International Journal of Modern Agriculture, Volume 10, No.2, 2021
ISSN: 2305-7246

Vision-Based Slam Algorithm For Quadcopter

B Anbarasu[1], Arjun S[2], Kaushik G Raman[3], Manikanta K[4], Stanislaus John[5]

[1,2,3,4,5] Department of Aeronautical Engineering, Hindustan Institute of Technology & Science, Padur-603103, Tamil Nadu, India
E-mail: [1]avianbu@gmail.com

Abstract

This paper presents efficiently building a 3D map of the environment for real-time computation, to define the pose of the agent in real-world settings using a Vision-Based SLAM (Simultaneous Localization and Mapping) Algorithm. This algorithm uses images acquired from cameras and other image sensors. Visual SLAM can use simple cameras (such as wide-angle and spherical cameras), compound eye cameras (stereo and multi-cameras), and RGB-D cameras (depth and ToF cameras). This algorithm measures pictures and allows to fabricate a guide and confine the vehicle continuously. Visual SLAM algorithms also permit the vehicle to map out non-specified environments. This map data is used to bring out performance such as path planning and obstacle avoidance. Intensive testing of four different extractors- SIFT, SURF, SHI-TOMASI, and ORB-was undertaken, to determine the best use for indoor/outdoor environments. The results were verified and have concluded using ORB-SLAM because it is a good alternative to SIFT and SURF in terms of computation cost, matching performance, accuracy, and fast detection of keypoint.

Key words: SLAM, Localize, SIFT, SURF, SHI-TOMASI, ORB

Introduction

In recent years, one of the most popular solutions for collision avoidance & autonomous vehicle tracking has been visual SLAM, where the environment is tracked in all 6 degrees of freedom using Monocular/RGBD Kinect Cameras. Visual sensor aid to generate 3D models of the real world, where the algorithm further closes any tracking information, such that a flight path can be planned & visualized.

VSLAM has a huge scope in the field of robotics & drone, to create low-cost & low-power processing systems, that will make the entire body lightweight & affordable. Data association & computational efficiency are a few of the processes that are challenged in this method, this system has to be structured with no redundancies.

The SLAM algorithm will have to process low-light visuals, with constant vibration [i.e from quadcopter] which will remain a key challenge. VSLAM currently faces computational inefficiencies, being unable to handle heavy video streams, but is being constantly developed by the robotics makers & developers as the computational power requirement is lesser when compared to utilizing extended Kalman filters for the same.

VSLAM batch-processes the footage as per the algorithm that is being run online. The use of cameras can be compared to ultrasonic sensors, whilst cameras provide an abundance of information. Visual SLAM has the goal of calculating the path of the camera traveled while restoring the environment.

Modeling

2.1 Introduction

Carrying out SLAM is a complex task in general, where the estimation of the map and the camera pose is difficult. This problem can be overcome by using datasets or algorithms, where many tasks can be executed parallelly to identify the vehicle pose and estimate the previous set of data for mapping and pose estimation. The pose estimation can be done using various feature extractors such as SIFT, SURF, SHI-TOMASI, and ORB. These feature extractors are a combination of feature point detectors and descriptors. A SLAM algorithm can be used to input data from any kind of sensor to estimate the position of the robot. position, for this project we are using a camera as a sensor to achieve higher accuracy.

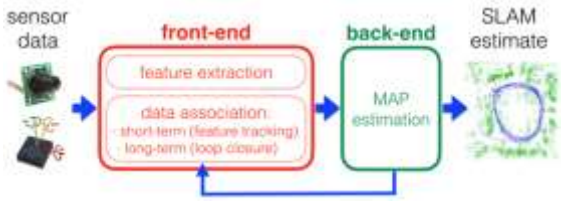


Figure 2.1.1 Front-end & Back-end module for VSLAM

Vision SLAM uses a camera to identify the maximum set of keypoints. For a better outcome, the software is developed in such a way that the quality of the keyframes is improved by enhancing the descriptors and detection algorithms. From Figure 5.1.1, the back-end divisions include the tracking, mapping, and environment building of a real-time visual SLAM system. And the front-end divisions include the feature extraction and the data association (short-term and long-term) for feature tracking and loop closure.

