MATCHING FILTER-BASED VSLAM OPTIMIZATION IN INDOOR ENVIRONMENTS

Shuangfeng Wei[1,2,3,4], Shangxing Wang[1]

[1] School of Geomatics and Urban Spatial Information, Beijing University of Civil Engineering and Architecture, Beijing 102616, China.

[2] Engineering Research Centre of Representative Building and Architectural Heritage Database, Ministry of Education, Beijing 102616, China.

[3] Key Laboratory for Urban Spatial Informatics of Ministry of Natural Resources, Beijing 102616, China.

[4] Beijing Key Laboratory for Architectural Heritage Fine Reconstruction and Health Monitoring, Beijing 102616, China.

KEY WORDS: Feature Matching, Motion Consistency Constraints, Geometric Models, VSLAM, Dense Point Cloud

ABSTRACT:

An important factor that reduced the accuracy of motion trajectories in existing VSLAM (Visual Simultaneous Localization and Mapping) systems is the poor estimation of the position pose of the vision odometer. The existing methods generate many incorrect matches during the feature matching process, resulting in low computational accuracy of rotations and translations between cameras, which further leads to a reduction in the robustness of the overall system. In addition, the sparse feature point maps do not provide a detailed description of the surrounding environment, which makes it difficult for the devices equipped with VSLAM systems to perform advanced tasks such as navigation, path planning and human-computer interaction. To address the accuracy problem, we select the set of matches from existing feature matching algorithms based on the motion consistency constraint and use a random sampling consistency algorithm to obtain the best quality matches from the selected samples for computing the geometric transformation model and estimating the current pose. To address the problem of sparse map points, we use the depth information from the RGB-D or Stereo camera to build a dense map module to ensure that information about the surrounding environment is recorded as a point cloud, which provides data support for the implementation of advanced tasks of the device.

1. INTRODUCTION

A device with SLAM (Simultaneous Localization and Mapping) technology can build a real-time map of its surroundings and localise itself in an unknown environment based on the data collected by sensors (Kazerouni, 2022), which is a prerequisite and basis for all tasks such as navigation, obstacle avoidance and path planning. Currently, Visual SLAM is one of the hot directions in SLAM research as it can acquire rich information about the surrounding environment with inexpensive vision sensors such as monocular, stereo, and RGB-D cameras (Chen, 2022).

There is no doubt that traditional visual SLAM frameworks and algorithms have achieved many results and have outstanding performance in the ideal indoor environment (Mur-Artal, 2017; Campos, 2021). However, these algorithms are suffering from problems such as simple manual design of feature descriptors and high error rate of feature matching sets, which leads to poor quality of point cloud maps with low accuracy of pose estimation. In addition, sparse point cloud maps are not helpful for the extension of more advanced functions such as navigation and human-machine interaction of the device. Therefore, it is important to improve the positional accuracy further and to build dense point cloud maps. Currently, with the improvement of computer hardware performance and the optimization of algorithms at the front and back end of visual SLAM. Many researchers have considered to migrate the functional modules that can only be used for map initialization, which are limited by hardware performance, to the whole VSLAM system, thus globally improving the accuracy of the positional estimation.

In this paper, a visual SLAM scheme combining match filtering is proposed based on the ORB-SLAM2 algorithm framework to address the problem of a large number of incorrect matches in feature matching sets. In the visual odometry, the feature similarity constraint and geometric similarity constraint are successively enforced on the matching sets to obtain a higher correct rate of matching sets to improve the accuracy of positional estimation and localization. For map reconstruction, VSLAM uses the pose information from the visual odometry estimation and combines it with the depth information of key frames to construct a dense point cloud map. The dense map provides an important foundation for the robot to perform tasks such as navigation, obstacle avoidance, path planning and human-robot interaction.

