A monocular ORB-SLAM in dynamic environments

Linyan Cui¹, Fei Wen*¹

¹Image Processing Centre, School of Astronautics, Beihang University, Beijing 100191, China
*Corresponding author: Fei Wen(wfei@buaa.edu.cn)

Abstract. The well-known ORB-SLAM is capable to deal with the localization and mapping tasks in static scenes. But in a dynamic environment, it is hard to perform well as the positions of 3D points in the map may change over time. In order to solve the problem, we propose a new slam framework, i.e. DORB-SLAM, which extends the application range of the ORB-SLAM in dynamic circumstances. We compared the appearance of the image blocks centered at the projection of the 3D points in current frame and the reference frame to decide whether the points are dynamic points or not. We test the DORB-SLAM in a public dataset and the results indicate that the algorithm is more robust than the origin ORB-SLAM in dynamic environments.

1. Introduction
In the computer vision and robotic communities, visual Simultaneous Localization and Mapping (vSLAM) becomes more and more popular. It plays an important role in driverless car, unmanned aerial vehicles as well as augmented reality.

There are two approaches to process information in a SLAM system: filtering methods and keyframe methods. The filtering methods use probability distribution to summarise the information while the keyframe methods use Bundle Adjustment (BA)[1] to perform an optimization process. Strasdat et. al.[2] proved that keyframe bundle adjustment obtains more accurate result than filtering methods in the same consuming time. PTAM[3], proposed by Klein and Murray, is the first SLAM system which uses keyframe method rather than filtering method. PTAM also separates the tracking and mapping tasks and runs them in two parallel threads. Engle et. al. proposed LSD-SLAM[4-5] which directly uses the image intensity to get camera pose and can build a semi-dense map in a large scale, which is more useful than sparse point cloud map. However, the loop detection of LSD-SLAM is also based on feature extraction. ORB-SLAM[6-7], the work of Mur-Artal et. al., utilises ORB features[8] to tracking, mapping, relocalization and loop closing. ORB-SLAM is one of the most popular keyframe based SLAM system and has more accurate state estimation result than the systems mentioned above. DSO[9] is proposed by Engel et. al.. It tracks points evenly in an image through direct method and uses sliding window to optimize the state of camera. DSO is one of the most accurate direct SLAM systems, but it cannot detect loop.

However, the above SLAM systems can only operate in stationary scene. Once the scene contains the dynamic objects, the accuracy of pose estimation will be reduced. This is caused by the following reason. Some points in the map may change their positions due to the object movement, so we will get wrong data correspondence if we still use them to find matches. The wrong data correspondence will influence the accuracy of pose estimation of the current frame and the structure we reconstructed.
We combined the ORB-SLAM and the methods of [10] to get a robust monocular SLAM system in dynamic environments. The algorithm which reduces the influence of dynamic environments can be named detecting dynamic 3D points. In the rest of the paper, the algorithm will be illustrated in details.

2. Monocular ORB-SLAM
The overview of monocular ORB-SLAM can be seen in figure 1. The system consists of three parts: tracking, local mapping, and loop closing. They run in different threads in parallel.

The tracking thread is the main thread of the system. It initializes the system, computes poses of the first two keyframe, creates 3D map points. The initialization algorithm parallel computes homography matrix and fundamental matrix. If the scene is nearly planar or there is low parallax, the homography matrix is used to initialize the system, otherwise, the fundamental matrix would be applied. The tracking thread is also in charge of estimating the pose of camera. It extracts ORB features in the image and performs feature matching between the two consecutive frames, then the initial pose estimation is computed through these matched features. To find more feature matching and refine the pose, we project some map points of previous keyframes to current frame, which is called tracking local map. At last, the tracking thread decides if it is necessary to insert keyframe.

The local mapping thread maintains the map points, keyframes, and their relationships. It will cull a map point if lots of keyframes which should observe the map point cannot find it. If 90 percent above of map points observed by a keyframe can be found in other keyframes, the keyframe will be culled. The local mapping thread is also creating new map points through triangulation when the camera exploring the environment.

The loop closing thread detect loops and optimize the whole camera trajectory if a loop is verified. Loop detection algorithm compute the similarity between current frame and some related keyframes using ORB features in images. This strategy is useful to limit the accumulate error of pose estimation. Once detecting a loop, a 7 degree of freedom Sim(3) transformation will be computed between the two looped keyframes. Then an optimization procedure is performed to eliminate the accumulate error.