A powerful architecture is required for image processing in VSLAM, and it is done by improvising a software design that eliminates errors. The algorithm is conducted in the feature extractors, and it already exists, so to rectify any drawbacks, the extractors are to be more robust to changes in the environment. We have studied and compared several detectors and descriptors to choose the best amongst them for this project. The evaluation, performance characteristics of all the mentioned extractors are presented in the following sections.

2.2 Performance Measures

Feature extractors- ORB, SHI-TOMASI, SIFT, and SURF were used to extract the dataset of codes and the keyframes were inserted. These extractors are shown distinctive results for the dataset and the images. Each extractor had its own results and these were compared by setting out a random number of feature points.

	COLOR	FEATURE	MAX TIME	MIN TIME	MAX FEATURES	MIN FEATURES
ORB	GRAY	100	0.03891	0	100	100
ORB	GRAY	700	0.05541	0	700	658
ORB	GRAY	1500	0.05896	0.00107	1500	1153
ORB	RGB	100	0.0429	0	100	100
ORB	RGB	700	0.0377	0	700	658
ORB	RGB	1500	0.0709	0.0055	1500	1153
ORB	BGR	100	0.0378	0	100	100
ORB	BGR	700	0.0533	0	700	641
ORB	BGR	1500	0.0468	0.0060	1500	1135
SHI-TOMASI	GRAY	100	0.0050	0	100	100
SHI-TOMASI	GRAY	700	0.0069	0	700	258
SHI-TOMASI	GRAY	1500	0.0089	0	960	258

Figure 2.2.1. Comparison table of ORB and SHI-TOMASI extractors based on their features

The SHI-TOMASI feature extractor was introduced by J.Shi and C.Tomasi. we've used a pre-defined dataset on OpenCV to extract the options of the image by the SHI-TOMASI methodology. Through the function cv2.goodFeaturesToTrack() in OpenCV, the strongest and therefore the sharpest corners are determined from the image. For this methodology, the image ought to be of grayscale, we can specify the number of options we would like and mention the standard level that is between 0-1, any worth below or higher than this vary are going to be rejected. Then the most limited geometric measure between the corner is distinguished. The extractor identifies the corner and so arranges the preceding corners to support their quality. Then the operation takes the primary strongest corner, casts away all the near corners within the limit of the tiniest distance, and returns N strongest corners.

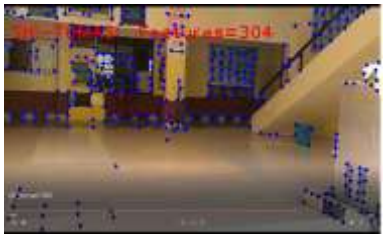
ORB means Oriented Fast and Rotated Brief (ORB). ORB is also performing based on feature detection and is faster compared to other extractors. It is engineered on the FAST keypoint detector and BRIEF descriptor. It uses FAST to identify the key points and then uses the harris corner detection to find the top N point within them. Pyramid is employed for multiscale features. The ORB detector generates a multiscale image pyramid, with variants of images at varied resolutions, where each level in the pyramid carries a subsampled version than the last level. Once this process is finished, the FAST detector is used to distinguish keypoints at each level, and therefore ORB is partial scale-invariant.

FAST does not measure the orientation and is a rotation variant. It measures the intensity weighted centroid of the spot with a located corner at the center. The direction of the vector from this corner point to the centroid gives the familiarization. Moments are estimated to improve the revolution invariance. The descriptor BRIEF inadequately functions if there is an in-plane rotation. In ORB, a rotation matrix is calculated using the introduction of the patch and then the BRIEF descriptors are driven according to the orientation.

(a) SHI-TOMASI Extractor for outdoor

(b) ORB Extractor for outdoor environment

environment



(c) SHI-TOMASI Extractor for indoor environment

(d) ORB Extractor for indoor environment

Figure 2.2.2. Detection of feature points in Indoor/Outdoor Environments

On comparing the images with the SHI-TOMASI and ORB extractors we have tabulated the results based on the nature of the image and the feature points. SHI-TOMASI works very well on grayscale images with very fast detection of feature points, but while considering for RGB or color images ORB has shown a big difference. The images are captured in both indoor and outdoor environments with a defined number of features to be extracted. Feature points are easily detected because the images are very clear and there are no depth errors, so the front-end estimation is done with high accuracy and the pose estimation of the algorithm is performed and the back-end information is also obtained without any failure in the tracking thread.