Related Work

2. RELATED WORK

The ORB algorithm (Rublee, 2011) is a binary manual descriptor with high matching performance, generated by the FAST (Rosten, 2006) feature detection algorithm combined with improved BRIEF (Calonder, 2010) descriptors, which is widely applicable in VSLAM systems. This algorithm is currently the most advanced algorithm in visual SLAM compared to traditional feature matching algorithms such as SIFT (Lowe, 2004), SURF (Bay, 2006), BRISK (Leutenegger, 2011) and KAZE (Alcantarilla, 2012). However, its performance is not stable in complex and variable scenes such as light and dark changes, viewpoint changes and weak textures, and it is prone to many incorrect matches, which seriously reduces the accuracy of pose estimation and the quality of environmental maps.

Researchers using deep learning techniques have developed many excellent feature matching algorithms as well. LIFT (Yi, 2016) introduced a novel deep learning network architecture that implements a complete pipeline of feature point processing, that is, detection, direction estimation and feature description. The LIFT-SLAM (Bruno, 2021) system, which uses LIFT as the front-end feature extraction module, achieves desirable results in texture-rich scenes. SuperPoint (DeTone, 2018) proposes a self-supervised training framework for interest point detectors and descriptors applicable to multi-view geometry problems. The method can detect richer interest points than traditional algorithms and has better single-strain estimation results compared to LIFT, SIFT and ORB. The GIFT (Liu, 2019) algorithm proposes a descriptor with transform invariance to compute feature descriptions for the corresponding feature points but lacks feature point detection functionality. The Patch2Pix

(Zhou, 2021) method achieves good performance by detecting and matching feature points in two steps: coarse matching and fine matching. However, the method is not transforming invariant and is not robust to changes in viewpoint. In addition, it cannot be further applied in VSLAM due to the absence of computational descriptors.

Whether using traditional feature matching algorithms or deep learning technology based visual odometry, good results have been achieved to some extent. However, there is a problem in the feature matching set that should not be ignored. There are many incorrect matches in the existing feature matching methods, resulting in reduced accuracy or even non-convergence of the positional estimation, as well as causing the constructed environment maps to contain many errors. Therefore, how to better remove erroneous matches from the matching set becomes an important factor to improve the performance of VSLAM. CODE (Lin, 2017) proposed a non-linear regression technique, using which the coherence-based separability constraint can be discovered from high-noise matches and embedded in the corresponding likelihood model. Using the model, it is possible to filter false matches in the nearest neighbourhood of a matching set, but the method is computationally complex and slow, which is not conducive to working in image streams. GMS (Bian, 2017) is a simple method by wrapping motion smoothing as the statistical likelihood of a certain number of matches in a region. It can convert a high number of matches into a high match quality, thus providing a real-time, robust feature filtering algorithm. RFM-SCAN (Jiang, 2019) creatively transforms feature matching into a spatial clustering problem with outliers. It does so by adaptively clustering the initial set of matches into several sets of motion-consistent clusters and an outlier cluster set. In addition to this, it devises an iterative clustering strategy to ensure improved matching performance in the presence of severe data degradation. Where the geometric model is known, the application of geometrically constrained models such as RANSAC (Fischler, 1981), Graph-Cut RANSAC (Barath, 2018) and MAGSAC (Barath, 2020) can likewise greatly improve the accuracy of the positional estimation.

Most of the existing algorithms use a combination of feature similarity constraints and geometric constraints to remove outlier matches from the matching set, but there are still problems such as high complexity of the algorithm, large computation, and inability to guarantee real-time, while the addition of deep learning techniques will greatly increase the overhead of computational resources, which is not conducive to the integration and embedding of devices. To address the above problems, this paper proposes a visual SLAM algorithm based on the ORB-SLAM2 algorithm as a basic framework to construct dense point cloud maps with improved matching filtering.

3. METHODOLOGY

3.1 Overview

To address the impact of error matching on visual SLAM, this paper proposes a visual SLAM algorithm combined with matching filtering to build indoor dense point cloud maps, and the overall framework of the algorithm is shown in Figure 1. ORB-SLAM2 is used as the basic framework, and the matching filtering module and the dense map building thread are added to remove the effect of incorrect matching on positional estimation and map building, while the depth information is used to build a dense map of the surrounding environment.

