Object spatial localization by fusing 3D point clouds and instance segmentation

Chenfei Xia¹ · Shoudong Han^{1,2} · Xiaofeng Pan¹

Received: 15 November 2019 / Accepted: 7 February 2020 / Published online: 17 February 2020
© Springer Nature Switzerland AG 2020

Abstract

Real-time detection and acquisition of localization information of instance targets in real three-dimensional space plays an important role in application scenarios such as virtual reality simulation and digital twinning. The existing spatial localization methods without the aid of lidar and other equipment often have problems in restoring the real scale. In order to overcome this problem and achieve more accurate object spatial localization, an object spatial localization by fusing 3D point clouds and instance segmentation is proposed. This method obtains sparse 3D point cloud data by binocular stereo matching, which is used to describe the real scale and spatial location information of the object. Then uses deep learning method to perform monocular instance segmentation on the specific category target of interest, and the segmentation result is used as the front/background prior information to complete the coordinate correction and densification of the 3D point cloud data inside and outside the object contour. Compared with the unsupervised depth estimation methods based on deep learning, our method can quickly and accurately achieve the three-dimensional precise localization of the instance target and its various components in real-world scenes, and the accuracy in the indoor scene is more than 90%.

Keywords 3D Point Clouds · Binocular vision · Instance segmentation · Mask · Spatial localization

1 Introduction

Real-time detection and acquisition of localization information of instance targets in real three-dimensional space plays an important role in application scenarios such as virtual reality simulation and digital twinning. Different from the pixel coordinates of the two-dimensional image, the three-dimensional localization information is mainly divided into the following two categories: (1) reflect the relative positions between different objects; (2) reflect the actual three-dimensional coordinates of the object in the real world. Obviously, the second category of localization information has more practical application value. The existing object spatial localization methods can be mainly divided into three categories: hardware-assisted methods,

traditional geometric methods, and depth estimation based on deep learning.

Hardware-assisted methods rely on non-visual sensors. Weng et al. proposed a monocular three-dimensional object detection algorithm based on pseudo-laser point cloud [1], Yang et al. proposed and implemented a monocular vision SLAM-based UAV autonomous landing system with fusion optical equipment and image recognition technology [2], Wei et al. proposed a monocular image obstacle detection algorithm that combines laser radar sparse point clouds and camera natural image density clustering [3], mainly focused on obstacle detection, indoor localization, indoor path planning and other application scenarios [4–8]. Traditional geometric methods are generally based on binocular vision. Yang

✉ Shoudong Han, shoudonghan@hust.edu.cn | ¹Key Laboratory of Image Processing and Intelligent Control, Ministry of Education, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, 430074 Wuhan, China. ²Research Institute of Huazhong University of Science and Technology in Shenzhen, 518057 Shenzhen, China.

et al. proposed a high-precision binocular stereo vision system, and designed experiments to find the influence of various parameters on the localization accuracy [9]. Lin et al. extended visual localization to underwater and monitored fish behaviors in real time [10]. Depth estimation based on deep learning has developed rapidly. Tateno et al. applied depth estimation to simultaneous localization and mapping (SLAM), and generated semantically consistent scene reconstruction [11]. In 2018, Cao et al. proposed a monocular depth estimation algorithm based on deep fully convolutional residual networks for depth classification, which transformed the depth estimation problem into a classification problem with good effect [12]. Zoran et al. focused on the visual features of the relative near-far relationship [13], Chen et al. improved the monocular depth recovery in an unconstrained environment and significantly improved the depth perception ability of the wild monocular image [14]. In order to meet the requirements of unsupervised depth estimation, Garg et al., Kuznetsov et al., Godard et al. and Tosi et al. also proposed related methods [15–17].

