修士論文概要書

Summary of Master's Thesis

Date of submission: 30th/Jan/2017

専攻名（専門分野）	Department of Computer science and Communications Engineering	氏名 Name	Yifei Zhang	指導教員 Advisor	印 Hiroshi Watanabe Seal
研究指導名 Research guidance	AudioVisual Information Processing	学籍番号 Student ID number	5115FG07-6		
研究題目 Title	Research On Trajectory Visualization Of Ego-motion Videos With Pedestrian Based On Monocular Visual Odometry And Machine Learning				

1. Introduction

Trajectory visualization of ego-motion videos is one of the main building blocks of vision-based robot navigation technology. One solution is visual odometry, and the process of estimating of the agent trajectory using the input of a single monocular camera is called monocular visual odometry (MVO). While most of MVO are geometrical methods, the precision of feature-based MVO system depends on the calculation correctness of features extraction and matching. The moving objects in the ego-motion videos contributes to error propagation. The geometrical methods performances drop drastically in the scene with dynamic background and moving objects. On the other hand, machine learning has outstanding performance in object detection and classification, e.g, Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM) are proved to be efficient in pedestrian detection. In this paper, moving human being is regarded as the main moving target in the process of visual odometry, and the accuracy of MVO is improved by eliminating the pedestrian feature points. A lean camera trajectory visualization system based on featured-based monocular visual odometry and machine learning is presented, and tested by two publicly available KITTI benchmark sequences with ground truth. According to the result of the experiments, improvements in accuracy are shown.

Keywords: **trajectory visualization; monocular visual odometry; feature extractor; machine learning; HOG + SVM;**

2. Related Work

To build the system of MVO, we adopt FAST corner detector [2] and combine it with SURF descriptor to improve the efficiency of whole system. After feature tracking by Kanade-Lucas-Tomasi method, sparse pixel wise correspondences is built. Eight point algorithm conjuncted with RANSAC [3] solve the non-linear equations in eight degrees of freedom with higher accuracy.

3. Experiments

KITTI datasets are captured by a monocular camera fixed on a cruise car. Camera intrinsic parameters are available and frames are undistorted.

4.1 Error metrics

We use Root Mean Square Error (RMSE) to evaluate the performance of the proposed system and simple geometric monocular visual odometry with ground truth in every parameters. Ground truth data includes 12 parameters of the camera positions about rotation and translation. The Table 1 show the result. From Table 1, computed on KITTI 05 dataset, we confirm that pedestrians feature points elimination reduce the error of camera pose estimation and trajectory generation.

Table 1 Performance Comparison of Estimation Parameters

	ML-VO				Geometric VO			
	Rotation			Translation	Rotation			Translation
RMSE	0	0.0007	0.0009	0.0027	0.0002	0.0022	0.004	0.2184
	0	0	0.007	0.0204	0.0022	0	0.0026	0.0187
	0.009	0.001	0	**0.1252**	0.0024	0.0026	0.0002	**0.3081**

4.2 Path generation

The dataset of KITTI 05 sequence include pedestrians, its reconstructed trajectories are shown in Fig 1.

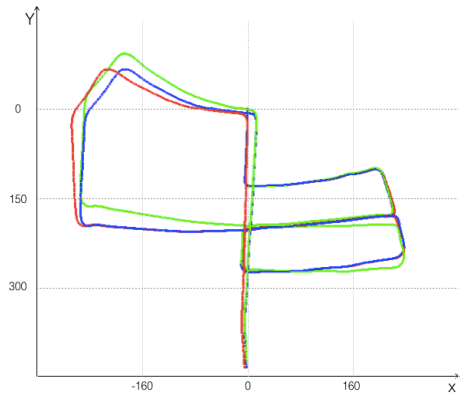


Fig 1: Trajectory computed on the KITTI Seq 05

The red path is ground truth, and Green one is simple MVO, and the blue one is the result of this paper.

References

1. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Trans on CVPR, San Diego, CA, USA, pp. 886-893 vol. 12005.
2. T.Drummond, E.Rosten, "Machine laerning for high speed corner detection," ECCV, Graz, Austria, vol.1, pp.430-443, 2006.
3. M.A.Fischler, R.C.Bolles,"RANSC sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," Communication ACM, vol.24, no.6, pp.381-395, 1981.

Research on Trajectory Visualization of Ego-Motion Videos with Pedestrian Based on Monocular Visual Odometry and Machine Learning

A thesis submitted to the department of Computer Science and Communication Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University
in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering

January 30th, 2017

by
Yifei Zhang
$(5115FG07-6)$ of
Audio-visual Information Processing
Professor Hiroshi Watanabe

Declaration of Authorship

I, Yifei ZHANG, declare that this thesis titled, 'Research on Trajectory Visualization of Ego-Motion Videos with Pedestrian Based on Monocular Visual Odometry and Machine Learning' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Hiroshi Watanabe for the continuous support of my study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, my sincere thanks also goes to Mr. Takaaki Ishikawa who provided me an opportunity to join image processing team.

I thank my fellow labmates in for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last two years. In particular, I am grateful to Mr. Hideaki Yanagisawa for enlightening me the first glance of machine learning.

Abstract

Trajectory visualization of ego-motion videos is one of the main building blocks of vision-based robot navigation technology. One solution is visual odometry, and the process of estimating of the agent trajectory using the input of a single monocular camera is called monocular visual odometry (MVO). While most of MVO are geometrical methods, the precision of feature-based MVO system depends on the calculation correctness of features extraction and matching. The moving objects in the ego-motion videos contributes to error propagation. The geometrical methods' performances drop drastically in the scene with dynamic background and moving objects. On the other hand, machine learning has been showing outstanding performance in object detection and classification, e.g, Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM) are proved to be efficient in pedestrian detection. In order to minimize the impact of moving objects on visual odometer reliability, it is better to be able to detect more types of moving objects. However, human beings are the most important groups of the environment and also are one of the main external disturbances during the robot navigation. In this paper, human being is regarded as the main moving target in the process of visual odometry, and the accuracy of MVO is improved by eliminating the pedestrian feature points. In this research, a lean trajectory visualization system is proposed, and the pipeline which combines featured-based MVO and HOG + SVM method is proposed to eliminate moving pedestrians . According to the result of the experiments, improvements in the accuracy of camera poses estimation are shown.

Keywords: trajectory visualization; monocular visual odometry; feature extractor; machine learning; HOG + SVM;

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

There are more and more intelligent robot agents entering the ordinary people's life recently. While they make our lives more convenience, one of the most important technology to support the robot implementation is Simultaneous Localization and Mapping (SLAM) [1, 2], which means the robot can generate a map of its surrounding environment during simultaneously estimate the motion and the poses of the robot agent, based on the count of wheels turns or the images from on-board cameras. Visual Odometry (VO) is part of SLAM without closing estimation. There are principally three inclinations to collect navigation data from the unknown environment. The first one is based on the counting of turns of the wheels, which is the original method proved effective. Not only the implementation on Mars exploration rovers [3] but also be widely adopted in the automatic sweeping robot for its simple operation and low cost. The second approach is based on distance sensors, for example, laser, radar, and ultrasonic sensors. This approach can directly measure the distance between the robot and obstacles, then generate the map of surroundings. Besides that, with the development of manufacturing technology, more and more cameras are used in the process of estimating the trajectory, including RGB-D cameras, stereo cameras, and monocular cameras [4]. Monocular visual odometry (MVO) is the third method by using the images or videos taken by a single camera as the input source data to generate the odometry. MVO is widely used in robotics application, augmented reality and sports broadcasting system, especially with the wearable camera outbreak in quantity [5].

These three different kinds of methods of building surrounding environment map all have their advantages and disadvantages. For the wheel odometry method, it requires precisely corresponding wheel mathematical model and error propagation frequently occurs after a long distance movement. There are more error when the wheels slip during the movement. For the method adopted distance sensors, the hight price prevent them from spreading. Sensors accumulated error is another drawback. Compared with the method using other sensors (e.g. Stereo, Laser, Radar), MVO adopt monocular cameras as the source data and estimate the pose of the agent and generate the trajectory. It makes low requirement about hardware, but it need to pay hight attention about illumination condition and enough feature points in the images content. However, by using the images taken by cameras, we can get the message about the surrounding environment, including structure, color concentration, image depth and the movement, especially in the GPS-denied environment. Robot vision navigation technology is just based on this belief to realize environment perception and autonomous control. Visual autonomous navigation technology get rapid development with the progress of the computer and camera hardware technology in 1980s [6]. Most of Visual odometry adopt stereo cameras as the input source. For the reason that stereo methods could achieve superior result in odometry reconstruction, they are widely used in the robot navigation [7]. However, monocular visual odometry only needs one single camera, which usually costs much cheaper and compute faster than stereo cameras. The monocular visual odometry pipeline is easier to put into implementation using the common hardware [8]. For these advantages, the monocular visual odometry attracts a lot of attention of research recently. The fellowing Fig 1.1 shows how the KITTI Vision Benchmark data collected [9].

FIGURE 1.1: Different ways to collect input data in Visual Odometry (Image taken from KITTI website [9]).