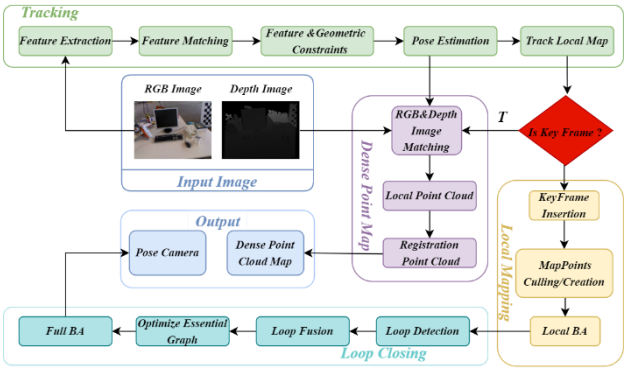


Figure 1. Overall flow and framework of the algorithm.

After system start-up, the RGB-D camera acquires both colour and depth images. The tracking module first extracts ORB feature points from the colour image. Then, the feature points are matched by ORB feature descriptors, and the matching set is filtered with the improved GMS algorithm. Finally, the geometric transformation model is determined by calculating both the homography and the fundamental matrix using the matched set, which is decomposed to obtain the positional transformation information. In addition, the tracking thread determines whether the current image is a keyframe based on the conditions. The dense builder module first adds the colour and depth images selected as keyframes to the queue. Next, the colour and depth images at the top of the queue are fetched and combined with the camera's intrinsic matrix to create a local colour point cloud. Finally, the local point cloud is stitched together based on the pose information of the current frame provided by the tracking thread.

3.2 GMS-RANSAC

The camera sensor has a continuing motion in the real scene, so the pixels about the feature points in the image have the same motion, therefore a certain number of feature points at both ends of the correct match have the same match in their respective neighbourhoods, as shown in Figure 2.

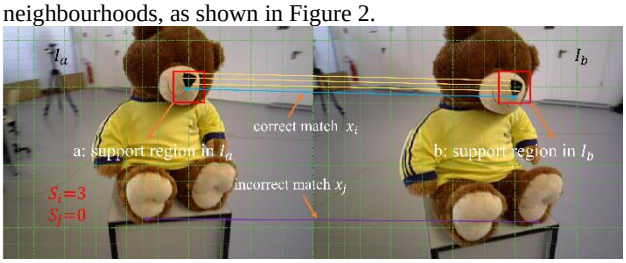


Figure 2. Mesh-based Motion Smoothing Constraints. In this image, x_i represents a correct match and the red box is its neighbourhood range. The same 3 matches (yellow coloured line segments) exist in that neighbourhood range, and we count the number of all such matches in the x_i neighbourhood as the score of the neighbourhood it is in. The x_j represents an incorrect match where there is no identical match within its neighbourhood, and therefore its neighbourhood score is 0.

The GMS algorithm models the binomial distribution of correct and incorrect matches in their respective domains from a probability estimation perspective, as shown in equation (1), respectively.

$$S_i \sim \begin{cases} B(n, p_t), & x_i = true \\ B(n, p_f), & x_i = false \end{cases} \quad (1)$$

where p_t denotes the probability of a correct match, p_f denotes the probability of an incorrect match, and S_i denotes the support of the match for x_i, which is calculated from equation (2), minus one to remove the effect of itself.

$$S_i = |\chi_i| - 1 \tag{2}$$

Finally, whether the match is true or not is judged by comparing the support of the match for S_{ij} with the threshold τ_i, as shown in equation (3), where in practice s_f can be obtained by averaging the number of all the matches involved, while the threshold τ_i represents the cut-off for finding a distinction between correct and incorrect matches, as shown in Figure 3.

$$cell - pair\{i,\ j\}\epsilon \begin{cases} T, & S_{ij} > \tau_i = \alpha\sqrt{s_f} \\ F, & otherwise \end{cases} \tag{3}$$

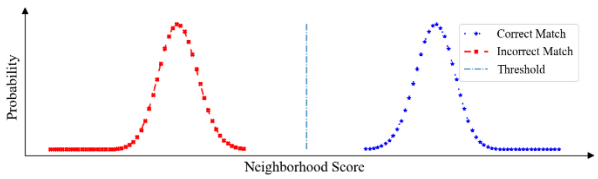


Figure 3. Probability distribution relationship of neighbourhood scores for correct or incorrect matches. In this image, the probability of incorrect matches peaks in the low scoring interval and correct matches in the high scoring interval. This means that the two are obeying different probability distribution models, so we can find a value for the appropriate neighbourhood score as a threshold that can reasonably distinguish between the two.