The ORB-SLAM chooses ORB features to perform slam tasks. In many aspects, such as track map points, relocalization, loop detection, and keyframe decision, ORB-SLAM has made some improvement. But it does not have strategy to deal with the dynamic object. Consequently, the accurate and robustness of ORB-SLAM will suffer a lot in a dynamic environment. For the reason above, it is important to incorporate the algorithm that can detect dynamic objects in ORB-SLAM.
3. Dynamic 3D point detection
Dynamic 3D point detection step aims at eliminating the dynamic 3D points in the map, we use the method in [10] to achieve this purpose. For each frame, we detect if the scene has changed. We use the histogram stands for the corresponding frame and select several keyframes which observe the most same 3D points with current frame, then calculate the correlation coefficient between keyframes and current frame. If the correlation coefficient is less than 0.9, the scene of current frame possibly has changed. So, we project the map points in the keyframe to current frame. For a 2D feature point \( p \) in keyframe, its corresponding 3D point is denoted as \( P \) and its projection in current frame is denoted as \( p' \). We computer the appearance difference of the patch centred at \( p \) in the current frame with respect to the patch centered at \( p' \) in the keyframe:

\[
D(p) = \min_{d} \sum_{a \in B(p)} |I_a - A_a \cdot I'_{a+\lambda}|
\]

where \( B(p) \) denotes the image-patch centred at \( p \). We apply an affine warping \( A_a \) as [11] to current patch, because the keyframe is typically a little far from current frame. Due to the estimation error, the depth of \( P \) may deviate from the true value. So, we violate the epipolar constrains to get a subpixel accurate feature correspondence (directly add a little translation \( d \) to projection position \( p' \)). If the difference \( D(p) \) is larger than a threshold, it is very likely that the point \( P \) has changed its position or occluded by other objects. So, we need to verify whether the occlusion occurs.

\( S(p') \) denotes the set of 2D points in current frame whose positions are near \( p' \). If the \( S(p') \) is empty, the \( P \) is more likely occluded by a dynamic object. This is because the points on the dynamic objects are not in the map. In this situation, the \( P \) should not be removed. In other cases, the \( S(p') \) is not empty. Denote \( a' \) as a member of \( S(p') \). The depth of \( p' \), \( a' \) are \( d_p \) and \( d_{a'} \). If for all \( a' \) in the \( S(p') \), \( d_{a'} > d_p \), the points in \( S(p') \) cannot occlude \( P \), we should remove \( P \) from the map. Otherwise, for these points \( a' \) whose \( d_{a'} < d_p \), there are two cases. First, \( A \) and \( P \) belong to a same dynamic object. Second, \( P \) is on a static object and occluded by \( A \). We project \( A \) and \( P \) to the keyframe in which the \( P \) is created. If the projection position of two points is near, it belongs to the first case, \( P \) should be removed. Otherwise, if for all points in \( S(p') \), they belong to the second case, we should maintain the 3D point \( P \). The above procedure is shown in figure 2.

4. Evaluation
We have tested our system in a popular public dataset: TUM RGB-D dataset[11]. It contains a category named dynamic objects which is proper to evaluate our system. We have discarded the \textit{fr3_sitting_static}, \textit{fr3_walking_static}, \textit{fr3_sitting_rpy}, and \textit{fr3_walking_rpy} because the first two sequences contain no motion and the last two sequences contain strong rotations which is not suitable for monocular slam system. Experiments were conducted on a desktop computer with an Intel Core i5-6500(4 cores @ 3.20GHz) and 8GB RAM. The sequences of TUM RGB-D dataset were recoded at 30Hz and a resolution of 640×480. The ground truth trajectories of these sequences are provided by a high-accurate motion-capture system. We operate the system five times per sequence and record the median result to get more reasonable conclusions, the ORB-SLAM2 runs in the same sequences for comparisons. Because the monocular slam system cannot estimate the scale of the trajectories, we perform a Sim(3) transformation to compare the system with ground truth. The keyframe trajectory root mean square error (RMSE) of our system and ORB-SLAM is showed in table 1.
Decide if occlusion occurs

- Compare the histogram of current frame and keyframe
  - Correlation < 0.9

- Project map points(P) in keyframe(p) to current frame(p')