However, these three kinds of methods mentioned above have some defects, such as signal transmission is easily disturbed or even shielded, the point cloud is sparse, the scene mobility is poor, and the true scale information cannot be restored. In order to solve the current problems of object spatial localization, in this paper we propose an object spatial localization method fusing 3D point clouds and instance segmentation. The main contributions are as follows:

1. Propose a integrating method of 3D point clouds and instance segmentation, which can accurately extract the dense pixel coordinates of the instance target and its components, and accurately define the semantic classification properties of the pixels.
2. Design an optimization strategy for edge error and background interference to further improve the accuracy of localization.
3. Realize the transformation from image localization to video localization, and realize the real-time spatial localization and trail tracking of the instance target.

2 Fusion of 3D point clouds and instance segmentation

Algorithm 1: An overview of fusing 3D point clouds and instance segmentation

Input: Two 640×480 images as a stereo image pair, left image L_i , right image R_i

Output: Object spatial three-dimensional coordinates (x, y, z)

- 1: Do image rectification, stereo matching and depth recovery for the image pair.
- 2: Save the 3D point clouds, the dimension is $640 \times 480 \times 3$.
- 3: Use the Mask R-CNN algorithm [19] on the left image L_i , output the instance categories, regression boxes, and corresponding masks.
- 4: Extract the specified target mask $mask_{ori}$, save the image coordinates of pixels in the target contour, the dimension is $PN \times 2$, and PN is the number of target pixels.
- 5: Sampling pixel points, SN is the number of sample points. When $PN > 3000$, randomly sample 3000 pixels and $SN = 3000$; when $PN \leq 3000$, select all pixels of the target and $SN = PN$.
- 6: Take the SN sample point coordinates as indexes, extract the three-dimensional coordinates of the corresponding positions in $640 \times 480 \times 3$ point clouds.
- 7: Mean filtering of all the three-dimensional coordinates of sample points and output the object spatial three-dimensional coordinates (x, y, z) .

Integrating 3D point clouds and instance segmentation is the key step of object spatial localization. The network framework of fusing 3D point clouds and instance segmentation is shown in Fig. 1. The network is mainly composed of two branches, binocular vision branch and instance segmentation branch. In Fig. 1, the binocular vision branch is inside the green dotted rectangle, and the instance segmentation branch is inside the red dotted rectangle. The input of the network is a stereo image pair, and the outputs are the three-dimensional spatial coordinates. For the instance segmentation branch, we adopt the classic Mask R-CNN algorithm as the backbone, and the framework is shown in the red dotted rectangle in Fig. 1 [19]. The algorithm uses region proposal network (RPN) to extract the regions of interest in the input image, each region of interest contains an object. Then use the pre-trained model and convolution network to classify and regress the object and output the object's category, bounding box and the binary mask.

As shown in Algorithm Algorithm I, it describes how to integrate the 3D point clouds and the object mask of instance segmentation. In the binocular vision branch, the left image L_i and right image R_i are stereo image pairs, which are taken by one binocular camera or two monocular cameras at the same time. Then exports the spatial 3D point cloud information corresponding to all the pixel points by image rectification, stereo matching and depth recovery to restore the parallax of the spatial points in the left and right images. The instance segmentation branch is used to segment the left image and output a binary mask $mask_{ori}$ with the same size as the original image. In $mask_{ori}$, each pixel value is 0 or 1, 1 indicates that the pixel belongs to the object, and 0 indicates that the pixel does

Fig. 1 Architecture for fusing 3D point clouds and instance segmentation

not belong to the object. In fusion, in order to obtain the pixel coordinates of the object in the image and ensure a relatively fast computing speed, when the total number of pixels of the target PN is more than 3000, the number of random sampling points $SN=3000$, otherwise $SN=PN$. Finally, the 2D image coordinates of SN sampling points are used as indexes to extract the corresponding 3D coordinates in the 3D point clouds generated by the binocular vision branch, and the final object spatial three-dimensional coordinates are obtained through mean filtering.

3 Optimization for edge error and background interference

The instance segmentation separates the background and the foreground, but there are still some problems such as edge segmentation is not precise enough and is easy to introduce background pixel noise. Therefore, the optimization of edge error and background interference is designed as shown in Algorithm Algorithm II. The input of the algorithm is the object binary mask $mask_{ori}$ generated by the instance segmentation branch and the 3D point clouds generated by the binocular vision branch.