1.2 Research status

1.2.1 Monocular visual odometry methods

The state-of-the-art Visual Odometry algorithm is mainly based on the feature matching. Compared with direct methods, such as LSD-SLAM [2], feature-based methods are more robust and relatively simple to implement [1]. Exclude loop closure part of feature-based SLAM, feature-based visual odometry extracts features and adopts them to the next step of triangulation and estimation. The feature matching is performed with creative liberties by adjusting appropriate parameters. The system accuracy and robustness are influenced by the featured amount, correctness of feature matching, lighting situations, camera rotation, and outlier correction. The difference of each algorithm depends on their robustness and the ability of outlier rejection. The general pipeline begins with the initialization of the pose of the robot, collect characteristics from the unknown environment, and creates environment 3D map. For the pose graph, estimation is based on the objects features update from the new environment. Whenever feature are lost, the pose initialization is applied. The loop constraints play an important role for improving map accuracy optimization by updating and modifying new pose graph. Just like the Fig 1.2 and Fig 1.3, different with SLAM, Visual Odometry is no need to reconstruct the environment, instead, generate the trajectory of the moving agent based on the robot motion vector. Compared with absolute location methods widely used in SLAM, this method is named with relative location. As the alternative program of the wheels odometry, visual odometry adopts cameras to collect messages from surroundings. The pipeline of visual odometry is leaner and easier to realize. Besides that, the result is not limited by the accuracy of the map.

Monocular visual odometry requires less computer investment and simpler to con-

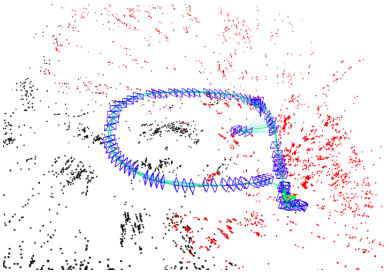


FIGURE 1.2: The SLAM result map. (Image taken from author's ORB-SLAM experiment).

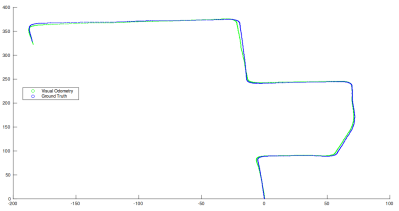


FIGURE 1.3: The VO result map. (Image take from author's MVO experiment [10]).

struct the framework. In addition, there is less image data and cost comparing with stereo visual odometry. In view of the merits of monocular visual odometry, a lot of research have been done about this technology. First, Nister et al. [11] designed and

implemented a monocular and stereoscopic odometer system, and realized the outdoor navigation of visual odometer in the real world, laying an important foundation for the later researches. They construct the odometer mainly through three steps to complete: feature point extraction, feature point matching and motion estimation, which is still the basic theoretical framework visual odometry construction. First, they uses Harris corner extraction algorithm to extract the feature points of the image sequence. Then, in the feature matching part, they use the disparity constraint and the interactive matching algorithm to reduce the error matching. Finally, They uses the five-point method and RANSAC algorithm to estimate camera poses and generate the position of the robot. They conducted extensive experiments in outdoors to verify the reliability of visual odometry. After the work of Nister et al., feature-based methods with a monocular camera over a long distances were presented, where include perspective and omnidirectional cameras [12, 13]. They still adopt five-point basic matrix method to eliminate outlier [14, 15].

1.2.2 Machine learning algorithms in visual odometry

Machine learning technology have developed a lot from 1950s. With the rapid development of Internet and information technology, machine learning has become a hot research topic. It is applied to many fields, such as data mining, natural language processing, search engine and so on. In short, the so-called machine learning is to train a computer to think like human, and use data or experience to make the performance further optimized. Typical machine learning technologies are Logistic Regression, kNN, k-means, decision tree, Nave Bayes, neural network, SVM and so on [16, 17].

In the field of vision navigation, the applications of machine learning algorithm are much less, and the form is relatively simple. A well-known application example is Google's autopilot technology for automatic car driving. In this research, autopilot technology was used to ensure the safety of automatic drive by detecting the distribution of pedestrians within a range of a few hundred meters form the car. Here one machine learning algorithm named pedestrians detection was used. This will be described in detail in the following chapter.

In addition, because that the classification ability of SVM (support vector machine) is excellent, some researchers have applied SVM to "scene recognition" which robot navigation needs. For example, Sangwoo et al. [18] used SVM to separate the feature points, and then use the cloud data as the invariable feature to estimate the motion of the robot. This can be seen as a global absolute positioning method, in which the cloud position is taken as a constant road signs to guide the direction. The disadvantage is that in most cases it is difficult for the robot to find a constant global landmark. In [19] and [20],

the SVM is trained by the environment pictures prepared in different scenes in advance. When the robot moves to the same scene, it can be distinguished by SVM. Next step, the classification helps to achieve the aim of positioning. However, it is troublesome to record the position information for each scene in advance, and the angle judgment of the scene is also prone to problems.

1.3 Main research topic in this paper

In this paper, we focus on the monocular visual odometry based feature-based visual odometry and machine learning, and we propose some methods to improve the accuracy and reliability of visual odometer of the ego-motion videos with pedestrians. The main contents of the paper are as follows:

- Chapter 1: The introduction part introduces the research background of this paper. It mainly introduces the current research situation of visual odometry and machine learning methods in robot navigation. Next, it expounds the advantages and disadvantages of different systems. At last, it introduces the significance and difficulty of this subject.
- Chapter 2: Simple geometric visual odometry system related theory, including pipeline. This chapter introduces the feature detector methods and monocular visual scale ambutation problem. The chapter also details the advantages and disadvantages of common visual estimation method.
- Chapter 3: In this chapter, the human pedestrian detection algorithm based on HOG + SVM is proposed to detect the human beings in the ego-motion videos and improve the visual odometer accuracy by eliminating the feature points of pedestrian.
- Chapter 4: In this chapter, we details the experimental setups and the prepare work of different sequence datasets of KITTI, and give the relevant experimental results to verify the accuracy and reliability of the proposed monocular visual odometry in this paper, which proves the necessity and validity of the innovation in this paper.
- Charpter 5: In this chapter, we summarizes the research work and prospects for the future research.

Chapter 2

Feature-based monocular visual odometry

Monocular visual odometry is a typical application of computer vision technology in the field of robot navigation. Feature-based monocular visual odometry utilize the feature extracted from the monocular images as the material for positioning navigation, just like the human being perceive the environment through images obtained by eyes. The steps of designing a visual odometry are generally similar, mainly collecting images first, then filtering and rectifying images, and then selecting appropriate features to detect and match. Finally, by calculating the motion of image features to estimate the poses of cameras. The odometer frameworks designed by the various researchers are broadly similar and generally differ in detail. This chapter also follows the above basic framework to construct a monocular visual odometry. The basic feature-based MVO pipeline is shown in the following Fig 2.1

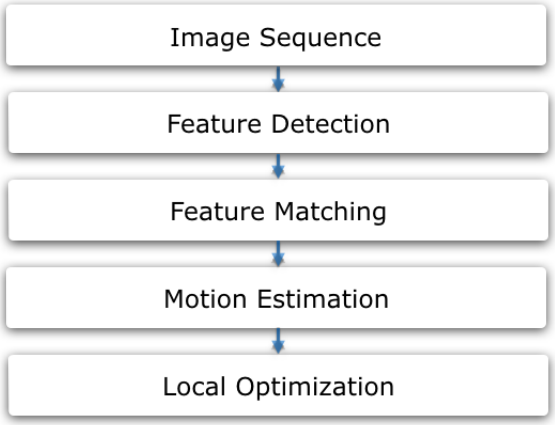


FIGURE 2.1: The simple pipeline of feature-based VO system.

2.1 Image preprocessing

2.1.1 Camera calibration algorithm

There is a mapping relationship between the image captured by cameras and the objects in the 3D world, and the main factor that affects this relationship is the camera parameters. The parameters obtained through camera calibration are the key to link the image and the actual scene, and the key to the accuracy of the odometry. At present, the popular calibration algorithms are Tsai algorithm [21], Zhang Zhengyou algorithm [22] and self-calibration algorithm [23]. As Zhang Zhengyou's board calibration method is simple to and accurate, in this paper, we adopt Zhang Zhengyou's approach as the calibration method.

According to Zhang Zhengyou's method, the points on the same plane can be linked through the internal reference matrix. It proves that the camera internal and external parameters can be calculated based on the images taken from different locations and angles of the same plane. The relationship between the spatial 3D point $M(x, y, z)$ and the corresponding 2D camera plane point $m(u, v, 1)$is as follows:

$$sm = A[R\ t]M \tag{2.1}$$

and for A :

$$A = \begin{bmatrix} f_0 & 0 & u_0 \\ 0 & f_1 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

where s is scaling factor, A is the internal parameters matrix of the camera, R is rotation matrix, t is translation vector, f_0, f_1 are the focal length ranges of the camera, u_0, v_0 represent the primary optical axis of the camera.

The following results can be obtained after simplification:

$$sm = HM \tag{2.3}$$

and where:

$$H = A[r_1\ r_2\ t] \tag{2.4}$$

H is the homography matrix that describes the relationship between three dimensional points of space and the two dimensional camera images. Since Zhang Zhengyou's method uses a chessboard to calibrate the camera, the rotation matrix can be described by r_1 and r_2. As the calibration points are the corners of the chessboard grid which can be known in advance, the camera's two dimensional points can be detected by any corner detection algorithm. In that case, H can be calculated by any single image.