The geometric constraints require a predefined transformation model, usually a homography **(H)** or a fundamental matrix **(F)** in VSLAM, and the choice between the two cannot be determined without any a priori information, thus the accuracy needs to be calculated and compared simultaneously to finalize the determination. These problems are usually solved by the RANSAC-like (RANdom SAmple Consensus) algorithm. The minimum number of matching pairs required for the model to be computed is chosen randomly, then the remaining pairs are brought into the model to check the score of the model, and iterations are performed until the optimal geometric model is selected. The number of iterations is therefore particularly important, and the number of iterations k is given by the RANSAC algorithm as shown in equation (4).

$$k = \frac{\log(1-p)}{\log(1-e^m)} \tag{4}$$

Where p is the probability of obtaining the correct model, m denotes the number of points needed to calculate the model, i.e., $m = 4$ if calculating the single response matrix. e denotes the percentage of internal points is a priori a value that represents the ratio of the number of points in the data that fit the true model to the total number, however the value of e is difficult to determine in practice, but according to equation (4) the higher the percentage of internal points, the less iterations the shorter the time taken.

To improve the matching accuracy and reduce the computational time consumption, a combined GMS and RANSAC algorithm has been developed. First the coarse matching set obtained by the feature matching algorithm is filtered for the first time by applying the motion consistency constraint to it to obtain a matching set with a high internal point rate. This set is then used to build a geometric model and the coarse matching set is filtered with this model to obtain a set of matches with a high correct rate. Finally, the filtered matching set is used to obtain the geometric model again. Due to the high quality of the samples for estimating the poses, the accuracy of the visual odometer can be greatly improved, and the overall stability of the system can be guaranteed.

3.3 Dense mapping

A visual SLAM solution with sparse feature points can achieve simple localization functions, but for more advanced tasks such as path planning, obstacle avoidance and human-machine interaction, a dense point cloud map is indispensable. With the development of cameras such as binoculars and RGB-D cameras that can acquire image depth information, it is possible to add a dense map building module to existing solutions.

After the current colour image is selected as a keyframe, the dense mapping thread will first add the current keyframe and its depth image to the queue, and then based on the conversion equation (5) from the camera coordinate system to the pixel coordinate system, the position of each pixel point in the keyframe image can be obtained in the camera coordinate system, so that the point cloud of the keyframe can be recovered, as shown in Figure 4.

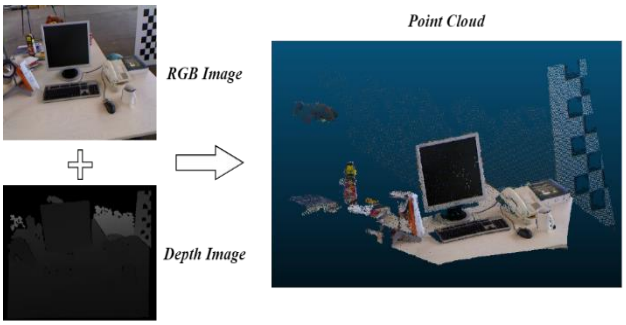


Figure 4. Point cloud display of key frames. We can use the depth image of the keyframe and the camera's internal reference matrix to create a point cloud of information about the surrounding environment, while using the colour image of the keyframe to colourise the point cloud.

$$s\boldsymbol{p} = s\begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{K}\boldsymbol{P}_c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5}$$

Where s denotes the scale relationship factor between the depth value and the actual spatial distance, given by the depth camera. $\boldsymbol{p}$ denotes the position in the pixel coordinate system. **K** denotes the camera intrinsic matrix, which needs to be calibrated. $\boldsymbol{P}_c$ denotes the position in the camera coordinate system.

