International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: 6.614
Available Online at www.journalijcar.org
Volume 7; Issue 8(F); August 2018; Page No. 14930-14934
DOI: http://dx.doi.org/10.24327/ijcar.2018.14934.2726

ResearchArticle

A STATIC FEATURE POINTS DETECTION ALGORITHM FOR VISUAL ODOMETRY USING OPTICAL FLOW VECTORS

Wenyan Ci., Mingxiang Zhu., Aiyu Dou and Jue Wang

School of Electric Power Engineering, Nanjing Normal University Taizhou Colledge, Taizhou, China

A R T I C L E I N F O

Article History:

Received 5th May, 2018
Received in revised form 24th June, 2018 Accepted 20th July, 2018
Published online 28th August, 2018

Key words:

Visual odometry, Dynamic scenes, Optical flow, Static feature points, Scatteredness.

A B S T R A C T

This paper proposes a robust and precise method for detection the static feature points in dynamic scenes. We use the optical flow vectors as the basis for distinguishing the static features from the moving ones. This method is suitable for the visual odometry systems of vehicles and robots. The static point is selected by a background point set determination process including motion clustering and motion recognition. The motion clustering separates moving objects from background according to optical flow orientation, and the motion recognition determines a background cluster according to scatteredness in image coordinates. The approach presented here is tested on substantial videos and the results prove the robustness and precision of the method.

INTRODUCTION

Visual odometry is a prerequisite for applications like autonomous navigation and obstacle detection, therefore acts as a key component for autonomous vehicles and robotics (Ligorio and Sabatini, 2013). The cruxe of visual odometry is the feature points selected for the calculation must be suitable. The majority of visual odometry remove moving points either by imposing additional constraints like (Badino, 2007, Persson *et al.*, 2015, Mur-Artal and Tardós, 2017, Wu *et al.*, 2017) or by determining the degree of deviation to the model like (Kitt *et al.*, 2010, Bellavia *et al.*, 2013, Deigmoeller and Eggert, 2016). In fact, the point set obtained by these methods unavoidably contains a certain number of moving points. To achieve true static feature points, Musleh*et al*. (Musleh *et al.*, 2012) proposed that only feature points on the ground surface were selected for ego-motion estimation. However, this method is comparatively poor in adaptability to environment because a plenty number of effective ground feature points is hard to be extracted in some situations. Furthermore, the ego-motion parameters obtained by the method only have 4 degrees of freedom. He *et al*. (He *et al.*, 2015) used visual and inertial sensors to deal with complex dynamic scenes and improved the robustness of motion estimation. The robustness of motion estimation was improved since they proposed a visual sanity check mechanism by comparing visually estimated rotation with measured rotation by a gyroscope. However, this method is not a pure vision-based method because of the introduction of other sensors.

Corresponding author:* **Wenyan Ci
School of Electric Power Engineering, Nanjing Normal University Taizhou Colledge, Taizhou, China

This paper aims to detect the static feature points in dynamic scenes. The main contributions of this work can be summarized as follows: This paper suggests an algorithm that can determine the background point set (BPS) according to the difference of optical flow vectors between background points and moving points without any prior knowledge. This method also provides a new idea for the analysis of visual problem in the dynamic scenes.

PROPOSED METHOD

Overview of the Approach

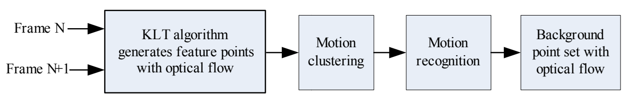


Figure 1 The flowchart of the approach.

Figure 1 gives an overview of the approach. The optical flow of the point set is obtained from KLT algorithm (Shi and Tomasi, 1994, Jun-Sik *et al.*, 2009). The BPS determination process begins with the KLT algorithm which detects Shi-Tomasiconner points and tracks these features using Kanade-Lucas optical flow algorithm. The feature points generated by KLT contain both background (static) and moving points. Two steps including motion clustering and motion recognition are used to determine background points.

Motion clustering is to separate moving objects from the background according to their motion. In consideration that the background points differ from the moving points in terms of their motion relative to the camera and the optical flow is an indication of the difference in image, we consider clustering the background points and moving points according to their optical flow vectors. The optical flow vectors contain two components: magnitude and orientation. Optical flow magnitude is not only related to the motion but also greatly affected by 3D position while the position has a relatively slow and continuous effect on optical flow orientation. Furthermore, the moving direction of a moving object is independent of surrounding background, which leads to a significant difference between optical flow orientations of moving objects and the surrounding background. Thus, we first cluster the feature points according to the optical flow orientation, and then use the optical flow magnitude as a constraint to remove outliers.