If we describe H in the format of $[h_1\ h_2\ h_3]$, there will be:

$$\begin{cases} h_1 = A \cdot r_1 \\ h_2 = A \cdot r_2 \\ h_3 = A \cdot t \end{cases} \tag{2.5}$$

According to the basic properties of the unit rotation matrix:

$$r_1^T \cdot r_2 = h_1^T \cdot A^{-T} \cdot A^{-1} \cdot h_2 = 0 \tag{2.6}$$

$$r_1^T \cdot r_1 = r_2^T \cdot r_2 \tag{2.7}$$

$$h_1^T \cdot A^{-T} \cdot A^{-1} \cdot h_1 = h_2^T \cdot A^{-T} \cdot A^{-1} \cdot h_2 \tag{2.8}$$

From (2.6), (2.8) H and two equation of A can be calculated, then we can get internal matrix.

2.1.2 Image correction algorithm

For ordinary cameras, especially for wide-angle lens, there are always some distortion in images. Distinctive image have a clear feature that straight lines are shown as curved lines. Obviously, if this kind of images are used directly to calculate the odometer, error will be increased. Therefore, it is necessary to perform image correction before the feature detection.

Gray-scale interpolation is a common image correction method, and its main idea is to calculate the current pixel value by using the value of neighborhood pixels. Bilinear interpolation method selects four most adjacent points from the distort image (x', y'). Their gray values $v(x', y')$ are known, and meet the following relationship:

$$v(x', y') = ax' + by' + cx'y' + d \tag{2.9}$$

Substitute the coordinates and pixel values of the four points into (2.9), and parameters a, b, c, d can be calculated. Then we use (2.9) to recalculate the value of each pixel.

Bilinear interpolation is very simple and easy to implement. More important, its computational requirement is small, so it will not affect the speed of odometer operation. This paper adopt bilinear interpolation method to complete the image correction.

2.2 Direct and indirect methods

Since visual odometry relies on the position change of the object in the image to estimate the actual camera poses, it is important to extract such a reference object. The environment of visual odometry is usually very complex, and specific object detection is difficult to achieve, so the most common method is to extract simple feature instead of object. Typically, these features include block features, corner points, dotted lines, and so on. However, all of these can be regarded as indirect methods, while the directly method is the process using the whole images.

2.2.1 Corner feature detection

Corner detection is the most common approach among feature detection methods. It should be noted that, for image processing, higher resolution means clearer image, and easier to detect feature points. However, higher resolution also means that the calculation will be more costly, so the feature point detection and matching need to balance the effect and speed at the same time. At present, there are FAST corner detection algorithm [24, 25], SUSAN corner detection algorithm [26], Harris corner detection algorithm [27], SIFT algorithm with scale and rotation invariance [28], its improved SURF algorithm [29] and ORB detection algorithm [30]. In this section, we present a method for feature point detection and matching based on the FAST and SURF algorithms.

2.2.1.1 FAST Corner detection

FAST was first proposed by E.Rosten and T.Drummond in 2006. As one corner detector, FAST is fast and good at locating the position in images. According to the computational constraints of hardware and the requirement of experiment environment, we adopt FAST as the feature detector in this paper.

FAST is a machine learning method to detect feature. It assumes that the pixels which differs from surrounding neighborhood may be corner. The Fig 2.2 from the paper of E.Rosten shows the principle introduction about FAST. The steps of FAST can be described like this:

(1) Suppose that there exists a H, we are not sure if it is a corner or not. At first, we draw a circle centered it with a radius of 3 pixels, which are 16 pixels unit $(H_1, H_2, H_3, \ldots, H_{16})$ around H as shown in Fig 2.2.

(2) Define a threshold T, which will compare with H later, if the absolute value is less than H, will delete, if not, will be kept and be further investigation;

(3) Calculate H_1, H_9, H_5, H_{13} and the center H, if they have at least three absolute

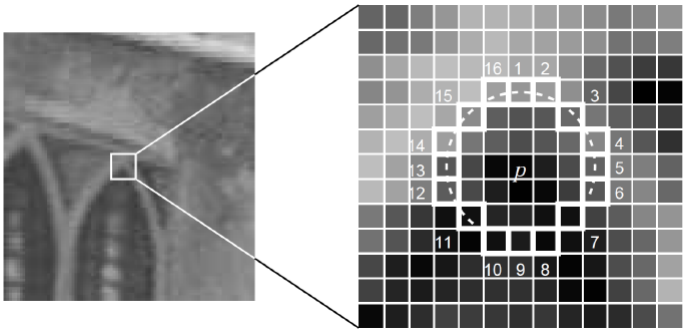


FIGURE 2.2: Introduction about FAST detector(Image taken from original FAST paper [24]).

values exceeded the threshold T, they can be saved as candidate points, and then the next step;
(4) Calculate H_1 to H_{16} the 16 points and the center H, if they have at least 9 more than the threshold, it is a feature point;

2.2.1.2 Feature matching based on SURF descriptor

When the feature points or corner points in two successive image sequences are detected, a matching relation is formed in order to find the corresponding relation among the feature points. We must add all the feature points to represent the identity of the description of the operator, and Harris or FAST and other corner detection algorithm itself does not contain the descriptor, so we must use other algorithms. They are SIFT, SURF, and so on. Considering that SIFT is far less than SURF in terms of speed, this paper adopts the SURF description algorithm to carry out feature matching [29]. SURF algorithm is a scale invariant image feature detection and matching algorithm, since it can simultaneously achieve feature detection and matching. SURF is known to have the same scale advantages. Another well-known drawback is the large amount of computing, which restricts its application of one of the factors. In order to meet the requirement of odometer, this paper selects a faster FAST feature detection algorithm, and uses it with the SURF feature descriptor to form a feature matching scheme for sequential image sequences. SURF characteristics of the calculation process is as follows:
(1) The First operation is the selection of main direction, since SURF can guarantee that the features have a rotation invariance. NO matter from which point of view of the same feature points, it can be considered as the same feature points. The way to achieve this is to assign a principal direction to the feature point and then describe the feature from the coordinate system in that direction whenever needed. SURF algorithm calculates the harr wavelet values. The direction of the largest fan-shaped area is defined

as the main direction, and the coordinate system is also established in this direction.
(2) We describe the feature point by taking the current feature point as the origin point, the main direction of the previous step as the Y axis, X axis. Then, we construct a square area with side length of 20 * s ($sisthecurrentpointsize$), and then divide the square into 4 * 4 = 16 small square areas. In each cell calculate all 25 pixels harr wavelet values in the horizontal and vertical direction. In this way, 16 regions can obtain a 64-dimensional feature vector, which is the feature vector of the current feature point.
(3) About feature matching, after adding descriptors to each feature point, it is possible to determine whether a pair of matching points is achieved by finding the similarity of the feature descriptors in the two images. We choose the most commonly used Euclidean distance as the measure of similarity. The formula is also the most familiar distance calculation method:

$$\rho\left(m,n\right)=\sqrt{\sum_{i=1}^{k}\left(m(i)-n(i)\right)^{2}} \tag{2.10}$$

where, m stands for 64-dimensional feature description vector of the first feature point; n is the 64-dimensional feature description vector of the second feature point to be matched.
In this paper, the nearest neighbor matching algorithm is adopted, the algorithm can be simply described as follows: Suppose we need to find the corresponding matching points of the first feature in the second image. First, we take point p_i of first picture as the candidate, and then traverse each feature point p_j sequentially in the second graph, and calculate the Euclidean distances $p\left(i,j\right)$ between them. If the current distance is smaller than the previous distance, update minimum record, and record i, j, until the last j traversal is complete, (i,j) is a pair of matching points.

2.2.2 Optical flow method

In the navigation research, camera poses are indirectly estimated based on the position and orientation of the moving object in the image, which is called the indirect method visual odometry. While there are also another methods known as direct methods, which directly estimate the camera movement based on image pixel value or gray value changes. In this way, the pipeline eliminate steps of feature detection and matching, and improve the accuracy of visual odometry by reducing the error accused by wrong matching. Among direct methods, the optical flow method is a typical representative one.

2.2.2.1 Overview

The concept of optical flow method was proposed in the 1950s [31], and it played a very important role in the research field of robot vision. Simply, optical flow method refers to the instantaneous velocity of the corresponding pixel due to the motion of objects in the image, and show the change in chronological order. And the optical flow of all the pixels of whole image is called optical flow field. The motion of objects in the real three-dimensional world is called motion field, while the projection of the motion field on the two-dimensional image plane is the optical flow field. The significance of studying optical flow is that the real motion can be estimated by the motion of pixels. However, the application of optical flow method has three premise assumptions:

(1) The brightness of the adjacent image frames is constant;
(2) During adjacent sampling frames, the scope of the object movement or amplitude can not be too large;
(3) Space movement must be consistent, that means all the pixels of the same image region have the same motion trend. Assuming the motion vector of a pixel in the image is $U = (u, v)$, where u and v are the components of velocity in X and Y directions. Suppose $I(x, y, t)$ is the gray value (brightness) of pixel (x, y) at time t, and according to the first assumption above, we have:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{2.11}$$

besides:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt \tag{2.12}$$

so we have:

$$I_x u + I_y v + I_t = 0 \tag{2.13}$$

$$u = \frac{dx}{dt}, v = \frac{dy}{dt} \tag{2.14}$$

in (2.13), I_x, I_y, I_t stands for the partial derivatives of I in x, y, $t$directions. (2.13) is the optical flow constraint equation, and it can be described in the format of vector:

$$IU + I_t = 0 \tag{2.15}$$

2.2.2.2 Lucas-Kanade algorithm

Horn-Schunck algorithm [31] and Lucas-Kanade algorithm [32] both are classical optical flow methods. The most common method of optical method is Lucas-Kanade method. It was published by Bruce Luca and Takeo Kanade. Mainly used in the calculation of

image optical flow based on least squares.