SIFT stands for Scale Invariant Feature Transform, introduced by David to perform approaches by transforming the image into a large collection of feature points that are being invariant to the change in the robustness of the environment, illumination, easy to extract, highly distinctive, and affine distortions. The major tasks carried out for SIFT extractor are Scale-space detection, Keypoint localization, orientation assignment, and keypoint description. We have used the OpenCV function cv2.drawKeyPoints, which draws small circles on the location of keypoints. These keypoints are extracted from a set of reference images and then stored in the database. Any object is identified in a new image by comparing features of the new image to the database and finding the matching features based on the euclidean distance of their feature vectors.

Key positions are characterized as maxima and minima of the assurance of the variety of Gaussian work applied in scale-space to a progression of smoothed and resampled pictures. The algorithm of SIFT has the following steps. The first is to determine scale-space extrema applying the Difference of Gaussian (DoG). Secondly, a key point localization wherever the keypoints are localized and separated by terminating the below contrast points. Thirdly, a key point adjustment pattern based on local image grade and finally a descriptor dynamo to compute the local image descriptor for every key point based on image grade size and orientation

SURF or Speed up Robust Features is a feature extractor based on the Hessian matrix and sums of 2D Haar wavelet response. They depend on the determinant of the Hessian matrix for choosing both scale and location. It is a patented feature extractor and is inspired by SIFT extractor. SURF is many times faster than SIFT and is more effective against multiple image transitions. A blob detector is being used which is based on the Hessian matrix to find the points of interest.

For introduction assignment, it uses wavelet acknowledgments in both horizontal and vertical directions by applying adequate Gaussian weights. A region around the key point is chosen and arranged into subregions and then for each subregion, the wavelet acknowledgments are considered and represented to get the SURF feature descriptor. The symbol of Laplacian which is previously calculated in the detection is employed for bearing interest points and it distinguishes clear blobs on

dark backgrounds from the reverse case. In the case of matching the features are compared only if they have some type of contrast (based on the sign) which allows faster matching.

(a) SIFT Extractor for outdoor environment

(b) SURF Extractor for outdoor environment

(c) SIFT Extractor for outdoor environment

(d) SURF Extractor for outdoor environment

Figure 2.2.3. Detection of feature points in Indoor/Outdoor Environments

The results of the comparison of feature matching of SIFT and SURF extractors in indoor and outdoor environments were seen. It is visible that SURF detects any feature point which has a blob point and varies it down distinctively whereas SIFT detects any smooth or circular corner as a feature point. The graphical comparison of SIFT and SURF is given in the next section.

2.3 Performance Analysis

Figure 2.3.1. Comparison Graph of ORB and SHI-TOMASI Extractors on grayscale and RGB images

The x-axis denotes the number of frames and the y-axis denotes the number of keypoints being matched. The graph is obtained from the excel sheet which has the tabulated values of the parameters and further graphically represented. The ORB feature can be used for any kind of image but SHI-TOMASI is only used for grayscale images.

ORB shows very little deviation for gray and RGB features for a different set of keypoints, whereas SHI-TOMASI gradually shows variation as the number of features increases with respect to the frames. ORB is also giving a better result because an exact number of points are being matched without any projection error.

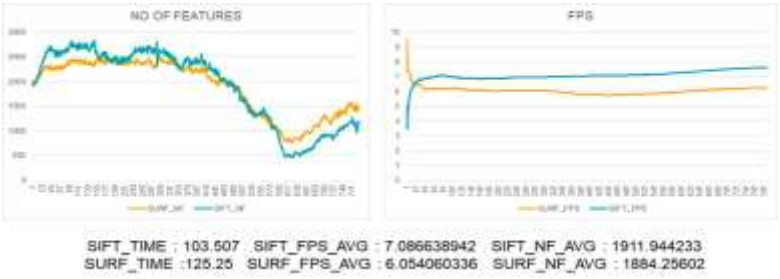


Figure 2.3.2. Feature point detection with SIFT and SURF; FPS matching SIFT and SURF X-axis- Frame number; Y-axis- No: of feature points; X-axis- Frame number; Y-axis-time(s)

From Figure 2.2.2 & 2.2.3., we have obtained the above graphical chart, which shows us the feature detection and fps matching of the images. SURF is faster than SIFT although SIFT provides better results in feature matching at different image conditions. The matching rate of SIFT is also higher and has a good result in the scaling area. While considering illumination, both had the same effect on finding different key points/feature points.