We can calculate the position of the point cloud in the world coordinate system for each frame by combining the key frame poses calculated by the tracking thread, as shown in equation (6). Where T_{wc} represents the pose of the transformation of the camera coordinate system to the world coordinate system and is a combination of the rotation matrix $\mathbf{R}_{wc}$ and the translation vector $\boldsymbol{t}_{wc}$. Finally, when the looping detection module finds a closure, the essential graph is globally optimized, and the point cloud is globally adjusted for stitching and output.

4. EXPERIMENTS AND EVALUATION

The improved feature matching algorithm in this paper was tested using the Mikolajczyk dataset (Mikolajczyk, 2005), which provides a variety of common real scenes including zoom and rotation, light change, image compression, viewpoint change, and realistic camera transformation models. Pose accuracy, computational time and dense map building experiments were conducted using the RGBD dataset from TUM.

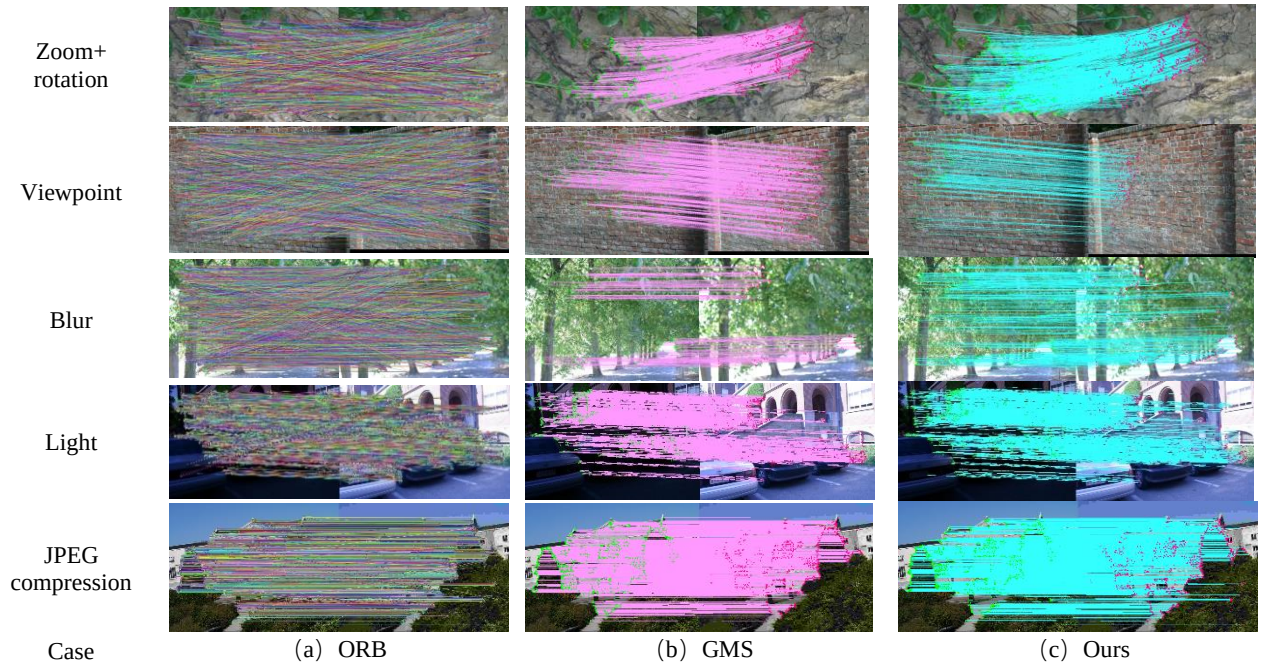
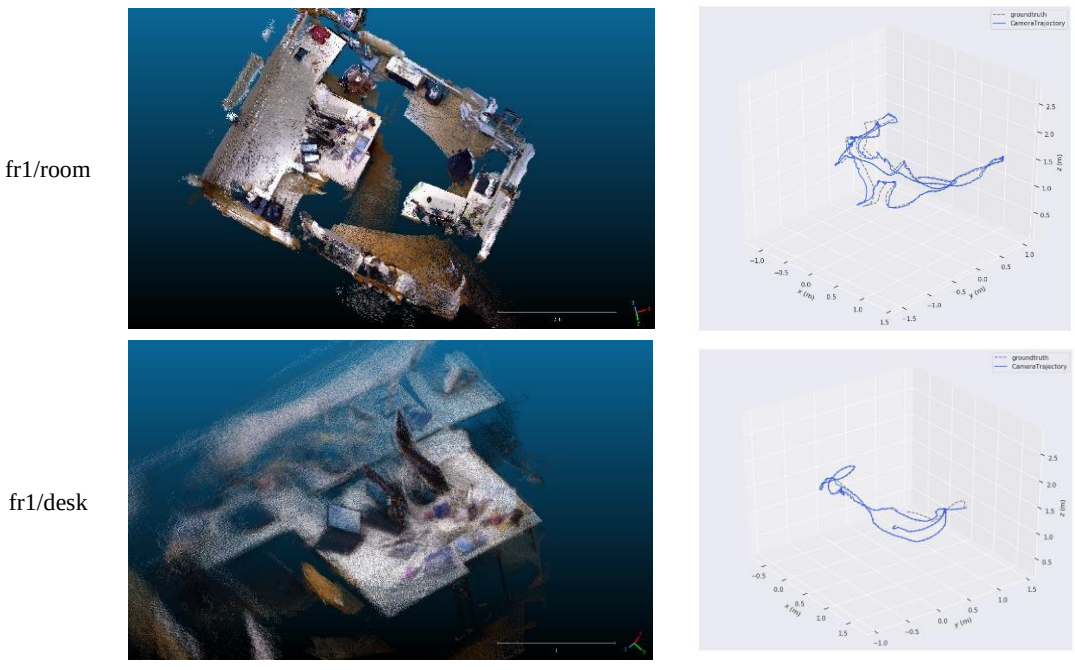