Motion recognition is to determine a background cluster from the point sets clustered by the motion clustering. This has been done according to the scatteredness in image coordinates of each cluster. Normally, a moving cluster is a continuous entity with a small scatteredness while a background cluster can spread in a wide region within the image, thus has a large scatteredness. We select the one with the largest scatteredness as background points set.

Motion Clustering

Motion clustering is performed according to the optical flow orientation with the optical flow magnitude as a constraint to remove outliers. The magnitude L and orientation θ of an optical flow vector are computed as follows:

$$L = \sqrt{(\Delta\alpha)^2 + (\Delta\beta)^2} \tag{1}$$

$$\theta = \begin{cases} \arctan(\dfrac{\Delta\beta}{\Delta\alpha}) & \Delta\alpha > 0 \,\&\&\, \Delta\beta \geq 0 \\ \arctan(\dfrac{\Delta\beta}{\Delta\alpha}) + 2\pi & \Delta\alpha > 0 \,\&\&\, \Delta\beta < 0 \\ \arctan(\dfrac{\Delta\beta}{\Delta\alpha}) + \pi & \Delta\alpha < 0 \\ \dfrac{\pi}{2} & \Delta\alpha = 0 \,\&\&\, \Delta\beta > 0 \\ \dfrac{3\pi}{2} & \Delta\alpha = 0 \,\&\&\, \Delta\beta < 0 \\ 0 & \Delta\alpha = 0 \,\&\&\, \Delta\beta = 0 \end{cases} \tag{2}$$

where α and β is real component and imaginary component of optical flow vector respectively.

A self-adaptive K-means algorithm is used to cluster the optical flow orientation of the feature points. For data set $X = \{x_1, x_2, ..., x_n\}$, assuming that the clustering number is K, and the initial clustering center is $V = \{v_1, v_2, ..., v_k\}$, and the optimization precision is ε. The clustering process of the traditional K-means algorithm (Hartigan, 1979, Mohd *et al.*, 2017) is as below:

1.The sample x_i is assigned to the cluster whose clustering center with the smallest distance to x_i by setting cluster label as:

$$T_i = \arg_k \min(x_i, v_k) \tag{3}$$

2.The new clustering center is computed to replace the previous one.

3.Compute the Following Criterion Function

$$SSE^t = \sum_{i=1}^{k} \sum_{j=1}^{n_i} d(x_j, v_i) \tag{4}$$

where t is the iteration number, and n_i is the sample number of cluster i.

4.If $\left| SSE^{t-1} - SSE^t \right| < \varepsilon$, end the loop and output V and T. Else $t = t + 1$, go to step 2).

This traditional K-means clustering method has two limitations: 1) It requires the user to specify the clustering number K, which is not applicable in many cases. In this work, it is not suitable to adopt the same K value for all frames since optical flow orientation varies. 2) It randomly selects initial clustering center, which may give an incorrect clustering result and prolong the iteration. In order to overcome these shortcomings, we introduce a self-adaptive mechanism into the K-means method so that an optimal clustering number and initial clustering center can be determined.

In this work, we select initial clustering centerbased on max-min distance method. The initial center of the method is relatively far from each other, which can avoid the influence of the initial center too dense. When the clustering number is $K_{\min}$, the max-min distance method is used to select $K_{\min}$ samples as the initial clustering center. Each additional a clustering number increases an initial clustering center in accordance with the principle of the max-min distance. Because the previous initial clustering center is unchanged, the clustering result is stable. The specific setting method is as below:

1.Select the nearest one to sample means as the first initial clustering center v_1.

2.When $K=2$, select the farthest one to v_1 as the second initial clustering center v_2.

3.When $3 \leq K \leq K_{\max}$, calculate the distance from samples that are not clustering center to every initial clustering center. d_{ij} represents the distance from the ith sample to the jth initial clustering center. Assuming $D_m = \max\left\{\min(d_{i1}, d_{i2}, ..., d_{i(K-1)})\right\}$, select the mth sample as Kth initial clustering center.