Flow constraint equation (2.13) contains three unknown parameters which can not be directly calculated, and this is the so-called aperture problem. The solution of this problem includes iterative or overdetermined equations. The Lucas-Kanade algorithm is a non-iterative method. Based on the third assumption in the previous section, we can see that the optical flow of given window is same. So they all satisfy the same constraint equation, as follows:

$$\begin{cases} I_x(q_1)\,u + I_y(q_1)\,v = -I_t(q_1) \\ I_x(q_2)\,u + I_y(q_2)\,v = -I_t(q_2) \\ \cdots\cdots \\ I_x(q_n)\,u + I_y(q_n)\,v = -I_t(q_n) \end{cases} \tag{2.16}$$

Here q_n are the points in the given window, and $I_x(q_1)$, $I_y(q_2)$, $I_t(q_3)$ are the partial derivatives of points in x, y, z directions. Because the number of constraint equations is greater than the number of unknown parameters, this is a typical problem of overdetermined equations.

For (2.16) can be written in the format of $AU = b$:

$$A = \begin{bmatrix} I_x(q_1) & I_x(q_1) \\ I_x(q_2) & I_x(q_2) \\ \cdots & \cdots \\ \cdots & \cdots \\ I_x(q_n) & I_x(q_n) \end{bmatrix} \tag{2.17}$$

$$U = \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.18}$$

$$b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \cdots \\ \cdots \\ -I_t(q_n) \end{bmatrix} \tag{2.19}$$

According to the least square method to solve the equation:

$$A^T A U = A^T b \tag{2.20}$$

or

$$U = \left(A^T A\right)^{-1} A^T b \tag{2.21}$$

In this case, the optical flow equation can be solved:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i) I_y(q_i) \\ \sum_i I_x(q_i) I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix} \begin{bmatrix} -\sum_i I_x(q_i) I_t(q_i) \\ -\sum_i I_y(q_i) I_t(q_i) \end{bmatrix} \tag{2.22}$$

2.2.3 Selection of indirect and direct methods

The advantages of direct visual odometry are described above,however, indirect methods Feature point detection and matching method is still adopted to construct monocular visual odometry in this paper. The reasons are as follows:

- For the sparse optical flow method,common approach is to specify the pixels which need to be tracked in advance, and then calculate the optical flow of these points. That is to say, the feature points need to be detected before the optical flow is calculated, and the steps are not reduced;
- For the dense optical flow, the optical flow of all the pixels on the whole image can be directly calculated, which is very large amount, so it is difficult to guarantee by normal computers;
- The optical flow is suitable for tracking the feature points moving in a small range, and the effect is poor for the fierce motion;
- Since there is no need to match the feature points, the optical flow method will directly calculate the interference of the moving object to the whole odometer system, and it is not convenient to remove the moving interference error. In this case, the method proposed later in this paper is useless.

2.3 Motion estimation methods

Based on image processing, the basic idea of motion estimation is to filter some pairs of points from the previously paired feature points. The filter prevents the incorrect matching points from effecting whole odometry system. Then the basic matrix is calculated according to the position change of the same point in the adjacent images, and then the essence matrix containing the motion parameters can be decomposed according to the basic matrix [20-24].

2.3.1 RANSAC random sampling consistent algorithm

After obtaining the matching points of the two images, some of the points will be used to estimate the camera's motion parameters, and some inaccurate points will inevitably produce some negative effect. Then if these mismatching points are used to calculate the motion parameters, it will give the odometry a certain error, it means, there will be robustness on error. Therefore, we need a suitable way to pick up matching and mismatching pairs. RANSAC (RANAC) algorithm is often used to solve this kind of problem and improve the robustness of feature matching system [33].

The RANSAC algorithm was first proposed by Fischler and Bolles to calculate a logical model from a set of data containing anomalous data, and then distinguish the valid and abnormal data from the set according to the model [33]. The basic steps of RANSAC are as follows:
(1) Suppose that there exists a set P including N data, which can be described by the model M, and the construction of M needs at least data n, and $n < N$;
(2) Randomly extract n data from the set P to build a subset S, and use these n data to calculate a model M_i;
(3) Ideally, the remainder of P excluding S_i, the data set S_c also satisfy model M_i. Compute the error e of each data in S_c separately under the constraint of model M_i. If e is less than a given threshold θ, this data can be incorporated into the set of samples S_i. The new set S_i^* is called the uniform set, and the data in S_i^* are called inlier, the others are outliers.
(4) Repeat the steps of (2) and (3) to re-sample the consistent set. If the number of points is greater than the previous one, update the consistent set, otherwise discard the results;
(5) If the sampling number reaches a given upper limit, still no consistent set is found, which means the algorithm fails for no matching point. Otherwise, it is best to use the most consistent set of points in the largest consistent set to estimate the camera poses movement. It should be noted that, in this monocular visual odometry system, the above mentioned model M is the base matrix mentioned below, and the threshold θ is the Simpson distance.

2.3.2 Hartley eight-points basis matrix algorithm

Calculates the relative motion of the camera through corresponding points in two images. Because the camera shoots the same scene from different locations, the overlapped parts of the scene satisfy geometric constraint relationship, and then the base matrix is the

algebraic representation of the pole geometry:

$$m'Fm = 0 \tag{2.23}$$

$$F = K^T[t]_x RX \tag{2.24}$$

during that, m stands for the coordinates of a pixel in the first image, m' stands for the pixel in the second image corresponding to m, F is the base matrix, K is camera internal reference matrix, $[t]_x$ is the antisymmetric matrix defined by the translation vector t, R is camera rotation matrix.
Assuming $m = (x, y, f_0)^T$, $m' = (x', y', f_0)^T$, based on (2.22):

$$\begin{bmatrix} x' & y' & f_0 \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{12} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ f_0 \end{bmatrix} = 0 \tag{2.25}$$

Expend it, we can get this:

$$\begin{bmatrix} x'x & x'y & x'f_0 & y'x & y'y & y'y & y'f_0 & xf_0 & yf_0 & f_0^2 \end{bmatrix} u = 0 \tag{2.26}$$

During it:

$$u = \begin{bmatrix} F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33} \end{bmatrix}^T \tag{2.27}$$

According (2.25), this this a 9 freedom equation, and its basic requirement is $detF = 0$. In that way, the above equation change to a 8 freedom equation. We only need 8 match points to calculate basic matrix F, and the steps like this:

- Find the eight pairs of suitable matching points;
- Construct the linear equations with the given points;
- Singular value decomposition of the coefficient matrix A of the above equations: $A = UDV^T$, the last column vector of V is the base matrix F.

As the error inevitably occurs during the feature detection and matching. It means that the position of the feature points maybe are not accurate or the coordinate may occurs incorrectness,then it causes motion estimation error. The equation (2.26) is usually solved by least squares method. Because of the existence of the coordinate error of the feature points, the matrix of this equation may be very large, which is extremely unfavorable to the calculation result, and it may result in instability. In mathematics, in order to avoid solving the equation with the above problems, it usually execute the normalization on the raw data at first, which means the original data has the same

scale in all directions. In this way, normalization can reduce the coordinate error caused by unreliability of the coordinate, and improve the accuracy of the basis of matrix calculation.

Hartley proposed a normalized approach [34], in which by first translating the global position to ensures that the data is not biased to one side. And then the approach converses the scale of the data, which can guarantee that the abnormal data will not be too large on the overall. Therefore, using the normalized Hartley eight-point algorithm to calculate the basic matrix can help the estimation of the motion parameters to have a certain robustness.

2.3.3 Construction of MVO based on essence matrix

Longuet-Higgins [35] find the camera pose information corresponding to the two views is contained in the essence matrix E, and the relationship of E and F is like this:

$$E = K^T F K \tag{2.28}$$

After calculating F in , for internal matrix K is already known, so we can get E. Based on Longuaet-Higgins algorithm, after singular value decomposition on E, we can get camera rotation matrix R and translation matrix T. However, here is another problem need to be solved, which is scale ambiguity. The translation matrix is unit vector without scale information, the specific method is dismiss E into $E = USV^T$, where T is equal to the third column of U without scale information. After that, we assume:

$$D = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.29}$$

And then we get $R = UDV^T$. After we got R and T, next step is to generate MVO. Assuming camera poses P in world coordinate system, at one time its position is $p_1 = (x_1, y_1, z_1)$, and its next time is $p_2 = (x_2, y_2, z_2)$, then we got:

$$p_2 = Rp_1 + T \tag{2.30}$$

During them,

$$R = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix}, T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.31}$$

During them, R is camera rotation matrix and T is camera translation matrix. In order to calculate it more easily, the former equation can write like this:

$$p_2 = Mp_1 \tag{2.32}$$

During it:

$$M = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} & t_x \\ r_{yx} & r_{yy} & r_{yz} & t_y \\ r_{zx} & r_{zy} & r_{zz} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.33}$$

That is to say, after obtaining the rotation and translation matrix from essential matrix, we can obtain the coordinates of the current time from the advance coordinate of the camera, which is the way to realize the monocular visual odometry.

2.4 Conclusion

In this chapter, the basic pipeline of monocular visual odometry design based on feature detection and matching is described. The FAST method is used to improve the matching speed of the feature, and the RANSAC random sampling algorithm is adopted to reduce the error matching and improve the robustness of odometry. In addition, the use of Hartley normalization algorithm can effectively reduce the base matrix generated when calculating the error and improve the accuracy odometry.