Figure 5. Showing the results of different feature matching.

Table 1. A comparison of the performance of different feature matching algorithms.

Scene	Correct matching			CMR (%)			Times (ms)		
	ORB	GMS	Ours	ORB	GMS	Ours	ORB	GMS	Ours
Zoom+rotation	1646	1179	1643	32.92	99.41	**99.82**	105.092	0.819	9.723
Viewpoint	300	141	279	5.92	12.88	**99.29**	134.043	1.004	1.926
Blur	787	346	652	15.74	99.42	**99.69**	133.224	0.933	12.511
Light	2044	1808	2042	45.98	99.39	**99.95**	103.395	0.833	3.792
JPEG compression	4092	3986	4092	81.84	99.67	**99.99**	111.423	0.820	1.351



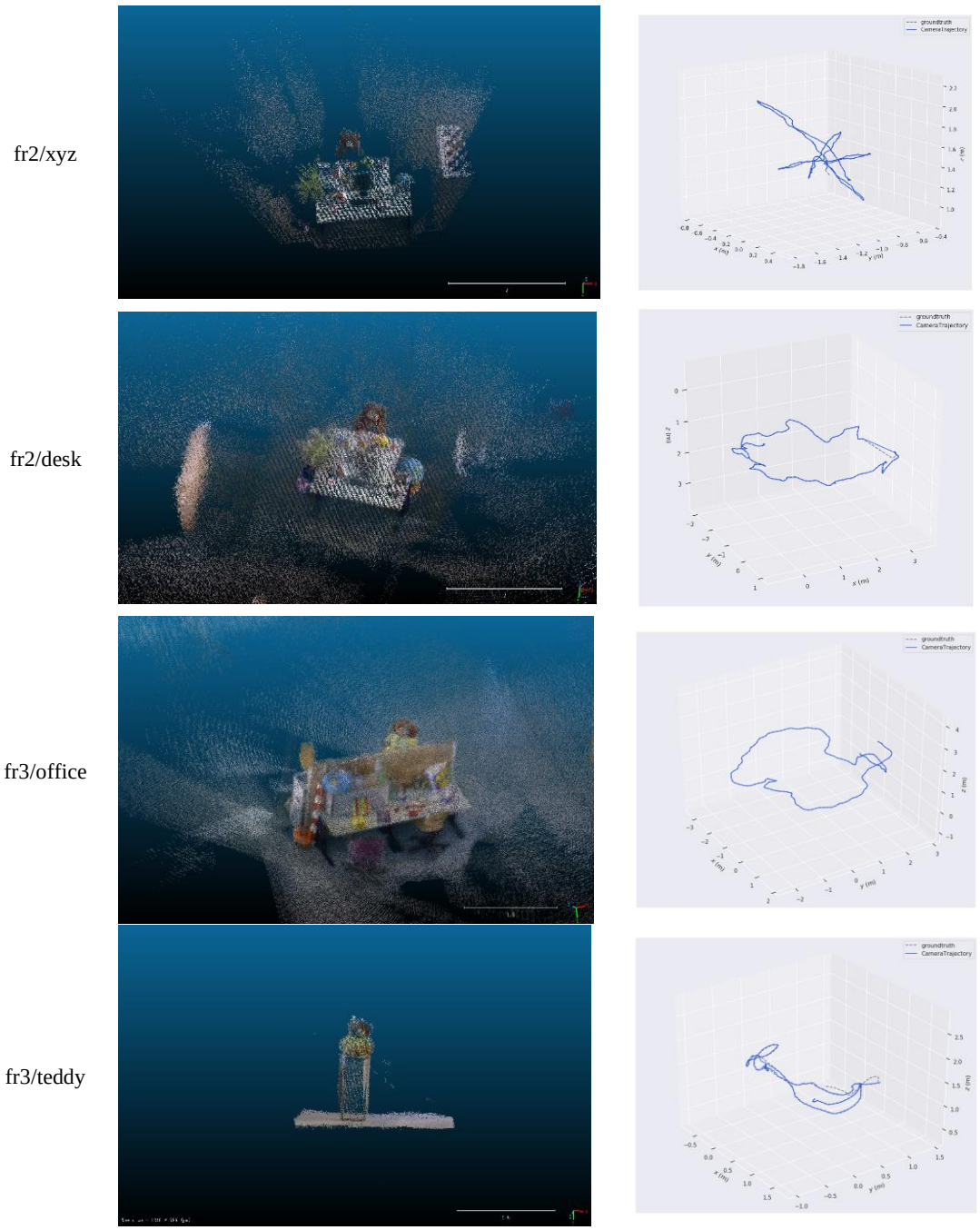


Figure 6. Display of point cloud maps and trajectory information.

4.1 Feature Matching

The performance of the improved feature matching algorithm is evaluated in terms of both correct matching rate and time consumption. Figure 5 and Table 1 show the matching results of the original ORB algorithm (a), the GMS algorithm (b) and the Ours (c) algorithm in which 5000 feature points are extracted, respectively, and it is easy to see from the figure and table that the Ours algorithm has many matches and no obvious errors.

In Figure 5, the Ours achieves significantly more matches than the other feature matching algorithms. Table 1 shows the time taken and the correct matching ratio (CMR) of the different algorithms, where the GMS and Ours do not calculate the time taken by the feature matching algorithms before it, so the actual time taken by this class of algorithms in feature matching should be increased by the time taken for feature extraction and matching. There is no doubt that ours achieves a noticeable advantage in viewpoint transformation and blur processing. Existing manual feature descriptors are robust in scenes such as rotation, scaling, movement, and brightness changes, but are weak in scenes such as viewpoint transformations, which are associated with more complex forms of motion. However, in practical cases, often, the motion involves a combination of rotation, scaling, and movement, so better matching of viewpoint transformations is fundamental to ensure improved accuracy in downstream applications.

4.2 Visualization

Sparse maps in visual SLAM resulting from feature points alone cannot provide advanced tasks such as semantic perception and indoor navigation for robots. Therefore, the use of colour images with its depth information to build dense point clouds offers the possibility to solve the above problem. Figure 6 shows the indoor scene and estimated camera trajectory recovered using the TUM dataset.

With column (a) we can easily see the actual environment information, ensuring that the 3D scene structure is recovered. Column (b) shows the camera trajectory information recovered by our VSLAM after synchronous alignment with the real trajectory information provided by the dataset. The two trajectory information are highly overlapping, thus ensuring the accuracy of the visual odometry positional estimation.

