A new architecture for simultaneous localization and mapping: an application of a planetary rover

Kuo-Kun Tseng[a], Jun Li[a], Yachin Chang[a], K.L. Yung[b], C.Y. Chan[b] and Chih-Yu Hsu[c]

[a]Schoolof Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China;
[b]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong;
[c]Shool of Information Science and Engineering, Fujian University of Technology, Fuzhou, China

KEYWORDS
System design; simultaneous localisation and mapping; ORB; extended kalman filter; motion estimation; planetary rover

ABSTRACT

A new architecture implements one Monocular Simultaneous Localization and Mapping (SLAM) system to track the unconstraint motion of a mobile robot. The modified ORB (Oriented FAST and Rotated BRIEF) features represent the landmarks for designing a grid feature detection algorithm. An upgraded feature matching method has improved the robustness of feature matching. The Modified coVariance Extended Kalman Filter (MVEKF) estimates the multiple dimension states of the free moving visual sensor instead of the familiar Extended Kalman Filter (EKF). The simulation navigation of Lunar and Mars surfaces proves that the proposed method is robust and efficient.

1. Introduction

Simultaneous localisation and mapping (SLAM) is the process of creating a map of the environment while simultaneously deciding the position of the rover in the map. Besides, SLAM has been used and validated by a lot of researchers. SLAM has played an important role in the intelligent rover and other research fields. Rovers have many sensors, such as laser, sonar, or visual camera, has a large impact on the algorithm of SLAM. Sometimes, additional sensorial sources are used to better sense rover state and the surrounding environment (Aulinas et al. 2008). Compared with those sensors, the camera is not only cheaper in price and lighter in weight, but also more convenient to be used in some conditions. Recently, there is an increasing interest in using as a single visual sensor to perform the SLAM. Monocular SLAM (Scaramuzza and Fraundorfer 2011) is the SLAM using images captured by mono cameras.

Even though it is unable to recover depth information of the observed landmarks directly, Monocular SLAM is favoured in the literature of visual SLAM and has received much consideration in the last years, which requires less computing resource than Binocular or Panoramic SLAM. Monocular SLAM is closely related to the Structure-from-Motion (SFM) problem for recovering relative camera poses and three-dimensional (3D) structure from a set of camera images. SFM methods as off-line algorithms require batch, simultaneous processing for all the images acquired in the sequence. After obtaining

globally consistent reconstruction of the camera trajectory and scene structure, local motion estimation are refined using an offline global optimisation (i.e., bundle adjustment) through the whole sequence and computation time grows with the number of images. Different from SFM techniques, Monocular SLAM focuses on estimating the 3-D motion of the camera sequentially and in real-time. As a new image arrives, SLAM solutions can update the latest state of the camera quickly. As a result, for consistent localisation over long sequences in real-time, Monocular SLAM is more suitable than SFM.

An autonomous rover using SLAM perceiving maps of the outside world for the requirement in different higher level works. As a result, the first encounter problem is how to extract the important features from the visual cameras and build the map with the surrounding environment. Since the SLAM is usually feature-based descriptor of map, a lot of algorithms are applied to extract the features and create their descriptors as the landmarks in the map, ex: Shi-Tomasi (Davison 2003), Harris (Lemaire et al. 2007), SIFT (Se, Lowe, and Little 2002; Karlsson et al. 2005), SURF (Zhang et al. 2008; Wang, Hung, and Sun 2011), FAST (Civera et al. 2010; Klein and Murray 2007) and so on. Furthermore, after adding the features into the map of the system, the state of camera and all the features can be updated by the innovative information of feature matching between consecutive images. The extracted feature points used as landmarks should be robust under all kinds of changes such as scale, viewpoint and illumination changes, or else, it will make feature matching incorrectly. That is, if the observations cannot be correctly associated with the landmarks in the map, the map will be inconsistent. In this case, it is a data association problem. Wrong matching will cause wrong data association (Fraundorfer and Scaramuzza 2011). Therefore, it is crucial to choose suitable features on the captured images by the camera as reliable landmarks. Meanwhile, it is also significant to design a good feature matching algorithm to improve the data association. Such as, some new feature extraction algorithms provide good performance for SLAM.

The other problem is how to match the feature map to estimate the rover state. Until now, there are two famous algorithms in SLAM, one is filtering-based SLAM, and the other one is key-frame-based SLAM. These methods (Strasdat, Montiel, and Davison 2010b) may have an advantage with low resources but high accuracy. The Extended Kalman Filter (EKF) can solve the issue of estimation of SLAM. Besides, several modified filters that have better performance have been found to improve the estimation system.

Our research presents a new Monocular SLAM to estimate the visual sensor's states which it is an extended work by creating the map from sparse feature (Civera et al. 2010; Davison et al. 2007). Moreover, ORB (Rublee et al. 2011) is used to detect the feature which can greatly reduce the initialisation time in the system, and a new feature detection algorithm is proposed to get a suitable number of landmarks. Due to the linear error caused by EKF, we can apply the MVEKF (Guo 2003) to estimate the system state recursively.

As for the space exploration, Lunar rover plays an important role in exploring the moon or Mars, but because the moon or Mars can't have the earth like GPS, and GPS can't provide shelter and high-accuracy location. Therefore, the SLAM algorithm can be used to assist the positioning. It allows the lunar rover to remember its trajectory and provide a functional service for the lunar rover to return to the original point. Thus, we also applied our proposed algorithm to a space rover application.

Organisation of this paper is as follows. Section 2 introduces the related works of Monocular SLAM. Section 3 describes the detail of the proposed new architecture of

SLAM. Section 4 shows the applications of SLAM for an office and desert. Finally, we have a conclusion in section 5.