- Compare the appearance of image patch centred at p and p'
  - (dp) > threshold
  - The Scene is likely have changed. Decide if occlusion occurs

- Take all features around p in keyframe, compose S(p)

- S(p) is empty?
  - Yes
    - P should be removed
  - No
    - Project p and a' in S(p) whose depth are less than the depth of p' to the keyframe in which p' is created
      - For all a', the projection is far from the projection of p'?
        - Yes
          - P should be removed
        - No
          - P should be maintained

- P is occluded

Figure 2 The steps of dynamic 3D point detecting

It can be seen in table 1 that DORB-SLAM achieves higher or similar performance than ORB-SLAM in terms of accuracy. It benefits by removing dynamic 3D points in the map. In the sequence fr2_desk_perpon, the RMSE of DORB-SLAM is slightly higher than ORB-SLAM. We think it is because the dynamic objects in the sequence are small and the static textured background occupies the most area of image. It is worth mentioning that the robustness of DORB-SLAM is better than ORB-SLAM in dynamic environments. In sequences fr3_sit_halfsph, fr3_walk_xyz, and fr3_walk_halfsph, the performance of ORB-SLAM is unstable. Once it uses large amount of dynamic points, the accuracy is going to be poor. Figure 3 shows the screenshot of two systems. It is obvious that DORB-SLAM eliminating the dynamic points.

Figure 3. Left: ORB-SLAM, Right: our system. It shows the features tracked in the same frame in the sequence fr3_sit_halfsph. In our system, the points on the person are detected and removed from the map.
Table 1. Comparisons of accuracy in TUM RGB-D dataset for DORB-SLAM and ORB-SLAM2.

| Length (m)  | RMSE(m)  | ORB-SLAM | DORB-SLAM |
|------------|----------|----------|-----------|
| fr2_desk_person | 17.044 | 0.009 | 0.009 |
| fr3_sit_xyz    | 5.496  | 0.011 | 0.010 |
| fr3_sit_halfsph| 6.503  | 0.016 | 0.011 |
| fr3_walk_xyz   | 5.791  | 0.013 | 0.011 |
| fr3_walk_halfsph| 7.686 | 0.021 | 0.016 |

5. Conclusion and discussion
In this paper, we utilize the ORB-SLAM and the dynamic points processing algorithm to make up a slam framework, i.e. DORB-SLAM. It detects dynamic 3D points in the map and then remove them to eliminate the influence of dynamic environments. The experiments proved the efficiency of the system. The accuracy and robustness in a dynamic environment can be improved by incorporating object recognition. The semantic map can provide a prior for tracking thread.

Acknowledgments
This work is partly supported by the National Natural Science Foundation of China (61875003) and the Fundamental Research Fund for the Central Universities (YWF-18-BJ-Y-119).

References
[1] Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000) Bundle adjustment: a modern synthesis. In: Vision algorithms: theory and practice, pp. 298–372.
[2] Strasdat, H., Montiel, J. M. M., and Davison, A. J. (2012) Visual SLAM: Why filter? Image and Vision Computing, vol. 30, no. 2, pp. 65–77.
[3] Klein, G. and Murray, D. (2007) Parallel tracking and mapping for small AR workspaces. In: IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pp. 225–234.
[4] Engel, J., Sturm, J., and Cremers, D. (2013) Semi-dense visual odometry for a monocular camera. In: Proceeding of the IEEE International Conference on Computer Vision, pp. 1449-1456.
[5] Engel, J., Schöps, T., and Cremers, D. (2014) LSD-SLAM: Large-scale direct monocular SLAM. In: European Conference on Computer Vision (ECCV), pp. 834-849.
[6] Mur-Artal, R., Montiel, J., and Tardos, J. (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. Transactions on Robotics, 31(5):1147–1163.
[7] Mur-Artal, R., Tardós, JD. (2017) ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics, 33(5):1255-1262.
[8] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., (2011) ORB: an efficient alternative to SIFT or SURF. In: IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, pp. 2564–2571.
[9] Engel, J., Koltun, V., and Cremers, D. (2018) Direct Sparse Odometry. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 3, pp. 611–625.
[10] Wei Tan, Haomin Liu, Zilong Dong, Guofeng Zhang and Hujun Bao. (2014) Robust monocular SLAM in dynamic environments. IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, Australia, pp. 209-218.
[11] Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D.. (2012) A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, pp. 573–580.