Chapter 3

Moving targets elimination in dynamic background

Vision-based robot navigation technology has always been a hot topic of research for scientists, and vision technology is one of the trends of artificial intelligence. However, most of research of monocular visual odometry focus on geometrical methods, MVO technology based on machine learning has little breakthrough news in recently years [36]. The reason is nothing more than that visual method or image processing technology is difficult to ensure the reliability of the real application environment. It is common for academia to verify that a visual algorithm is assumed to be in a particular and single experimental scenario, and that it usually ignores external disturbances. When these algorithms are applied to the real environment, it can not meet a variety of harsh conditions in real environment. For the monocular visual odometry pipeline discussed in this paper, most of the papers tested their result in the monotonous background to achieve the navigation tasks. In fact, robots or agents are usually working in the complex environment where are full of moving objects.

The monocular visual odometry studied in this paper is based on matching feature points of inter-frame images to estimate the agent motion and trajectory. That is to say, the detection and matching of feature points are crucial part during the whole positioning and navigation, because the positional variation of the camera is estimated by the variation of identical feature points in different frames. If there are moving objects in the field of view while camera keep motionless, the matching points on the inter-frame will also move with the moving objects, which will be assumed that the camera is also moving and results in the wrong motion estimation. The RANSAC algorithm is adopted in the feature matching part of the previous chapters, which can weaken the influence of the dynamic feature points by sampling method. However, this method can not fundamentally solve the problem. In this paper, we change the viewpoint from

mitigating wrong match the feature to eliminating moving features. If we do not let the algorithm to match the feature points of the moving objects to avoid the wrong motion estimation, we can remove moving objects features to reduce interference and achieve a more accurate result.

3.1 Moving object detection algorithm

Moving object detection has always been a popular research direction in computer vision. One of the applications is video surveillance. For example, road junctions, train stations, parks, banks, corridors and other public places, often need to layout monitoring equipment. Video surveillance can help managers to detect unusual behavior or suspicious people. At present, most video surveillance need manual operations and observation. If they can automatically detect the target, it will greatly improve the efficiency of video surveillance.

Different from video surveillance, moving target detection in ego-motion videos has particularity. First of all, the monitoring situation, the camera is generally fixed or man-made control of small-scale rotation, so that most of the time, backgrounds of video are fixed. The target detection in this case is called moving target detection in static background. In the monocular visual odometry, the camera is moving with the agent where cameras are fixed, so the backgrounds of the videos are constantly changing with the movement.In another situation, the cameras are still, the targets are moving, and the backgrounds are also moving. We call moving object detection in these two cases as moving object detection in dynamic background.

At present, the moving targets detection algorithm under static background has been relatively mature, the actual effect is also good, such as inter-frame subtraction method [37], background subtraction method [38] and optical flow method [31] and so on. Detection of moving objects in dynamic background is more complicated. The common methods include optical flow method and background compensation method [39]. In the following, we introduce the target detection algorithm in both static and dynamic background situations.

3.1.1 A Moving Object Detection Algorithm in Static Background

3.1.1.1 Inter-frame subtraction algorithm

Inter-frame subtraction method should be the most basic of all moving targets detection algorithms. Its basic idea is to detect the difference between the two adjacent images. At first, subtract the corresponding pixel gray value in the adjacent two images, and

then classify the difference between each point. If the difference is small, it is considered as static object; if the difference is more than a certain threshold value, it means the pixel gray value changes greatly, then it is considered as moving area. Moving target area is set to 1, while still scene is set to 0, as follows:

$$D(x,y) = \begin{cases} 1, |f_{k+1}(x,y) - f_k(x,y)| > T \\ 0, others \end{cases} \tag{3.1}$$

Here, $D(x,y)$ represents the difference images; $f_{k+1}(x,y)$ means the gray value of the corresponding point in the next frame; $f_k(x,y)$ represents the gray value of corresponding point in the current frame image; T means the threshold of binarization. According points value in the binary images to judge the motion status of every point.
The inter-frame difference method is easy to build and work very fast, while it has many defects. For example, when the object moves slowly, the location may change little, which will arouse overlapping areas and hollow region in binary map. Besides that, if the target suddenly stops during the continue movement, then the algorithm can not detect the tracking targets for the temporary interruption of movement. In order to overcome these shortcomings, the researchers proposed a lot of improved inter-frame difference method, such as three frames difference method and so on.

3.1.1.2 Background subtraction algorithm

Based on the previously mentioned method with two adjoint images, background subtraction algorithm subtract background and moving objects to extract targets. At first, collect a scene without moving objects in the background, and then use the current image gray values to minus the background image corresponding pixels. In this case, the region where the moving objects does not exist is naturally reduced to 0, and the region where the target exists is not 0. The process can be described as follows:

$$D_x(x,y) = |f_k(x,y) - f_b(x,y)| \tag{3.2}$$

Here, $f_b(x,y)$ is the prepared background image, $f_k(x,y)$ is the current image, $D_k(x,y)$ is the image after subtraction. Since the background image is the essential part in this method, the final target detection will be greatly effected if the image is not properly extracted, which creates a natural flaw. Firstly, it is impractical to get the background image in advanced, and the camera can not be slightly moved. When the camera is moved to another place or changed to another field of view, you have to change the background image. Second, even if the background scene itself does not change, as time changes, the light condition will be different, and then the gray value of background

image should change, the same one can not be used. To solve these problems, Gaussian background modeling method [40] was proposed. According this method, the gray values of the pixels in the background image should meet the Gaussian distribution. In this way, we can collect some background images under different lighting conditions, and use these images as the training data to establish the Gaussian distribution model of each pixel, that is:

$$I(x,y) = N(\mu(x,y), \varphi(x,y)) \tag{3.3}$$

Gaussian model is established, according to the value of the Gaussian function, all the pixels on the image are divided into background or foreground. After this transform, the motion detection problem becomes a typical binary classification problem. According to the principle of Gaussian distribution, if a pixel is the background, then it should be a certain probability close to its mean $u(x,y)$, that is to meet:

$$|I_k(x,y) - u_{k-1}(x,y)| > p \cdot \sigma_{k-1}(x,y) \tag{3.4}$$

3.1.1.3 Optical flow method

The concept of optical flow has already been introduced in Chapter 2, which concludes that the actual real world motion field will be represented by the change of pixel gray value as the optical flow field in the image. Conversely, we can determine the existence of moving objects based on the existence of the optical flow field.
The advantage of using the optical flow method to detect moving objects is that motion can be detected both in dynamic backgrounds and in static backgrounds. But it also has two problems. First, when both the background and the foreground are moving, the optical flew method computes both of them without distinction. In this case, we cannot determine which is the movement of the moving objects by the flew field. This shortcoming limit the application of optical flow method in the dynamic background of the moving object detection. Another problem is that, optical flow method is divided into sparse optical flow and dense optical flow two directions research. Sparse optical flow refers to only specify pixels of the designated field, which is not enough to detect the whole movement; On the other hand, dense optical flow is to calculate the motion of all pixels on the whole image. While the calculation of the optical flow itself is already very complicated, the computation within all the pixels to be calculated is very large, so its speed will delay the generation of the visual trajectory.

3.1.2 Moving object detection in the dynamic background

Because of the difference between dynamic background image and static background image, even the motion detection algorithm in static background can not be used in dynamic circumstance, these algorithms provide some ideas for the research. Commonly used dynamic background detection algorithms include the optical flow method [31]which is described in the above chapter and the background compensation differential method [39].

The so-called background compensation differential method, refers to estimate the motion parameters of camera at first, and then to implement to background image motion compensation, which include background translation, rotation, affine transformation. After that, compare the transformed background and the current frame to detect the moving objects. In fact, it uses motion compensation to transform a detection problem in a dynamic background into a relatively simple object detection problem in a static condition. The key of background compensation method is how to perform motion compensation, or how to estimate the camera's motion parameters. This requirement conflict with the original purpose of this paper, which is in order to improve the accurate of camera move estimation to detect the moving targets. Algorithm requirements can not be met, therefore, this paper have to abandon the use of background compensation differential method.

The two methods are based on the same premise, that is, we do not know what the moving objects are. So the only way to detect the goal is according to the image pixel brightness or gray value changes. In another way, we can turn a moving object detection problem into a recognition problem. For example, the face recognition problem is very successful in the field of image processing, which is to detect the moving face in the dynamic background [41]. Usually the recognition steps are, to find a bunch of priori features as a training data to train a classifier at first, and then through the classifier to determine whether the target in the images. Commonly used classifiers are Adboost [42], Random Forest [43], GBDT [44], SVM [17] and so on.

3.2 Pedestrian detection based on moving object extraction

In order to minimize the impact of moving objects on visual odometer reliability, it is better to be able to detect more types of moving objects. If you want to recognize all these moving objects from pattern recognition, you must build a variety of classifiers, which need a variety of moving objects, such as animals, vehicles, human beings and other moving objects in the real life environment. The algorithm implement will be

very complicated. During the daily life, human beings are the most important groups of the environment and also are one of the main external disturbances during the robot navigation. In this paper, human is regarded as the main moving target in the process of robot visual odometry, and the accuracy of MVO is improved by eliminating the pedestrian feature points.

The key to pedestrian detection is how to describe human characteristics in images. Pedestrian appearance, clothing, and even action posture will be different. It is difficult to use a limited number of samples to describe all these features. A typical solution is the HOG + SVM pedestrian detection scheme which is published by Dalal at CVPR [45]. In this paper, the HOG feature and SVM classifier are used to implement pedestrian detection.

