Monocular Visual Odometry Based on Homogeneous SURF Feature Points

Zengxiu Si*, Xinhua Wu, Gang Liu

Computer Science Technology,Wuhan University of Technology, Peace Avenue ,Wuhan, China
1192462302@qq.com
*Corresponding author

Keywords: Visual odometry, SURF feature point, Homogenization, Feature matching

Abstract. In this paper, a monocular visual odometry method based on the uniformized SURF feature point is proposed in view of the accurate real-time location problem of the robot when moving on a flat road. By using the quadtree structure to segment the image, the feature points are evenly distributed in the image under the condition that the total number of feature points is constant. For the feature point matching, this paper builds a straight slope model based on the KNN algorithm, and uses RANSAC to quickly select the correct matching result, which further improves the accuracy of feature point matching. The experimental results show that the method of this paper has high accuracy while ensuring realtime performance.

1. Introduction

In recent years, the visual odometry has become an important choice for autonomous localization of mobile robots [1]. Compared with the traditional positioning method, visual odometry is not affected by wheel sideslip, not affected by electromagnetic field and occlusion effect, what's more, the picture can be used to analyze the surrounding scene, road barrier, and so on, so the visual odometry can be used in many traditional positioning methods do not apply the scene, has been widespread concern [2].

Visual-based odometry are divided into monocular and binocular. Nister et al. Achieved a binocular visual odometer in 2004 [3], and their implementation of the Harris operator feature matching based on the odometer has a high accuracy. Nister et al. First achieved a visual odometer based on binocular vision in 2004 [3], and their implementation based on the Harris feature matching algorithm has a high accuracy. Compared with binocular vision odometry, monocular visual odometry has a simpler system structure and lower algorithm complexity. In 2005, Campbell et al. Proposed a method based on monocular visual odometry ([4]), which uses the optical flow method to estimate the pose of the camera. However, their algorithm lacks a general method to partition the environment. Lovegrove et al. Realized the dense pose estimation algorithm [5] based on the homography and the ground information. But the adaptability of their algorithm is not very well. Pretto et al using SIFT feature point to realize the visual odometry [6], although the SIFT feature has good performance for image rotation and translation, but the computational complexity

and long processing time, seriously affecting the real-time performance of the algorithm. Compared with the high complexity SIFT feature points, the SURF feature points have better performance and faster computing speed, which is more suitable for the visual odometry[7].

In the process of extracting feature points,we often find that the feature points are often concentrated in a few local regions where scene texture is rich,and the other rest regions often have no feature points. This will reduces the robustness of the algorithm. In response to this phenomenon,this paper presents a method of homogenizing the SURF feature points. In addition,we also optimized the process of feature points matching . By analyzing and comparing the advantages and disadvantages of the existing feature point matching optimization methods, based on the relatively high real-time KNN algorithm ,we use a fast simple RANSAC model to filter matched feature points pairs.The number of error matching points are greatly reduced, meanwhile the real-time performance of the algorithm is not affected, and the precision of the algorithm is increased. Finally, we verify the feasibility of this algorithm by by experiments.

2. Visual odometry

2.1. Homogenize the surf feature points

SURF(Speed Up Robust Features) feature point method is an improved method of SITF feature points proposed by Herbert in 2006. The SURF feature point has an similar performance as SIFT, but its computational efficiency is an order of magnitude higher than SIFT [8]. SURF feature points have good scale invariance, rotation invariance and good time efficiency. Therefore, it is widely used in the field of computer vision.

The core of SURF features is the Hessen matrix, Its main idea is to extract the stable interesting point by calculating Hessen matrix determinant.

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \tag{1}$$

SURF uses the Hessian determinant response value of pixels to represent the strength of the feature points, the greater the value, the stronger the feature points. In practical use, we usually set a threshold T for the Hessian determinant response , only when the response value of the Hessian determinant of the pixel is greater than the threshold value of T, the pixel is selected as an feature point. So the threshold T directly determines the number of feature points extracted from an image. The larger the T we set, the stronger those selected feature points are, and the smaller the number of feature points; On the contrary, the smaller the threshold T we set, the more the number of feature points, while some of the points whose performance is not very strong also will be selected as the feature points. The feature points are extracted for feature matching, and then used to compute the camera motion. If the number of feature points is too small, the accuracy of the subsequent computation will be affected; Of course, the number of feature points is not the more the better, if the number of feature points is too much ,although the calculation accuracy can be ensured, but the computational complexity is also improved, so it is necessary to control the number of feature points in a certain range of number N.