Feature matching is utilized which takes the descriptor of one component in the initial set and is coordinated with any remaining highlights in the subsequent set utilizing some distance computation. What's more, the nearest one is returned. We have experimented with ORB, SIFT, and SURF.

The feature matching technique which we have used is FLANN which stands for Fast Library for Approximate Nearest Neighbor. It is inbuilt with a set of algorithms for searching the nearest neighbors in incomparable datasets. It is the fastest and most accurate matching technique. It consists of two dictionaries, index and search parameters.

We have analyzed FLANN and BF matching techniques using an ORB extractor and the result is shown below. BF matching technique takes the descriptor of each and all the features from the first set and matches with all other features from the second feature using distance evaluation. Though it is time-consuming, it matches all the features at varied periods.

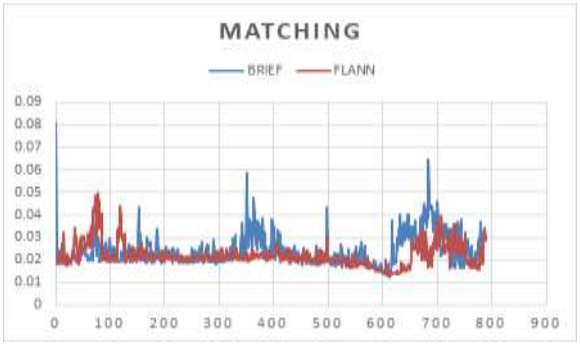


Figure 2.3.3. FLANN vs BF, X-axis: Feature points; Y-axis: Time(s)

(a)

(b)

(c)

Figure 2.3.4 Feature matching using various extractors with 100 features; (a) represents feature matching using FLANN with ORB extractors, (b) with SIFT extractors, and (c) with SURF extractors

2.4 Summary

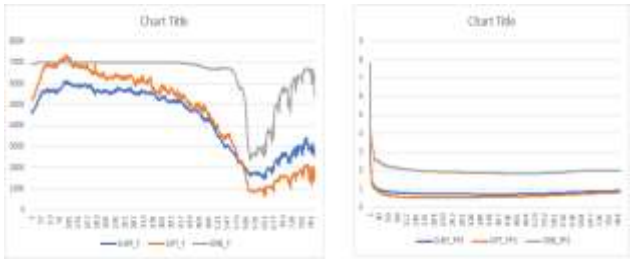


Figure 2.4.1. Feature point detection with SIFT, SURF, and ORB; FPS matching SIFT, SURF, & ORB X-axis- Frame number; Y-axis- No: of feature points; X-axis- Frame number; Y-axis-time(s)

The matching method is a task that must be considered when choosing a detector/descriptor combination, as the algorithm can lose time during this stage, particularly if the algorithm is complex. There are a lot more Key-points in the dataset that need to be matched. This is clearly shown in performances where the advantages and disadvantages of each detector/descriptor combination are discussed. We have compared image matching with four different extraction techniques and compared it on different environments and the parameters have been displayed in the table and graphs.

Design & Development

3.1 Introduction

Shi-Tomasi: OpenCV has a function, cv.goodFeaturesToTrack(). If finds N strongest corners in the image by Shi-Tomasi. The image is grayscale then you specify the number of corners you want to find. Then you specify the quality level, which is between 0-1, which denotes the minimum quality of the corner below which everyone is rejected. At that point, we give the base euclidean distance between corners distinguished. With this data, the capacity discovers corners in the picture. All corners underneath the quality level are dismissed. At that point, it sorts the leftover corners' base quality in slipping requests. At that point, the capacity takes the principal most grounded corner, discards every one of the close by corners in the scope of least distance and most grounded corners. Afterward, Brief takes all keypoints found by the FAST calculation and converts it into a paired element vector with the goal that together they can address an article.

SIFT: SIFT stands for Scale-Invariance Feature Transform.

The working of the SIFT algorithm comprises 5 steps. At first, Scale-Space Extrema Detection is done as follows: Blurring of the images using Gaussian filters. Then the Octave is taken. That pixel will be selected and compared with the 8 pixels around it. These 9 pixels are considered to be an octave. After which 2 more octaves above and below the pixel we considered before is chosen. Then, local extremum is searched for and if found, then it will be considered a potential keypoint. After the extrema detection, Keypoint Localization is done.