Aiming at the problem that the edge segmentation is not precise enough, we make the convolution kernel of $n \times n$ as the corrosion factor $corrode$, and use the $corrode$ to scan the $mask_{ori}$ pixel by pixel. According to the steps shown in Algorithm Algorithm I and Algorithm II, the final mask $mask_{cor}$ can be obtained and fused with 3D point clouds. This edge optimization strategy can reduce

or even eliminate the possibility of selecting non-target pixels as sample points. Compare the original mask, 3×3 pixel corroded mask, 5×5 pixel corroded mask and 10×10 pixel corroded mask, the accuracy is improved sequentially. Therefore, the greater the degree of pixel corrosion, the lower the probability of introducing edge errors.

Algorithm 2: An overview of optimization for edge error and background interference

Input: 640×480 Instance segmentation target mask $mask_{ori}$, 3D point clouds which dimension is $640 \times 480 \times 3$

Output: Three-dimensional coordinates of the target after optimization (x', y', z')

- 1: Select a kernel of $n \times n$ and values are all 1 as the corrosion factor $corrode$, use the $corrode$ scan the $mask_{ori}$ pixel by pixel, take the minimum value of the corresponding position and operation as the scan result.
- 2: After the full-image scan is completed, the minimum value of the scan results of at most n^2 scans per pixel is selected as the new pixel value, and the final target mask is updated as $mask_{cor}$.
- 3: Mapping of the target mask $mask_{cor}$ and the $640 \times 480 \times 3$ 3D point clouds according to Algorithm I.
- 4: Clustering all sample point depths and mean filtering for each depth category.
- 5: Suppose that there are two categories of depth clusters, their numbers of point members are $class_{1n}$ and $class_{2n}$ respectively, their depth mean values are $depth_1$ and $depth_2$ respectively. When $depth_2 \geq 2 * depth_1$ and $class_{1n} \geq 5 * class_{2n}$, then judge that $depth_2$ is generated by the background points, delete the corresponding sample points.

Input: 640×480 Instance segmentation target mask $mask_{ori}$, 3D point clouds which dimension is $640 \times 480 \times 3$

Output: Three-dimensional coordinates of the target after optimization (x', y', z')

6: Mean filtering of all three-dimensional coordinates of undeleted sample points and output the optimized object spatial three-dimensional coordinates (x', y', z')

In order to solve the problem of background noise interference, the sample points involved in the calculation are clustered by depths. For example, for two types of points, if the numbers of pixels are $class_{1n}$ and $class_{2n}$ and the mean depths are $depth_1$ and $depth_2$ respectively, if there are $depth_2 \geq 2 * depth_1$ and $class_{1n} \geq 5 * class_{2n}$, that is, when the mean value of depths of one type is greater than twice the mean of the other type and the number of point members is less than one fifth, the class is judged to be a background class and will be filtered to avoid causing large deviations in depth.

The experimental results show that the optimization effectively improves the accuracy of object spatial localization. The experimental results are shown in Table 1.

Table 1 Optimization results for edge error and background interference

Experiment groups	True depth (cm)	Localization result (cm)	Accuracy
No optimization	122	122.7599	0.9938
	212	218.6942	0.9684
	300	322.1327	0.9262
Edge error optimization	122	121.9208	0.9994
	212	216.6119	0.9782
	300	318.2470	0.9392
Background interference optimization	122	122.3074	0.9975
	212	217.7474	0.9729
	300	320.2381	0.9325
Fusion optimization	122	122.0474	0.9996
	212	216.4758	0.9789
	300	318.1533	0.9395