Refer to the method of controlling the number of feature points in ORB algorithm, when the number of feature points is greater than the upper bound N, all the feature points are sorted according to the Hessian determinant response value of the feature points, keep the top N feature points in order ,and delete the remaining feature points. The result of this method ensures that the number of feature points will not be too much, but this often lead to the distribution of feature

points concentrated in some rich texture local areas, While the other areas of the image have no feature point. In other words, the distribution of feature points is not uniform. This will results in two bad effects on subsequent calculations: First, the focus of the feature points will lead to feature points in the image position is also very close, while the descriptor of the feature points is generated according to the information of the pixels around the feature points, this will leads to their descriptors are also relatively similar, so the probability of false feature points matching will increase; The second is when the robot movement is too fast or turning a corner, the image scene will have a large change, at this time, the overlapping portions between adjacent frames are relatively small, if the feature points are concentrated, it is easy to cause the non matching feature points, and then can not calculate the camera's pose. This reduces the robustness of visual odometry.

In this paper, we use the idea of four tree to process the feature points. In this paper, we use the idea of quadratic tree to deal with the homogenization of feature points. The main process of the algorithm is described as below:

First calculate the total number of feature points can be extracted according to the Hessian matrix determinant threshold T, if the total number of feature points is not greater than the threshold value of N, not processed and the algorithm ends;

Divide the image into four regions, corresponding to the four branch nodes of the quadtree, if there is no feature in a region then does not retain its branch node;

Count the number of leaf nodes, if it is greater than N, the end of division, go to (5); otherwise, go to (4);

Continuously carry out (2) (3) operation on the leaf nodes of the quadtree; If there is only one feature point in a region corresponding to a leaf node, the leaf node is no longer divided, it becomes a final leaf node;

Count the number of feature points in the image area corresponding to each leaf node, and if it is 1,then the leaf node will not be processed. Otherwise, the feature points in this region are sorted according to their Hessian determinant response values, save those feature points whose response value is maximum and remove the rest.

2.2. Feature points matching optimization

After the feature points are extracted, the feature descriptors are obtained by calculating the Harr wavelet response of the pixel regions around the feature points. There are two commonly used methods: BF(Brute Force Match) and FLANN (Fast Approximate Nearest Neighbor Search Library). BF method calculate match points through the exhaustive way, so its efficiency is very low. FLANN method can improve the search efficiency by the K-d tree partition of the feature descriptor, so we use the matching method based on FLANN.

After the feature matching method carried out,the result always contain a lot of error matching feature point pairs, If directly calculate the motion estimation on the basis of the above result, it will bring great error, so we need to optimize the matching feature points. There are four kinds of commonly used optimization methods [9]: The first is cross matching, after the matching calculation, the feature points A in the first image and feature point B in the second images are matched, performing a reverse calculation by matching the first image with the second image, if the feature point who matched with B is A, then save the matched feature points A and B, otherwise, delete them. The second method is KNN method, In the feature matching calculation, for feature point A reverse two nearest feature points B, c, only when the ratio of the distance between A to B and C reach a certain threshold , then they are matched. In other words, the distance from a feature point B to A is much less than the distance from other feature points to A, then A ,B are real matched. The third and fourth methods are similar, both of them use the RANSAC method to filter

matched pairs, just the filtered model they used is different. One use the homography matrix model, the other is to use the basic matrix model. Through experiments we find that KNN method is superior to the cross-matching method in terms of time efficiency and accuracy. The third and fourth method is more accurate than KNN method, but their algorithm are more complex and take much more time compared to KNN method; Therefore, based on the KNN method, this paper proposes a fast RANSAC method, the accuracy of feature point matching is improved while maintain the time efficiency of the algorithm.