3.2.1 HOG features

HOG refers to the direction gradient histogram, which is the operator to describe the characteristics. The method is to calculate the color histogram of image pixels in the gradient direction. Because the gradient direction is adopted, it can detect the edge of the object very well, and it also has some robustness to the change of illumination. Since the shape of the normal human is basically fixed, only the size of different, so HOG is very appropriate detection method to detect human being.

The gradient of the pixels x, y in the image is calculated as follows: Horizontal gradient:

$$G_x(x,y) = I(x+1,y) - I(x-1,y) \tag{3.5}$$

Vertical gradient:

$$G_x(x,y) = I(x,y+1) - I(x,y-1) \tag{3.6}$$

And about the point (x,y) in the images: Gradient Amplitude:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{3.7}$$

Gradient direction:

$$a(x,y) = tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \tag{3.8}$$

The function of HOG is similar to the SURF described in the previous chapter, which is to provide a descriptive label for the image. The steps to build HOG are like this:

- First, we should divide the original images into smaller units with fixed pixel;
- Then, calculate the statistical histogram from every gradient directions for each unit. When the unit circle are evenly divided into nine regions, each region can be

called one gradient direction. So each unit can be described with a 9-dimensional feature vector. Since the unit gradient is strongly influenced by the light, it is necessary to avoid the regional interference. The method is to class the adjacent 4 units into one block, so each block can get a 36 dimension eigenvector.

- Finally, scan the block window with the size of one unit, each scan can get a 36-dimensional vector. The whole image can be linked by all of these 36-dimensional vector series.

The HOG is also a feature that is required for the following SVM model.

3.2.2 Classification SVM

Support Vector Machine (SVM) [17] is a machine learning algorithm based on statistical learning theory, which is firstly put forward by Vapnik. It has the unique advantage in solving small sample, nonlinear and high latitude problems. SVM is known as an algorithm that allows applied mathematicians to be truly applied and is often used in engineering to solve classification and prediction problems. In this paper, we will introduce the theoretical basis of SVM application of classification.
Classification SVM is often divided into two class classification and multi-classification problems. According the application in this paper, we all involve the two class classification. As shown in the following Fig 3.1, the known data sets are divided into two categories: blue points A and red points B. The problem is to determine which class of C should belong to.

A very intuitive method is to draw a line $g(x) = wx + b$ between A and B as a dividing

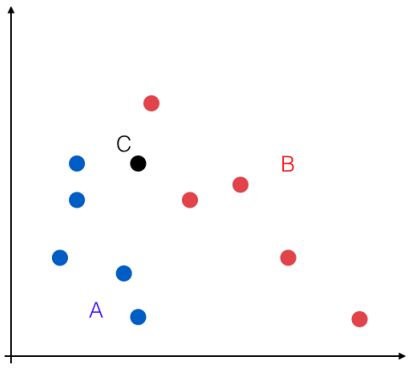


FIGURE 3.1: Two class classification problem in SVM.

line just like Fig 3.2, the points above the line belong to the class A, and the points below the line belong to the class B. But there are a lot of straight lines can divide the above data into two types, we need to find the optimal one.

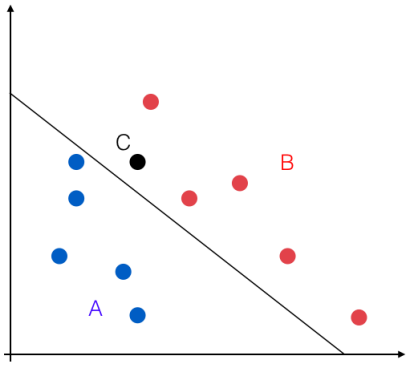


FIGURE 3.2: Draw a line between two class in SVM.

On the A class side, we find a straight line $g(x) = wx + b = 1$, which passes through the boundary point of the A class; On the B class side, we find a straight line $g(x) = wx + b = -1$, which passes through the B class boundary. The distance between the two lines is called the margin distance, showing in red in Fig 3.3. Such a line is the best one when it lies between $g(x) = wx+b = 1$ and $g(x) = wx+b = -1$, and the interval is at its maximum. This is also the essence of the principle of SVM, which is called the largest interval classification. The straight lines $g(x) = -1$ and $g(x) = 1$ are called the support vectors.

From the above analysis, we can see that we need to find the maximum interval of the

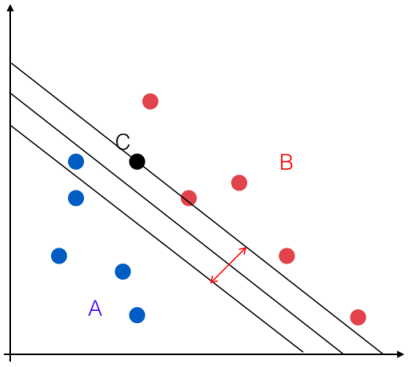


FIGURE 3.3: Maximum margin classifier in SVM.

boundary $g(x) = wx + b$. From geometry viewpoint, the interval is $M = 2/\|w|$, so the classification problem can be transformed into the following optimization problem:

$$max \frac{1}{\|w\|}, s.t. y_i (wx_i + b) \geq 1, i = 1, 2, \ldots n \tag{3.9}$$

Equal to:

$$max \frac{1}{2}\|w\|^2, s.t. y_i (wx_i + b) \geq 1, i = 1, 2, \ldots n \tag{3.10}$$

The above equation implies that, under some certain constraint conditions, when the value of the objective function is minimum, w and b are the solutions of the original

boundary. This is a convex quadratic programming problem, and the direct solution is very troublesome. The solution is based on the Lagrangian duality principle to convert the original optimization problem into its dual problem. According to (3.10), we define the Lagrangian function as:

$$\zeta(w, b, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i \left(y_i \left(w x_i + b\right) - 1\right) \tag{3.11}$$

Our aim is to let w, b to be the smallest using ζ, according to the nature of derivative, we need only let the partial derivatives of w and b be zero.

$$\frac{\vartheta\zeta}{\vartheta w} = 0 \Rightarrow w = \sum_{i=1}^{n} a_i y_i x_i \frac{\vartheta\zeta}{\vartheta b} = 0 \Rightarrow w = \sum_{i=1}^{n} a_i y_i \tag{3.12}$$

Then we get:

$$\zeta(w, b, a) = \frac{1}{2}\sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j x_i^T x_j \tag{3.13}$$

With the constraints, the new optimization problems can be solved:

$$max \frac{1}{2}\sum_{i=1}^{n} a_i \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j x_i^T x_j \tag{3.14}$$

Then:

$$sum_{i=1} n a_i y_i = 0, a_i \geq 0, i = 1, 2, \ldots n \tag{3.15}$$

In this way, we can solve the problem of two class classification problem using SVM.

3.2.3 HOG + SVM modeling and training

In order to obtain a suitable SVM classifier for pedestrian detection, we need to use a large number of pedestrian images with HOG features to train our own classifier model. This paper designs a general pedestrian detection pipeline, the steps can be described like following:

- Collect the training sample dataset, which should include the positive sample with pedestrians and negative sample without pedestrians. Train data is the essential part to build a efficient classifier model. About the train data, it should contain all the possible positive features and necessary number of negative samples. For the proportion of positive and negative samples need to be adjusted by experiment.
- Extract the positive and negative samples of HOG features. Before extracting features, we need to crop the sample image to the same size, such as 64 * 128.

- All the positive and negative samples of the HOG feature labels, such as 1 for pedestrians, 0 for no pedestrians.
- All the positive and negative samples with HOG features and tag values are input into SVM training to obtain the initial classification model.
- Randomly choose part of original training data set as test samples, and classify them using the model of previous step. The misclassified images are collected as hard examples.
- Add the hard examples to the negative sample of the original training data, and retrain the SVM model. The model become more accurate than the previous one.

After this, the pipeline Fig 2.1 can transform to 3.4 in this paper.

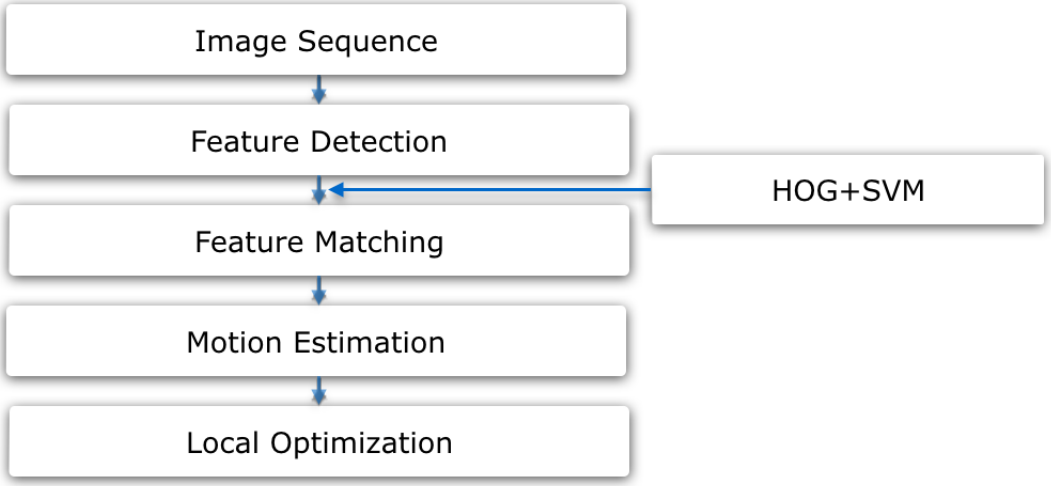


FIGURE 3.4: The pipeline of MVO combined with HOG + SVM

3.3 Conclusion

In this chapter, we discusses the interference problem of moving objects in MVO system when it is adopted in the real scene, and proposes a pedestrian detection algorithm using HOG + SVM to detect moving human. The scheme can solve the problem of motion interference to a certain level, but it still has defects.