Bold values indicate the best performance

4 Real-time spatial localization

There are three kinds of frame extraction methods for real-time detection of video: frame-by-frame detection, fixed-interval frame extraction, and key frame extraction based on inter-frame difference. Since the frame-by-frame detection is remarkable in detecting continuity and the key frame extraction based on the inter-frame difference have significant performance in detecting speed, further experiments are carried out on these two methods. The results of real-time localization are shown in Fig. 2, which is the real-time spatial localization results in different modes. In Fig. 2, the x-axis and y-axis respectively represent the scale in the real world. We select a point in the space as the reference, and in the experiment, we select the upper left corner of the indoor scene as the coordinate origin. Therefore, the abscissa X represents the horizontal position of the object relative to the fixed coordinate origin, and ordinate Y represents the vertical position, all in centimeters. In the experiment, the experimental object moves randomly, and the method proposed in this paper can output the object's trail in real time. The sample video is a 19-s video of moving pedestrians indoor. The video size is 640×480 , the frame rate is 20 fps, and the total number of frames is 392.

The frame-by-frame detection of the original video is shown in Fig. 2a, the speed is 2.78 fps but the detection is continuous, suitable for application scenarios that have low requirements for real-time performance. Key frame extraction based on inter-frame difference is shown in Fig. 2b, the detection speed is 19.99 fps, which is suitable for most video surveillance scenarios. From Fig. 2 we can see that different frame extraction strategies will bring different real-time localization performances, but in general, they can all roughly restore the real moving trail of the object.

Fig. 2 Real-time spatial localization results in different modes

5 Experiments

5.1 Evaluation of pixel corrosion and background filtering

We designed experiments to demonstrate the effectiveness of the optimization, the optimization results for edge error and background interference are shown in Table 1.

The experiment is divided into four groups, namely no optimization, edge error optimization, background interference optimization and fusion optimization. Each group carries out spatial localization of targets at short, medium and long distance, and calculates the accuracy. According to Table 1, the average accuracy is 0.9689 and the standard deviation is 0.0276 under different real depths, which shows the high accuracy and good robustness of our method. When use edge error optimization and background interference optimization respectively, the accuracy is higher than that without optimization; when use the two optimizations simultaneously, the accuracy is further improved. For the optimization methods, both of them are effective and the edge error optimization effect is more obvious. For the object distance, the closer the target is, the higher the localization accuracy is.

5.2 Comparison with unsupervised depth estimation based on deep learning

The experimental environment is an indoor scene. The devices are two Logitech C920 webcams. The basic framework is based on image stereo matching. It is impossible to use the common dataset for quantitative comparison. Therefore, a total of 30 sample images are collected and produced. To obtain the localization accuracy at different depths, each ten sample images are divided into a group.

The real depths are respectively short distance (122 cm), medium distance (212 cm), long distance (300 cm).

In order to evaluate the quantitative localization effect, we divide the experiment into two parts and do not use any optimization strategies. For the same sample data, Part A uses the method proposed in this paper to get the disparity map, Part B uses the unsupervised depth estimation methods based on deep learning [17, 18] respectively to get the disparity maps, and the following instance segmentation and 3D point clouds fusion process are completely the same as Part A, which guarantees the reliability of the experimental results. Table 2 shows the localization results at a short distance, Table 3 shows the localization results at a medium distance, and Table 4 shows the localization results at a long distance. We use the formula (1) to calculate the location accuracy, $depth_p$ represents the predicted depth and $depth_t$ represents the true depth, " $|*|$ " represents the absolute value calculation. In order to get the effective accuracy, that is, the object predicted depth is comparable with the real depth, the predicted depth should not be more than twice the real depth, otherwise it is meaningless.

$$\text{Accuracy} = 1 - \frac{|depth_p - depth_t|}{depth_t} \quad (1)$$

Although the results [17] and [18] provided show that the methods can achieve average accuracy of 97.9% and 98.1% respectively in KITTI, the depth estimation based on deep learning needs to be trained with a lot of specific datasets, which results in poor scene mobility. It can be seen from Tables 2, 3 and 4 that there is a huge gap between the comparison methods and the real scale, the calculation of accuracy is meaningless, so it is not listed in the Tables.