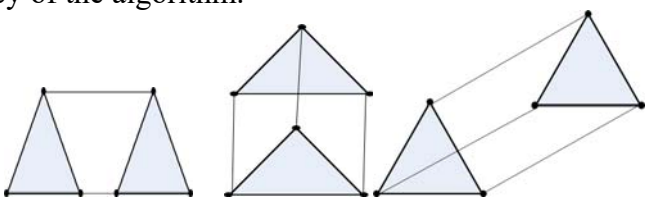

Figure 1 Horizontal movement, Vertical movement, Horizontal +Vertical movement

In this paper, a fast matching optimization scheme is proposed for the motion of a flat road with small rotation amplitude. From the above three pictures we can see, when the image scene does not rotate, the straight line between matching points and the matching points are parallel. The slope of these lines is the same; When a small amplitude change occurs, these lines are no longer parallel, but their slope is still very similar, their slope keep in a very small changed range. The calculation of the slope is very simple, only need one matched feature pair the calculation can be done:

$$k = \frac{y_2 - y_1}{x_2 - x_1} \tag{2}$$

After the KNN method has obtained the preliminary matched result, establish a slope model between matched feature points pair, the RANSAC method is used to iterate to further filter the matched feature point pairs. The calculation process is as follows:

Set a slope change threshold s and the number of iterations N;

A matched feature point pair is chosen randomly as the initial model, and the slope of the corresponding line is k. The acceptable range of the model can be expressed as [k-s, k+s];

Sequentially calculating whether the slope of the other feature point pair falls within the model range in step (2); If falls in this range, the feature point pair is regarded as an inliner point, or it is regarded as an outline point. Record the number of points within the model.

Perform the (2) (3) step N times, and the model with the largest number of inliner points will be regard as the final model. The inliner points of the final model will be regard as the last matched pairs of feature points.

The model is simple and the calculation speed is very fast. The experimental results show that the slope RANSAC algorithm's time-consuming in 20~40ms, and this will not affect the timeliness of the whole algorithm.

2.3. Motion estimation

After the feature point matching, we calculate the relative motion of the camera between two images by the homography. In computer vision, the plane homography refers to the mapping relationship between two planes, It can be easily understood that every pixel in the image taken by the camera is mapped to the points in the scene. This relationship includes the relationship between the camera imaging plane and the relative position of the object plane. Specifically described by the following formula:

$$p=sH \tag{3}$$

P represents a point in the scene space, p denotes the pixel corresponding to P after spatial point imaging, H is a homography matrix, s is scale factor. The matched feature points are actually the same points in the space in different images, This relationship can be described by the following diagram:

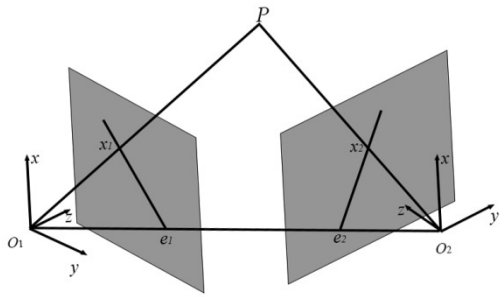

Figure 2 The relationship of one spatial point imaging in two different images

As shown in above FIG, the projection point of the same spatial point p in the two frames is x1, x2, They all have a homography relationship with the spatial point P, so we can get two of the above formula, combine these two formulas, get a new formula:

$$x_1=sHx_2 \tag{4}$$

Homography matrix is 3x3 matrix, there are eight unknown parameters after homogeneous:

$$H=\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{31} & 1 \end{bmatrix} \tag{5}$$

The image of a point in the two frame image is described as the following equation:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = s \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{6}$$

There are 9 unknown parameters in the formula, so it is necessary to have at least four feature points to solve the homography matrix H. The feature points are more than 4 after the feature points matching, The optimal solution is obtained by using the RANSAC method. The homography matrix H contains all the information about the rotation and translation of the camera in two frames, After H is obtained, the rotation R and the translation T are separated from it.