Chapter 4

Experimental results and analysis

In this chapter, we will test the algorithms described in the previous chapters and compare with similar methods. The experimental platform information is shown in Table 4.1:

Table 4.1: Experiments setup

	Configuration parameter
Computer hardware	CPU: inter core i7 RAM: 8G
OS	Ubuntu 14.04 LTS
Program Software	Python, Matlab
Related	OpenCV, ROS, LibSVM

4.1 Image correction experiment

About camera cailibration, Zhang Zhengyou checkerboard calibration algorithm is available in many platforms, such as Matlab, OpenCV and ROS. ROS calibration library is easy to use and can directly calibrate the camera, while Matlab can only calibrate sequences. For the experiment image correction, since some images data have distortion, it is necessary to carry out image correction before the fellowing experiments. In this paper, the bilinear interpolation method is used to correct images, and the results are shown in Fig 4.1. As the result can be seen, the distorted straight lines has been re-straightened, the line is relatively smooth, and maintain the original gray value. The corrected image meets the requirements of monocular visual odometry and SVM model.

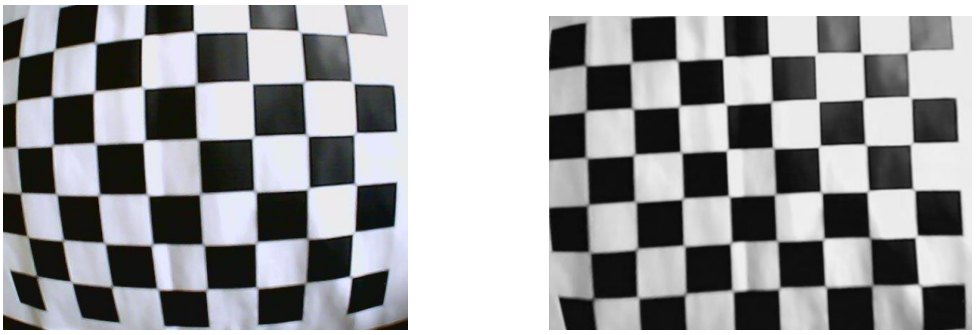


FIGURE 4.1: Image correction using bilinear interpolation method.(Left: before, Right: after).

4.2 Feature detection and matching experiment

In this paper, we use the "FAST + SURF" feature point detection and matching scheme, which is different from simply using SURF algorithm to detect and match at the same time. We use FAST corner detection algorithm to detect most of the corners in the image clearly, and then use the SURF to describe these feature points for feature matching. The following Fig 4.2 shows the experimental results of feature detection, and Fig 4.3 show the matching result. Standard SURF combination program detected 502 corner points

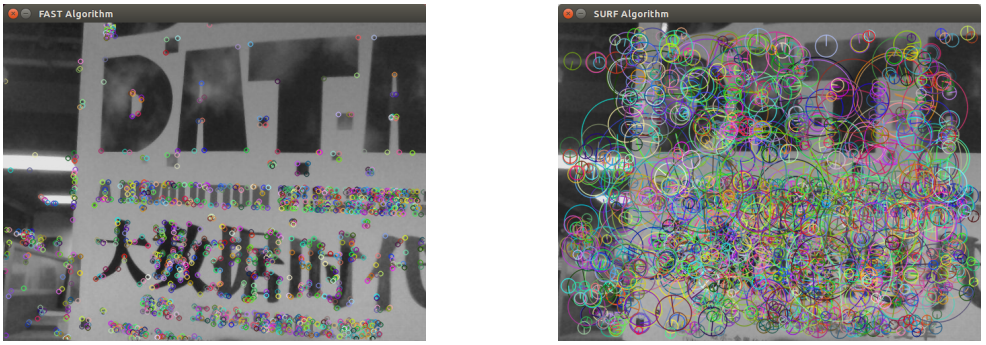
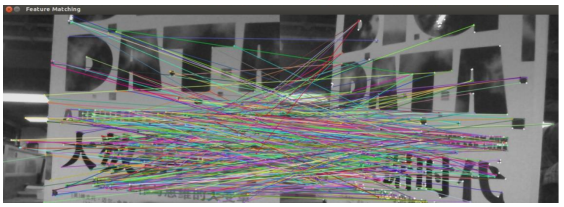
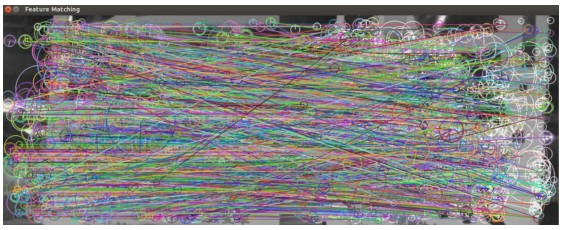


FIGURE 4.2: Comparison of the feature detection algorithms.(Left: FAST, Right: SURF).

in the left images, 625 corner points in the right one; while FAST+ SURF algorithm, the left image detected 1507 corners, 1548 corner points in the right one. Furthermore, there is still a wrong match between the two schemes, which is why RANSAC sampling algorithm is emphasized in monocular visual odometry.

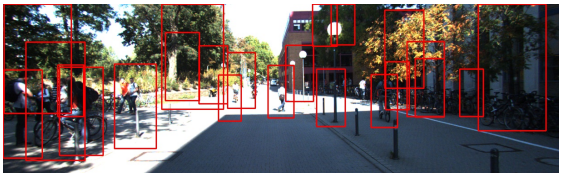


(a) FAST+SURF feature matching



(b) SURF feature matching

FIGURE 4.3: Comparison of the feature matching algorithms.



(a) The pedestrian detection result 1 (Positive: 2400, Negative: 12000)

(b) The pedestrian detection result 1 (Positive: 2400, Negative: 12000, Hard example: 400)

FIGURE 4.4: Comparison of the pedestrian detection results in different training data.

4.3 Pedestrian Detection and elimination experiment

4.3.1 HOG + SVM model train

For the pedestrian detection training and test sets, all the images are resized in 64 pixel * 128 pixel. We divide sequence images into positive and negative parts and combine with existing INRIA human samples [45]. Next, we retrain the negative samples to detect the hard examples using the trained SVM to improve the accuracy. The training and test result is shown in the fellowing Fig 4.4. It indicate that, with hard example retrain, we can get a better accuracy in pedestrian detection.

4.3.2 KITTI data experiment

4.3.2.1 Error metrics

In the experiment, we use Root Mean Square Error (RMSE) to evaluate the performance of the proposed system and simple geometric monocular visual odometry with ground truth in every parameters. The KITTI Ground truth data [9] includes 12 parameters of the camera position about rotation and translation. The fellowing Table show the result, we confirm that pedestrians feature points elimination reduce the error of camera pose estimation and trajectory generation. In there, ML+MVO means monocular visual odometry with machine learning methods. From Table 4.2, we can see the RMSE of translation in ML+MVO is smaller than simple geometric MVO, which means the error is reduced by the pedestrian elimination at a certain level.

TABLE 4.2: Comparison between ML+MVO and geometric MVO

	ML+MVO				Geometric MVO			
	Rotation			Translation	Rotation			Translation
RMSE	0.0000	0.0007	0.0009	0.0026	0.0003	0.0023	0.0041	0.2183
	0.0008	0.0000	0.0007	0.0205	0.0023	0.0000	0.0025	0.0187
	0.0009	0.00011	0.0000	0.1252	0.0040	0.0026	0.0003	0.3081

4.3.2.2 Path generation

The dataset of KITTI 05 sequence include several pedestrian in the video, its reconstructed trajectories are shown in the following Fig 4.5.

4.4 Experimental conclusion

In this paper, we proposed MVO system combined with machine learning. Both methods had accumulated error problem. However compared with geometric MVO, the performance of MVO combined with machine learning showed higher accuracy and precision. It indicates that the moving objects effect the poses estimation negatively. The pipeline in this paper provides a baseline for future work to detect more moving objects. From this experiment, it also indicates that the result of machine learning also depends on good labelled training and test datasets, and computational burden is another issue with the increase of error correlations.