Table 2 Object localization results at short distance (true depth of 122 cm)

Sample number	Result in [17]	Result in [18]	Result in ours	Accuracy in ours
1	1631.4290	1868.9740	122.1234	0.9990
2	1627.8353	1805.0423	121.8419	0.9987
3	1703.1894	1789.3533	125.4417	0.9718
4	1670.2947	2039.6095	136.1700	0.8839
5	1546.7148	1737.9632	136.1260	0.8842
6	1551.4999	1866.1658	135.4211	0.8900
7	1679.7115	2140.4267	127.0008	0.9590
8	1671.4112	2086.6709	128.1065	0.9499
9	1548.4322	2142.8038	129.7639	0.9364
10	1521.0360	2217.4788	127.6231	0.9539
AVG	1615.1554	1969.4488	128.9618	0.9427
STDEV	67.1604	174.2539	5.3941	0.0438

Bold value indicate the best performance

Table 3 Object results at medium distance (true depth of 212 cm)

Sample number	Result in [17]	Result in [18]	Result in ours	Accuracy in ours
1	1642.1424	2219.9441	216.4929	0.9788
2	1685.7123	2244.7420	205.4757	0.9692
3	1971.6072	2836.6801	213.2234	0.9942
4	2037.2815	3062.9776	182.4187	0.8605
5	1798.1902	2506.9340	226.9386	0.9295
6	1743.4684	2304.8310	221.7642	0.9539
7	1646.0122	2192.1683	231.5842	0.9076
8	1706.0470	2200.6971	229.9983	0.9151
9	1527.6909	2196.3243	229.3984	0.9179
10	1582.9764	2198.0785	225.5890	0.9359
AVG	1734.1129	2396.3377	218.2883	0.9363
STDEV	162.3819	311.3204	15.1189	0.0394

Bold value indicate the best performance

Table 4 Object results at long distance (true depth of 300 cm)

Sample number	Result in [17]	Result in [18]	Result in ours	Accuracy in ours
1	2570.4146	3127.9641	318.2555	0.9391
2	2090.0023	3145.4263	308.1303	0.9729
3	2290.7849	4784.0648	321.2555	0.9291
4	2371.4686	4469.0900	282.7671	0.9426
5	2060.2942	3783.8016	312.2543	0.9592
6	2040.7220	3880.6105	289.5238	0.9651
7	2118.6416	4102.5382	299.8942	0.9996
8	2146.4319	4428.2460	296.9587	0.9899
9	2266.6953	3733.6186	305.8864	0.9804
10	2211.6918	3394.1572	311.4004	0.9620
AVG	2216.7147	3884.9517	304.6326	0.9640
STDEV	164.3563	566.2836	12.3258	0.0226

Bold value indicate the best performance

It can be seen from Table 2 that under the real distance of 122 cm, our method has great advantages in both mean value and standard deviation. The average localization result is 128.9618 cm, and the average accuracy is 94.3%.

From Tables 3 and 4 we can see that, similar to Table 2, our method still has significant advantages in mean value and standard deviation. The overall average localization accuracy is 94.8%, which has high accuracy and good robustness.

Under the same experimental conditions, the average accuracy of the method we proposed is 94.8%. Need no-training but only camera calibration for any scene. And the instance segmentation algorithm and key frame extraction technology can be combined to achieve real-time spatial localization.

6 Conclusion

In this paper, object spatial localization by fusing 3D point clouds and instance segmentation is proposed. The 3D point cloud data is obtained by binocular stereo vision and the object mask is obtained by instance segmentation. The coordinates of densely pixel points of the instance target and its components are further accurately extracted to precisely define the semantic classification of pixels and achieve the three-dimensional localization of objects in real space. In order to improve the accuracy, the optimization of edge error and background interference is designed and implemented, and the real-time spatial localization is also completed.

Experiments show that this method can also overcome the drawbacks of poor scene mobility and difficulty in restoring real scale in deep-learning-based depth estimation, and achieve fast, efficient and accurate spatial localization.