3. Experiment

The experimental data are provided by KITTI[10]. This experiment is implemented in vs2013 environment using opencv3.1.0 coding; The program puts the rotation and translation of each frame into a text, and then use matlab to achieve the drawing of the road map. All of the work in this lab is in a Windows 7 system notebook, Notebook configuration Intel Core i3 2.40GHz CPU and 512M graphics card.

The following experiment sets the maximum number of feature points to 500, The following left picture shows the result of the top 500 feature points based on the Hessian determinant response

sorted after the SURF feature points are extracted, The right picture is the SURF algorithm after homogenization, They are the same picture of the feature point extraction operation. The two pictures below can be seen through this algorithm can make the feature points effectively distributed evenly in the whole image.

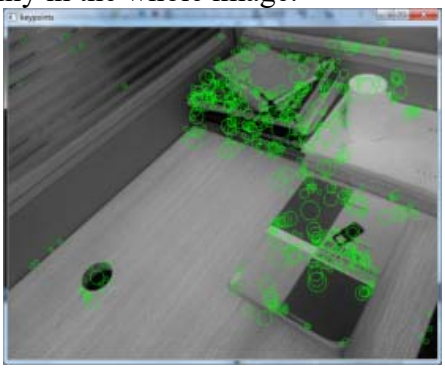

Figure 3 before improvement

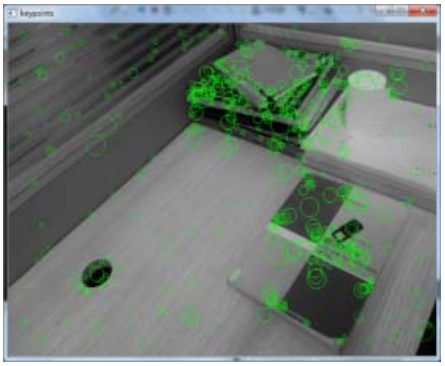

Figure 4 after improvement

Here are the commonly used feature point matching optimization algorithm comparison. The result of the cross-matching filtering method contains a lot of false matches; Knn method is better than the cross filter method, but there are still a lot of error matching; The results of the RANSAC filtering method using the single matrix should be significantly improved relative to the former two; Our algorithm is almost no wrong match, The results obtained in these methods are the most ideal.

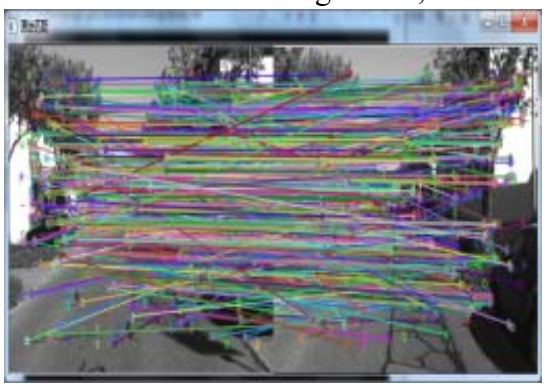

Figure 5 cross filter

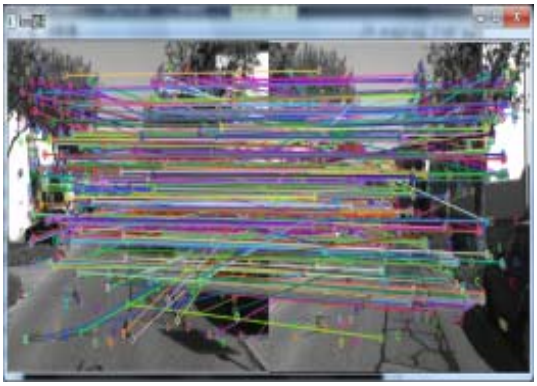

Figure 6 KNN filter

Figure 7 homography filter

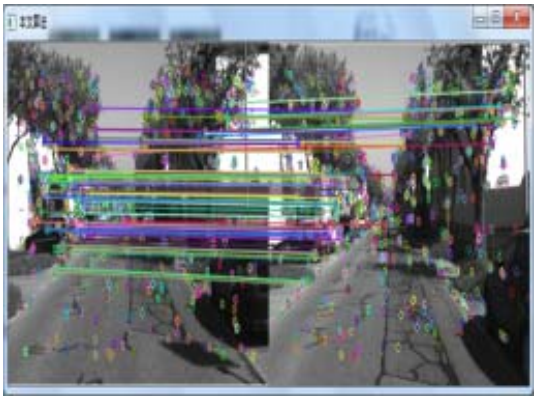

Figure 8 our algorithm

Through the first 500 photos of the data sets, respectively carry out the time-consuming test of various algorithms, then take the average time. The KNN method takes the least time. In this paper,

the RANSAC filter is used based on the KNN algorithm, the time consuming is similar to the cross matching method, but it is far below the homography matrix method.