5. CONCLUSIONS

In this paper, we address the problem of many mis-matched feature points in current visual SLAM systems by applying a feature filtering algorithm which combines motion consistency constraints and geometric constraints to improve the accuracy of the system in estimating the positional transformation. However, the addition of the new feature filtering algorithm must increase the running time of the system, but it takes negligible time compared to the original algorithm, so it can still ensure the real-time operation of the system. In addition, the point cloud map created by the dense mapping module provides the basis for advanced tasks such as path planning, obstacle avoidance and human-machine interaction.

REFERENCES

Alcantarilla P F., Bartoli A., Davison A J.,2012. KAZE features. *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*. Springer Berlin Heidelberg, 214-227.

Bay H., Tuytelaars T., Van Gool L., 2006. Surf: Speeded up robust features. *Lecture notes in computer science*,3951: 404-417.

Barath, D., & Matas, J. 2018. Graph-cut RANSAC. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp.6733-6741).

Barath, D., Noskova, J., Ivashechkin, M., & Matas, J. 2020. MAGSAC++, a fast, reliable and accurate robust estimator. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1304-1312).

Bian, J., Lin, W. Y., Matsushita, Y., Yeung, S. K., Nguyen, T. D., & Cheng, M. M. 2017. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp.4181-4190).

Bruno, H. M. S., & Colombini, E. L. 2021. LIFT-SLAM: A deep-learning feature-based monocular visual SLAM method. *Neurocomputing,455*, 97-110.

Calonder M, Lepetit V, Strecha C, et al.2017. Brief: Binary robust independent elementary features.C*omputer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer Berlin Heidelberg(pp.778-792).

Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., & Tardós, J. D.2021. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics,37*(6), 1874-1890.

Chen, W., Shang, G., Ji, A., Zhou, C., Wang, X., Xu, C., ... & Hu, K.2022. An overview on visual slam: From tradition to semantic. *Remote Sensing*,*14*(13), 3010.

DeTone, D., Malisiewicz, T., & Rabinovich, A.2018. Superpoint: Self-supervised interest point detection and description. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp.224-236).

Fischler, M. A., & Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*,*24*(6), 381-395.

Jiang, X., Ma, J., Jiang, J., & Guo, X. 2019. Robust feature matching using spatial clustering with heavy outliers. *IEEE Transactions on Image Processing*, *29*, 736-746.

Kazerouni, I. A., Fitzgerald, L., Dooly, G., & Toal, D. 2022. A survey of state-of-the-art on visual SLAM. *Expert Systems with Applications*, *205*, 117734.

Leutenegger, S., Chli, M., & Siegwart, R. Y.2011. BRISK: Binary robust invariant scalable keypoints. *2011 International conference on computer vision* (pp. 2548-2555). Ieee.

Lin, W. Y., Wang, F., Cheng, M. M., Yeung, S. K., Torr, P. H., Do, M. N., & Lu, J. 2017. CODE: Coherence based decision boundaries for feature correspondence. *IEEE transactions on pattern analysis and machine intelligence*, *40*(1), 34-47.

Liu, Y., Shen, Z., Lin, Z., Peng, S., Bao, H., & Zhou, X. 2019. Gift: Learning transformation-invariant dense visual descriptors via group cnns. *Advances in Neural Information Processing Systems, 32*.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*, 91-110.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., ... & Gool, L. V. 2005. A comparison of affine region detectors. *International journal of computer vision*, 65(1): 43-72.

Mur-Artal, R., & Tardós, J. D. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, *33*(5), 1255-1262.

Rosten, E., & Drummond, T. 2006. Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9* (pp. 430-443). Springer Berlin Heidelberg.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. 2011. ORB: An efficient alternative to SIFT or SURF. *2011 International conference on computer vision* (pp. 2564-2571). Ieee.

Yi, K. M., Trulls, E., Lepetit, V., & Fua, P. 2016. Lift: Learned invariant feature transform. *Computer Vision–ECCV 2016: 14th*

European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14 (pp. 467-483). Springer International Publishing.

Zhou, Q., Sattler, T., & Leal-Taixe, L. 2021. Patch2pix: Epipolar-guided pixel-level correspondences. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4669-4678).