Its process majorly does 2 important things. They are removing edges using eigenvalues and ratios and removing the extrema of intensities lesser than the threshold values set. Thus, the removal of low contrast key points and edge keypoints happens in this step. After all these removal processes, Orientation Assignment is done. In this step, orientation will be assigned to the keypoints. With these orientations for the key points being created, an orientation histogram will be derived from the orientations. And when the orientation is more than 80% for a keypoint that keypoint will be taken into consideration for calculating the orientation. After the orientation setup, Keypoint Descriptor is derived. The size of the vector is the number of keypoints * 128.

The algorithm will create a 16*16 neighbor around the keypoint in which each subblock will have 4*4 pixels. Now, each of these pixels will have 8 pixels around it. Thus, 8*4*4 gives 128. After derivation of the description from the orientation Keypoint Matching will be done. It is the process where keypoints are matched between images irrespective of the rotation, contrast, or scaling. This is done by identifying the nearest neighbors and doing ratio analysis between closest and second closest neighbors. Also, if this ratio is greater than 0.8, then the keypoints are rejected. This is done to avoid the closest match of neighbors which happens due to noises.

ORB: Oriented FAST and Rotated BRIEF (ORB) was developed at OpenCV as an efficient and viable alternative to SIFT and SURF. ORB was considered principally because SIFT and SURF are algorithms that hold a patent. ORB's principle commitments are the expansion of a fast and exact direction part to the FAST and effective calculation of situated BRIEF highlights. It utilizes FAST, and afterward, Brief takes all keypoints found by the FAST calculation and converts it into a paired element vector with the goal that together they can address an article.

SURF: SURF(Speeded Up Robust Features) is a robust and fast technique preferred for its fast computation of operators using box filters, thus enabling real-time applications such as tracking and object recognition.Consists of 2 steps, Feature Extraction, and Feature Description. Feature extraction. Feature extraction uses an integral image, which is a way of calculating the average intensity of pixels in a selected box. Surf utilizes the Hessian framework in light of its great exhibition in calculation time and precision. Descriptors are relegated by first allocating a fixed direction to each intrigue point, and afterward separate the component descriptor.

4. Conclusion

Based on Performance Measures and Performance Analysis the following result is obtained:

ORB focuses on close range and does dense extraction thereby leaving out far-off features unnoticed. SIFT and SURF performs similarly in an outdoor environment and are computationally expensive focusing on sparse extraction. ORB is the fastest algorithm while SIFT performs the best in the most scenario Controlling the number of features to be extracted of SIFT and SURF is not possible thereby making it difficult to manage tradeoff in the extraction process. Shi-Tomasi has the least number of features comparatively in an outdoor environment but has the sparsest features extracted. It Clearly defines the boundary of the environment compared to other techniques.

References

1. Strite S and Morkoc H 1992 *J. Vac. Sci. Technol.* B **10** 1237
2. Durrant-Whyte, H.; Bailey, T. (2006). *Simultaneous localization and mapping: part I.* 13(2), 99–110. doi:10.1109/mra.2006.1638022
3. Bailey T, Durrant-Whyte H (2006) *Simultaneous localization and mapping (slam): Part ii*. IEEE Robot Autom Mag 13(3):108–117
4. Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D. Tardos. "*ORB-SLAM: a versatile and accurate monocular SLAM system.*" IEEE Transactions on Robotics 31, no. 5, pp 1147-116, 2015.
5. Raul Mur-Artal, J. M. M. Montiel ´ , Member, IEEE, and Juan D. Tardos´ , Member, IEEE *ORB-SLAM: A Versatile and Accurate Monocular SLAM System*
6. H. Strasdat, J. M. M. Montiel, and A. J. Davison, "*Scale drift-aware large scale monocular SLAM*," presented at the Proc. Robot.: Sci. Syst., Zaragoza, Spain, Jun. 2010.
7. Taketomi et al. *IPSJ Transactions on Computer Vision and Applications* (2017) 9:16 DOI 10.1186/s41074-017-0027-
8. STANISŁAW KONATOWSKI, PIOTR KANIEWSKI, JAN MATUSZEWSKI Military University of Technology, Warsaw, Poland
9. Thrun S, Leonard JJ (2008) Handbook of robotics. Chap. Simultaneous localization and mapping