Table 1 Algorithm time consuming comparison

Algorithm	cross filter	KNN filter	homography filter	Our algorithm
Time consuming（ms）	154.71	133.26	443.71	152.43

Through the above two experiments, it can be verified that compared with the other commonly used methods, our method has the highest accuracy and lowest time-consuming.

Finally, experiments are carried out by using the original SURF algorithm and the improved algorithm. The movement starts from the coordinate (0,0), the entire movement is in the smooth road alley. The following graphs and tables are the comparison of the results of the two algorithms and the average error of the positioning, It can be seen that this algorithm really can improve the accuracy of odometry:

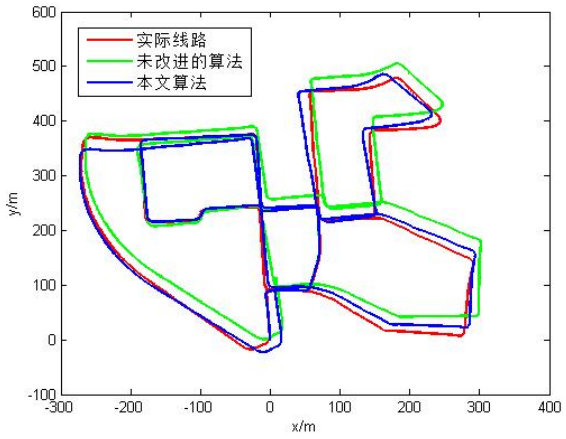

Figure 9 Odometry route compare

Table 2 Average relative error comparison

Algorithm	Before improvement	After improvement
Average relative error	1.85%	1.31%

4. Conclusions

In this paper, we homogenized the feature points and improved the feature point matching method ,improved the robustness and accuracy of the odometry while contain the efficiency. The experiments proved that this method is effective, on the flat road movement can achieve good results.

Acknowledgment

National Natural Science Foundation of China (51179146, China) Abstract: In this paper, the characteristics of river water fog and the video enhancement mechanism of maritime surveillance are studied.

Reference

[1] Wei Lu.Research on the key technology of high-precision real-time vision location [D]., Zhejiang University, 2015 [2] Juan Li. Research on SLAM problem of mobile robot with monocular vision [D]. Harbin Institute of Technology, 2013
[3]Nister D, Naroditsky O, Bergen J. Visual odometry[C]. Proceeding of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004: 652-659.
[4] Campbell J, Sukthankar R, Nourbakhsh I, et al.. A robust visual odometry and precipice detection system using consumer grade monocular vision[C]. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005: 3421-3427.
[5] Lovegrove S, Davison A J, Ibanez-Guzman J. Accurate visual odometry from a rear parking camera[C]. IEEE Intelligent Vehicles Symposium, 2011: 788-793.
[6]Pretto A, Menegatti E, Bennewitz M, et al. A visual odometry framework robust to motion blur[C]// in Proc. of the IEEE International Conference on Robotics & Automation (ICRA. 2009:2250-2257.
[7] Chunbao Suo,Dongqing Yang,Yunpeng Liu.Comparing SIFT, SURF, BRISK, ORB and FREAK in some different Perspectives, Beijing mapping, [J]., 2014 (4): 23-26.
[8] Bay H, Tuytelaars T, Gool L V. SURF: Speeded Up Robust Features[J]. Computer Vision & Image Understanding, 2006, 110(3):404-417.
[9] Dongyun Xu. Study of visual simultaneous localization and mapping based on Kinect[D]. Kinect Harbin Institute of Technology, 2016
[10] http://www.cvlibs.net/datasets/kitti/.