If the clustering number is too large, it will cause the number of background points obtained is too small. So we set upper limit $K_{\max} = 6$ and lower limit $K_{\min} = 2$. Clustering number setting method is as follows:

1.For $K = 2:6$

a. Initialize K initial clustering centers according to the clustering center algorithm above;

b. Update clustering label T and clustering center V Using K-means algorithm;

c. Check termination condition. If the condition is not met, go to b);

d. Compute $CH^{(+)}$ index using clustering result. Go to 1).

2. Take the K value corresponding to maximum $CH^{(+)}$ index as the optimal clustering number K_{opt}.

3. Output the optimal clustering result and validity index.

It's important to note that the difference between optical flow orientations should not greater than π. In this paper, the distance d_{xy} between sample x and sample y is defined as follows:

$$d_{xy} = \begin{cases} |x-y| & |x-y| \le \pi \\ \big||x-y|-2\pi\big| & |x-y| > \pi \end{cases} \tag{5}$$

Because the optical flow magnitude of a moving point significantly differ from the one of a static point, optical flow magnitude is used as a constraint to exclude the potential moving points in the background point set. The points whose optical flow magnitude is seriously off the average are discarded.

Motion Recognition

The motion recognition is performed according to the scatteredness in image coordinates of each cluster. Normally, a moving cluster is a continuous entity with a small scatteredness while a background cluster can spread in a wide region within the image, thus has a large scatteredness. We select the one with the largest scatteredness as background point set. In this study, the volume of the hyperellipsoid of a point cluster is used to measure the scatteredness. The distribution of a point cluster can be represented as a bivariate normal density function as follows:

$$p(x) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left[-(1/2)(x-\mu)`\Sigma^{-1}(x-\mu)\right] \tag{6}$$

where μ is the mean value, Σ is the 2×2 covariance matrix and $(x-\mu)`\Sigma^{-1}(x-\mu)$ is the squared Mahalanobis distance from x to μ.

The points with the same density probability have the same Mahalanobis distance, and are located on the same hyperellipsoid. Using the covariance matrix and Mahalanobis distance, the average volume of the hyperellipsoids of a cluster is defined as follows:

$$V_{avg} = \frac{1}{n}\sum_{i=1}^{n} \pi \cdot |\Sigma(x_i)|^{1/2} \cdot m(x_i)^2 \tag{7}$$

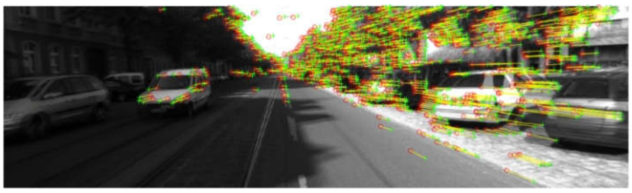
where n is the features number of the cluster and $\Sigma(x_i)$ is the covariance matrix of x_i and $m(x_i)^2$ is the squared Mahalanobis distance from x_i to μ. The eigenvectors and eigenvalues of the covariance matrix Σ determine the shape of the hyperellipsoid while the Mahalanobis distance m determines the radius of the hyperellipsoid. The volume of the hyperellipsoid indicates the scatteredness of the point cluster. The cluster with the greatest V_{avg} is selected as the background cluster in this work.

BPS Algorithm Validation

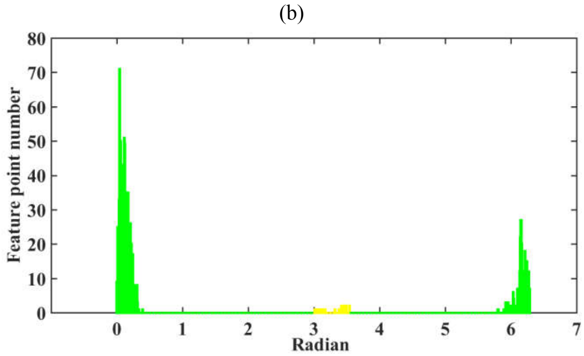
Experiments have been conducted on the public database KITTI (Karlsruhe Institute Technology and Toyota Technological Institute) (Geiger *et al.*, 2012, Geiger *et al.*, 2013). Two typical traffic scenarios as shown in Figure 2(a) and 4(a) are selected as examples to evaluate the approach. The image resolution is 1241×376 pixels. The first scenario involves two oncoming cars and background like parked cars, buildings and trees, where the equipped vehicle moves in longitudinal direction. In the second scenario, the vehicle is turning right in a bend and the moving objects are two pedestrians and one car.