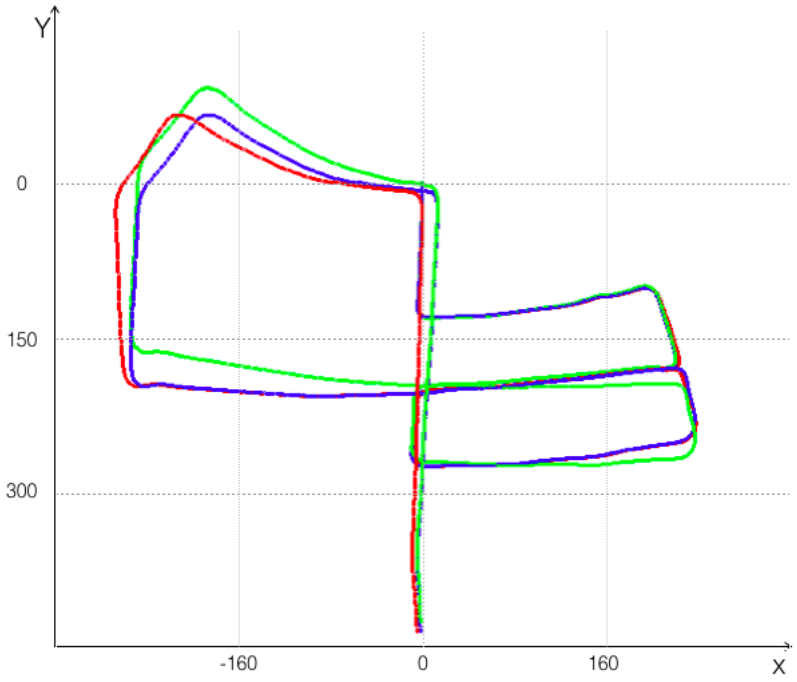


FIGURE 4.5: Trajectories of on the KITTI 05 set.
(Red: Ground truth, Green: Geometric MVO, Blue: MVO + machine learning).

Chapter 5

Conclusion

The main content of this paper is to study and design a monocular visual odometry system based on feature-based MVO and machine learning. In order to realize the autonomous positioning and mapping using ego-motion videos with pedestrian, this paper proposes a new method from two aspects to improve the accuracy and stability of system, which mainly includes the following innovative contents:

- A framework of feature detection and matching is proposed. In this paper, we use the FAST corner detector algorithm to detect the feature points and SURF to describe the feature points, which help to balance the points quantity and compute speed.
- Considering the negative effect of moving objects during the camera estimation, a moving objects elimination system based on pedestrian detection is designed. Based on the idea of machine learning, we first use the HOG feature of the image to train the SVM classifier. Then the pedestrian is detected and the feature points of pedestrians are removed to avoid the effect of motion estimation. The validity and feasibility of the above method are validated through several experiments. The monocular visual odometry in this paper does not have much theoretical significance, but some of the schemes or methods can be used for reference. Of course, the research of this paper still has the shortcomings, mainly has following two points:
- (1) The result of machine learning depends on good labeled training and test data, and computational burden is another issue with the increase of error correlations.
- (2) Second, on the dynamic interference solution, based on pedestrian detection method, the current algorithm is not fast enough, will affect the efficiency of odometry process. If we want to detect more kinds of moving objects and eliminate them

using SVM to improve the accuracy of MVO, we have to figure out how to raise the speed of training and classification from the algorithm level in the future work.

Appendix A

Appendix

A.1 List of academic achievements

[1] Y. Zhang and H. Watanabe: "Trajectory data visualization of sports video based on SLAM", ITE Annual Convention 2016, No.34C-1, Sep. 2016

[2] Y.Zhang and H.Watanabe: "Research on Trajectory Visualization of Ego-Motion Videos with Pedestrian Based on Monocular Visual Odometry and Machine Learning", IEICE Annual Conference 2017, Mar. 2017

Bibliography

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, Oct 2015.

[2] J. Engel, J. Stckler, and D. Cremers. Large-scale direct slam with stereo cameras. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1935–1942, Sept 2015.

[3] C. Lewicki J. Krajewski, K. Burke and C. Voorhees D. Limonadi, A. Trebi-Ollennu. Mer: from landing to six wheels on mars. 2:1791–1798, Oct 2005.

[4] A. Levin and R. Szeliski. Visual odometry and map correlation. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 1:I–611–I–618, June 2004. ISSN 1063-6919.

[5] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics Automation Magazine*, 18(4):80–92, Dec 2011. ISSN 1070-9932. doi: 10.1109/MRA.2011.943233.

[6] H.Moravec. Obstcal avoidance and navigation in the real world by a seeing robot rover. *Ph.D disseratation*, 1980.

[7] C. F. Olson, L. H. Matthies, H. Schoppers, and M. W. Maimone. Robust stereo ego-motion for long distance navigation. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000*, 2:453–458, 2000. ISSN 1063-6919. doi: 10.1109/CVPR.2000.854879.

[8] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. Svo: Semi-direct visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, (99):1–17, 2014. ISSN 1552-3098. doi: 10.1109/TRO.2016.2623335.

[9] Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago. Kitti vision benchmark suite. `http://www.cvlibs.net/datasets/kitti/`.

[10] Hiroshi Watanabe Yifei Zhang. Trajectory data visualization of sports video based on slam. *ITE Annual Convention.*

[11] O.Naroditsky D.Nister and J.Bergen. Visual odometry. *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 652–659, 2004.

[12] O. Naroditsky D.Nister and J. Bergen. Visual odometry for ground vehicle application. *J.Field Robot*, 23(1):3–20, 2006.

[13] P. Corke, D. Strelow, and S. Singh. Omnidirectional visual odometry for a planetary rover. *IEEE/RSJ on Intelligent Robots and Systems*, 4:4007–4012, Sept 2004. doi: 10.1109/IROS.2004.1390041.

[14] M. Lhuillier. Omnidirectional visual odometry for a planetary rover. *Proc. IEEE Workshop Omnidirectional Vision*, 4:1–8, 2005.

[15] Y. Pavlidis J. Tardif and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 2531–2538, 2008.

[16] Christopher. M Bishop. Pattern recognition and machine learning. pages 325–359, 2006.

[17] Vandewalle. J Suykens J. Least squares support vector machine classifier. *Neural Processing Letters*, 9(3), 1999.

[18] Sangwoo Cho, E. Dunn, and J. M. Frahm. Rotation estimation from cloud tracking. *IEEE Winter Conference on Applications of Computer Vision*, pages 917–924, March 2014. ISSN 1550-5790. doi: 10.1109/WACV.2014.6836006.

[19] J. G. Rogers, A. J. B. Trevor, C. Nieto-Granda, and H. I. Christensen. Simultaneous localization and mapping with learned object recognition and semantic data association. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1264–1270, Sept 2011. ISSN 2153-0858. doi: 10.1109/IROS.2011.6095152.

[20] J. Collier and A. Ramirez-Serrano. Environment classification for indoor/outdoor robotic mapping. *2009 Canadian Conference on Computer and Robot Vision*, pages 276–283, May 2009. doi: 10.1109/CRV.2009.6.

[21] R.Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, 1986.

[22] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000. ISSN 0162-8828.

[23] Q.-T. Luong O. Faugeras and S. Maybank. Camera self-calibration: Theory and experiments. *ECCV Lecture Notes in Computer Science*, 588, 1992.

[24] T. Drummond E. Rosten. Machine learning for hight-speed corner detection. *Proc. European Conf. Computer Vision*, 1, 2006.

[25] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119, 2010. doi: 10.1109/TPAMI.2008.275.

[26] Stephen M. Smith and J. Michael Brady. Susan&mdash:a new approach to low level image processing. *Int. J. Comput. Vision*, 23(1):45–78, May 1997.

[27] Chris Harris and Mike Stephens. A combined corner and edge detector. *Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[28] Steven Henikoff Pauline C. Ng. Sift: predicting amino acid changes that affect protein function. *Nucl Acids Res*, 31(13):3812–3814, 2003.

[29] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. pages 404–417, 2006. doi: 10.1007/11744023_32.

[30] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. *International Conference on Computer Vision*, pages 2564–2571, 2011.

[31] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. 1980.

[32] A. Hussain N. Ibrahim M. M. Mustafa L. Y. Siong, S. S. Mokri. Motion detection using lucas kanade algorithm and application enhancement. *International Conference on Electrical Engineering and Informatics*, pages 537–542, 2009.

[33] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[34] Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 19(6), June .

[35] H. C. Longuet Higgins. Readings in computer vision: Issues, problems, principles, and paradigms. *A Computer Algorithm for Reconstructing a Scene from Two Projections*, pages 61–62, 1987.

[36] Tomasz Malisiewicz. The future of real-time slam and deep learning vs slam. http://www.computervisionblog.com/2016/01/why-slam-matters-future-of-real-time.html, 2016.

[37] G. Bjntegaard A.Luthra T.Wiegand, G.J. Sullivan. Overview of the h.264/avc video coding. *Standard. IEEE Transactions on Circuits and Systems for Video Technology*, 3(7), July 2003.

[38] M. Piccardi. Background subtraction techniques: a review. *IEEE International Conference on Systems, Man and Cybernetics*, 4:3099–3104 vol.4, Oct 2004.

[39] Xuan Dai Pham, Jung Uk Cho, and Jae Wook Jeon. Background compensation using hough transformation. *2008 IEEE International Conference on Robotics and Automation*, pages 2392–2397, May 2008.

[40] Bertrand Vachon. Thierry Bouwmans, Fida El Baf. Background modeling using mixture of gaussians for foreground detection - a survey. *Recent Patents on Computer Science, Bentham Science Publishers*, 1(3):219–237, 2008.

[41] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, Jan 1998. ISSN 0162-8828. doi: 10.1109/34.655647.

[42] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. 1995.

[43] Van den Poel D. Prinzie A. Random multiclass classification: Generalizing random forests to random mnl and random nb. *DEXA 2. Lecture Notes in Computer Science*, 4653, 2007.

[44] Jerry Ye, Jyh-Herng Chow, Jiang Chen, and Zhaohui Zheng. Stochastic gradient boosted distributed decision trees. pages 2061–2064, 2009. doi: 10.1145/1645953.1646301. URL `http://doi.acm.org/10.1145/1645953.1646301`.

[45] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.