Acknowledgment This work was supported by the National Natural Science Foundation of China under Grant No. 61105006; the Shenzhen Strategic Emerging Industry Development Special Fund under Grant No. JCYJ20170307172130906; the Aerospace Science and Technology Foundation under Grant No. 2018-HT-HZ; the Open Fund of Key Laboratory of Image Processing and Intelligent Control (Huazhong University of Science and Technology), Ministry of Education under Grant No. IPIC2019-01.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- Weng X, Kitani K (2019) Monocular 3D object detection with pseudo-LiDAR point cloud. [arXiv:1903.09847v2\[cs.CV\]](https://arxiv.org/abs/1903.09847v2)
- Yang T, Li P, Zhang H et al (2018) Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments. *Electronics* 7(5):73
- Wei Y, Yang J, Gong C et al (2018) Obstacle detection by fusing point clouds and monocular image. *Neural Process Lett.* <https://doi.org/10.1007/s11063-018-9861-1>
- Díaz-Vilariño L, Boguslawski P, Khoshelham K et al (2016) Indoor navigation from point clouds: 3D modelling and obstacle detection. *ISPRS: Int Arch Photogram Remote Sensing Spatial Inf Sci* 41(B4):275–281
- Al-Kaff A, Meng Q, Martín D, et al (2019) Monocular vision-based obstacle detection/avoidance for unmanned aerial vehicles. In: 2016 IEEE intelligent vehicles symposium (IV), Gothenburg, Sweden, pp 92–97
- Jing C, Yi Z, Wei X (2018) Unsupervised indoor localization based on smartphone sensors, iBeacon and Wi-Fi. *Sensors* 18(5):1378
- Paracha KN, Rahim SKA, Soh PJ et al (2019) A low profile, dual-band, dual polarized antenna for indoor/outdoor wearable application. *IEEE Access* 7:33277–33288
- Ruotsalainen L, Kirkko-Jaakkola M, Rantanen J et al (2018) Error modelling for multi-sensor measurements in infrastructure-free indoor navigation. *Sensors* 18(2):590
- Yang L, Wang B, Zhang R et al (2018) Analysis on location accuracy for the binocular stereo vision system. *IEEE Photonics J* 10(1):1–16
- Lin K, Zhou C, Xu D et al (2018) Three-dimensional location of target fish by monocular infrared imaging sensor based on a L–z correlation model. *Infrared Phys Technol* 88:106–113
- Tateno K, Tombari F, Laina I, et al (2017) CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, USA, pp 6565–6574
- Cao Y, Wu Z, Shen C (2018) Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Trans Circuits Syst Video Technol* 28(11):3174–3182
- Zoran D, Isola P, Krishnan D, et al (2015) Learning ordinal relationships for mid-level vision. In: 2015 IEEE international conference on computer vision (ICCV), Santiago, Chile, pp 388–396
- Chen W, Fu Z, Yang D, et al (2016) Single-image depth perception in the wild. In: 2016 neural information processing systems (NIPS), Barcelona, Spain, pp 730–738
- Garg R, Vijay Kumar BG, Carneiro G, et al (2016) Unsupervised CNN for single view depth estimation: geometry to the rescue. In: 2016 fourteenth European conference on computer vision (ECCV), Amsterdam, Netherlands, pp 740–756
- Kuznetsov Y, Stuckler J, Leibe B (2017) Semi-supervised deep learning for monocular depth map prediction. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, USA, pp 2215–2223
- Godard C, Mac Aodha O, Brostow GJ (2017) Unsupervised monocular depth estimation with left-right consistency. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, USA, pp 6602–6611
- Tosi F, Aleotti F, Poggi M, et al (2019) Learning monocular depth estimation infusing traditional stereo knowledge. In: 2019 IEEE conference on computer vision and pattern recognition (CVPR), Long Beach, USA
- He K, Gkioxari G, Dollar P, et al (2017) Mask R-CNN. In: 2017 IEEE international conference on computer vision (ICCV), Venice, Italy, pp 2980–2988

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.