The BPS extraction process for scenario 1 is shown in Figure 2. Figure 2(a) shows left image of frame 1406 with feature points extracted by the KLT algorithm. Figure 2(b) isthe optical flow distribution of the detected feature points between frame 1406 and 1407. The positions of the feature points in frame 1406 and 1407 are marked with red "∘" and green "+", respectively. Figure 2(c) is the histogram of the optical flow orientation. The horizontal axis shows the span of optical flow orientation, i.e., $0-2\pi$ radians, and the vertical axis represents the number of feature points within each interval. It can be seen that the optical flow orientation is mainly located in two regions: 5.81-0.40 radian and 3.02-3.55 radian. Figure 2(d) shows the clustering result clustered by the self-adaptive K-means algorithm. It can be seen that feature points are divided into 2 clusters which are represented with the green and yellow. The green cluster is the background cluster while the yellow cluster is the foreground cluster. The clustering result after imposing the optical flow magnitude constraint is shown in Figure 2(e), which removes the points whose optical flow magnitude seriously deviate from the average.

(a)



(b)



(c)

(d)

(e)

Figure 2 BPS extraction process for scenario 1. (a) Left image of frame 1406 with feature points extracted by the KLT algorithm. (b) Optical flow distribution of the feature points between frame 1406 and 1407. (c) Histogram of optical flow orientation. (d) Clustering result clustered by the self-adaptive K-means algorithm. (e) Clustering result after imposing the optical flow magnitude constraint.

The number of feature points and scatteredness of each cluster are shown in Table 1. The green cluster is selected as the background cluster because it has a greater scatteredness.

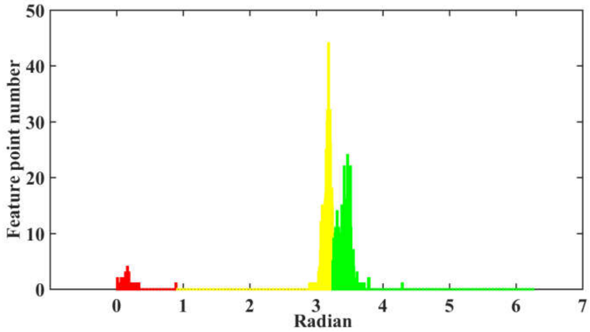
Table 1 Feature point number and scatteredness for scenario 1.

	Green cluster	Yellow cluster
Feature point number	887	21
Scatteredness	77387	5172

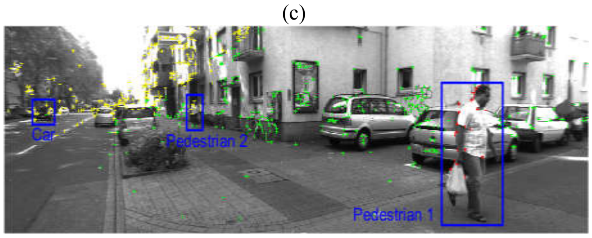
Figure 3 shows the BPS extraction process for scenario 2. Figure 3(a) shows left image of frame 116 with feature points extracted by the KLT algorithm. Figure 3(b) is the optical flow distribution of the detected feature points between frame 116 and 117.As shown in Figure 3(c), the optical flow orientations in scenario 2 are more dispersed than that in scenario 1. Figure 3(d) shows the clustering result clustered by the self-adaptive K-means algorithm. It can be seen that the feature points are divided into 3 clusters represented with green, yellow and red. There is mixed phenomenon. The clustering result after imposing the optical flow magnitudes constraint is shown in Figure 3(e). As the figure shows, all the moving points on the pedestrian 2and car have been removed already.

(a)

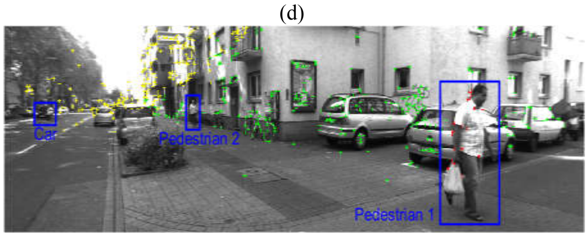
(b)



(c)



(d)



(e)

Figure 3 BPS extraction process for scenario 2. (a) Left image of frame 116 with feature points extracted by the KLT algorithm. (b) Optical flow distribution of the feature points between frame 116 and 117. (c) Histogram of optical flow orientation; (d) Clustering result clustered by the self-adaptive K-means algorithm. (e) Clustering result after imposing the optical flow magnitude constraint.

