table 1. Accuracy evaluation in terms of RMSE for TUM RGB-D dynamic dataset in our ODTS system against ORBSLAM2 [9] and DynaSLAM [5]

Sequences	Ours	[9]	[5]	Improv.
Walking rpy	0.027	0.662	0.035	36.0%
Sitting halfsphere	0.012	0.02	0.017	2.3%
Walking halfsphere	0.14	0.351	0.025	28.6%
Walking xyz	0.12	0.459	0.015	32.3%
Sitting xyz	0.13	0.009	0.015	10.5%
Desk_with_person	0.058	0.061	0.080	16.2%
Walking static	0.017	0.026	0.019	22.5%
Sitting static	0.007	0.010	-	30.0%

it contains a sequence of images of people moving around a room.

Next, we examine the performance of the system in the outdoor dynamic sequence for KITTI dataset sequence 00. This sequence shows a car moving in a busy neighborhood with people and other vehicles. We plot trajectory results in Fig.6, from running the KITTI visual odometry dataset [13] on the ODTS system.

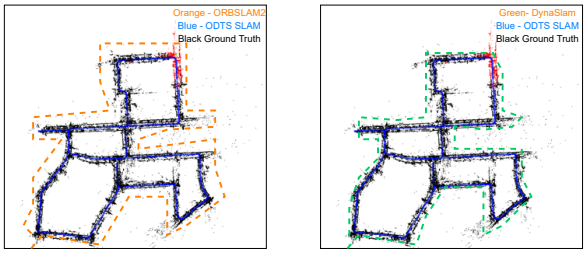

Fig. 6. KITTI odometry dataset sequence trajectory results for ODTS and ORBSLAM2(left), and ODTS and DynaSlam(right).

Better estimation from ODTS trajectory for KITTI Sequence Fig 6. We examine the trajectory estimation of our system against the ground truth and ORBSLAM2. Then, again against the DynaSlam system. ODTS Slam trajectory results are more accurate to the ground truth compared to the mentioned previous works.

For our image sequences, we observed the number of feature points tracked consecutively for every five frames and recorded the average table below.

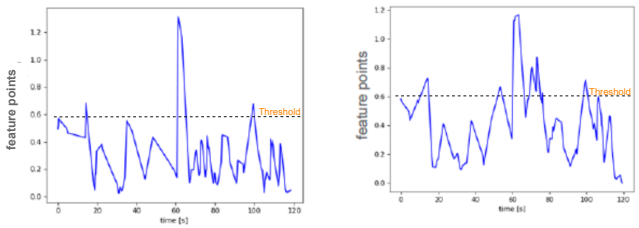

Fig. 7. Threshold determination for the number of feature point tracking: (left) the background feature point tracking, (right) the objects feature point tracking.

The number of feature points tracked by our system for our image sequences is 738 for background feature points and 1150 for object feature points. The fewer background feature points provide consistent tracking at a determined hard threshold. This considerably improves feature points tracking and mapping.

IV. DISCUSSION

In future research, we will work towards executing the system in real-time and further examine the impact of adjusting weights differently based on the scenes, either indoor or outdoor, and track the feature points for outdoor sequences

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: a survey from 2010 to 2016. ipsj t comput. vis. appl. 9, 16 (2017)."

[3] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[4] H. Liu, G. Liu, G. Tian, S. Xin, and Z. Ji, "Visual slam based on dynamic object removal," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 596–601.

[5] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[6] J. Li, L. Pei, D. Zou, S. Xia, Q. Wu, T. Li, Z. Sun, and W. Yu, "Attention-slam: A visual monocular slam learning from human gaze," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6408–6420, 2020.

[7] L. Lyu, Y. Ding, Y. Yuan, Y. Zhang, J. Liu, and J. Li, "Doc-slam: Robust stereo slam with dynamic object culling," in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, 2021, pp. 258–262.

[8] D. Lai, C. Li, and B. He, "Yo-slam: A robust visual slam towards dynamic environments," in *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*. IEEE, 2021, pp. 720–725.

[9] R. Mur-Artal, "Tardo s j d. 2017 orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, pp. 1255–1262.

[10] N. S. Sakpal and M. Sabnis, "Adaptive background subtraction in images," in *2018 International Conference on Advances in Communication and Computing Technology (ICACCT)*. IEEE, 2018, pp. 439–444.

[11] T. Chu, Y. Chen, L. Huang, Z. Xu, and H. Tan, "A grid feature-point selection method for large-scale street view image retrieval based on deep local features," *Remote Sensing*, vol. 12, no. 23, p. 3978, 2020.

[12] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.