Table 2 shows the number of feature points and scatteredness of each cluster. The green cluster is selected as the background cluster.

Table 2 Feature points number and scatteredness for scenario 2.

	Green cluster	Yellow cluster	Red cluster
Feature point number	340	333	23
Scatteredness	81685	27955	7014

DISCUSSION AND CONCLUSIONS

This paper presents a method of static feature points detection in dynamic scenes for visual odometry systems. The point set is selected by the BPS determination process which separates moving objects from background according to the optical flow orientation, and determines the background cluster according to the scatteredness. The approach presented here is tested on KITTI database. The experimental results demonstrate that the approach gives a precise detection result. The future work will try to apply the emerging feature detectors and descriptors in the proposed method to further improve the accuracy.

Acknowledgment

This research was supported by Taizhou Science and Technology Support (Social Development) Program (Project No. TS201701).

References

Ligorio, G. and Sabatini, A. M. 2013. Extended Kalman Filter-Based Methods for Pose Estimation Using Visual, Inertial and Magnetic Sensors: Comparative

Analysis and Performance Evaluation, Sensors, 13, 1919-1941.

Badino, H. A Robust Approach for Ego-Motion Estimation Using a Mobile Stereo Platform, in Complex Motion: First International Workshop, IWCM 2004, Günzburg, Germany, October 12-14, 2004. Revised Papers, B. Jähne, R. Mester, E. Barth, and H. Scharr, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 198-208.

Persson, M., Piccini, T., Felsberg, M., and Mester, R. 2015. Robust stereo visual odometry from monocular techniques, in 2015 IEEE Intelligent Vehicles Symposium (IV), 686-691.

Mur-Artal, R. and Tardós, J. D. 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras, IEEE Transactions on Robotics, 33, 1255-1262.

Wu, M., Lam, S., and Srikanthan, T. 2017. A Framework for Fast and Robust Visual Odometry, IEEE Transactions on Intelligent Transportation Systems, 18, 3433-3448.

Kitt, B., Ranft, B., and Lategahn, H. 2010. Detection and tracking of independently moving objects in urban environments, in 13th International IEEE Conference on Intelligent Transportation Systems, 1396-1401.

Bellavia, F., Fanfani, M., Pazzaglia, F., and Colombo, C. Robust Selective Stereo SLAM without Loop Closure and Bundle Adjustment, in Image Analysis and Processing – ICIAP 2013: 17th International Conference, Naples, Italy, September 9-13, 2013. Proceedings, Part I, A. Petrosino, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 462-471.

Deigmoeller, J. and Eggert, J. 2016. Stereo visual odometry without temporal filtering, in 38th German Conference on Pattern Recognition(GCPR 2016), Hannover, Germany, 166-175.

Musleh, B., Martin, D., Escalera, A. d. l., and Armingol, J. M. 2012. Visual ego motion estimation in urban environments based on U-V disparity, in 2012 IEEE Intelligent Vehicles Symposium, Alcala de Henares, Spain, 444-449.

He, H., Li, Y., Guan, Y., and Tan, J. 2015. Wearable Ego-Motion Tracking for Blind Navigation in Indoor Environments, IEEE Transactions on Automation Science and Engineering, 12, 1181-1190.

Shi, J. and Tomasi, C. 1994. Good features to track, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '94, 593-600.

Jun-Sik, K., Myung, H., and Kanade, T. 2009. Realtime affine-photometric KLT feature tracker on GPU in CUDA framework, in 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, 886-893.

Hartigan, J. A. 1979. A K-Means Clustering Algorithm, Appl Stat, 28, 100-108.

Mohd, M. R. S., Herman, S. H., and Sharif, Z. 2017. Application of K-Means clustering in hot spot detection for thermal infrared images, in 2017 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 107-110.

Geiger, A., Lenz, P., and Urtasun, R. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite, in 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 3354-3361.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. 2013. Vision meets robotics: The KITTI dataset, *International Journal of Robotics Research,* 32, 1231-1237.

How to cite this article:

Wenyan Ci *et al* (2018) 'A Static Feature Points Detection Algorithm for Visual Odometry Using Optical Flow Vectors', *International Journal of Current Advanced Research,* 07(8), pp. 14930-14934.
DOI: http://dx.doi.org/10.24327/ijcar.2018.14934.2726

****