2. Related works

In the early 1980s, (Moravec 1980) presented the research of predicting a rover's movement from visual data. The work of Moravec includes the movement prediction approach without motion estimation solutions. Besides, he also proposed one of the earliest corner detectors called the Moravec corner detector, which is a predecessor of the popular corner detector known as (Harris and Stephens 1988). Building upon Moravec's work, both SFM and Visual SLAM have long-term development.

Nowadays, many research works focus on Monocular SLAM (Scaramuzza and Fraundorfer 2011; Fraundorfer and Scaramuzza 2011). Two systems among those Monocular SLAM systems are milestones in the development history, and other systems are almost improved versions based on them to some degree.

The first successful application of the SLAM methodology with one single camera is developed by Davison (Davison et al. 2007), which can operate at 30FPS and be capable of coping with an agile motion for an uncontrolled camera with the standard EKF framework. In Davison's system, point features are detected automatically using the detection operator of Shi and Tomasi. (Klein and Murray 2007) proposed the second famous system. Due to linearisation errors implicit in the filtering approaches, they advocated the optimisation of a sparse selection of images that are called key-frames instead of using a sliding window. Their system applied FAST corner detection to extract the features and performed a global bundle adjustment over the tracked features and a selected set of key-frames in the image sequence. Based on their works, many other researchers reported their improved methods to solve the issues in the Visual SLAM. In paper (Lemaire et al. 2007), Lemaire presented a robust interest point matching algorithm that can work in very diverse environments with the improved Harris corner detector. Considering the various changes of images, (Karlsson et al. 2005) proposed a method to extract the features by using the scale-invariant feature transform (SIFT) to provide a feature extractor. (Wang, Hung, and Sun 2011) proposed to apply the SURF in the algorithm to provide a better representation of a visual SLAM system. Besides, a tracking window and a nearest-neighbour algorithm are integrated to improve data association in SLAM. (Strasdat, Montiel, and Davison 2010b) presented a new architecture for visual SLAM with the keyframe optimisation method. Recently, (Civera et al. 2010) proposed a novel method to combine one-point RANSAC into the EKF, which can get more reliable data association. In Civera's system, a FAST algorithm is used to detect the positions of the landmarks.

Our research follows the works of (Davison et al. 2007) and (Civera et al. 2010) to make improvements. In this paper, we design a new Monocular SLAM system and propose some improved measures for the issues mentioned for current vision-based SLAM solutions. Experiments are carried out with our system to test its performance. Furthermore, we proposed a new Lunar rover location module with the improved algorithm.

3. Proposed architecture and algorithm description

3.1. System architecture

SLAM algorithms, in general, have four major steps – prediction, data association, measurement update, and augmentation. In the prediction step, the algorithm predicts the new state vector and covariance matrix from the previous state and covariance. The proposed architecture for monocular SLAM shown in Figure 1 has some parameters of the system to make an initialisation, which may be essential for the system. After that, the system begins to capture a new image into the loop. In the map management module, there are two important processes to be accomplished. One is removing unstable, and the other is initialising new features. The system map merges the new image into the map. Then, the system's predicted state for the next time is the motion model of the camera. At this moment, we load the new image into the estimation system and project all 6D features into 2D points to search the measurements in the new image plane through the measurement model of the camera. The search region of the measurements for feature landmarks can be calculated using the active search method proposed by Davison. After checking each observation data, the modified filter update the system state. Finally, the camera's new pose and the positions of sparse features in the map can be able to be estimated.

3.1.1. System initialisation

In the phase of initialisation processing, it mainly proceeds with the system parameterisation. In our SLAM system, the system state consists of two parts: the camera state and the positions of all the feature points in the map. Here we use W to specify the world coordinate frame which is defined by ourselves as a static reference frame, and

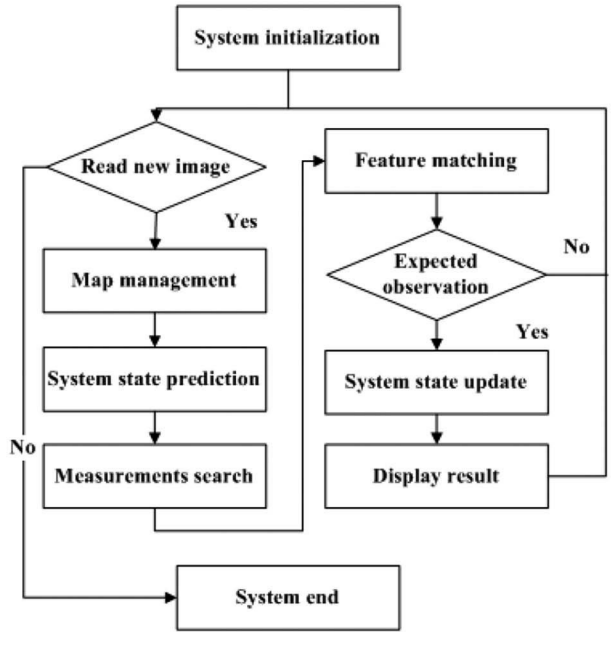


Figure 1. The proposed architecture of the SLAM system.

C stands for the visual motion coordinate frame. Figure 2 illustrates the process of system parameterisation. The denotations are as follows:

$$x_c \frac{1}{4} \; r^{WC}; q^{WC}; v^W; w^C \;^T \tag{1}$$

Where r^{WC} means the position of a motion camera; q^{WC} stand for the camera orientation of the world frame W; v^W , w^C is linear and angular velocities. Therefore, we can have:

$$r^{WC} \frac{1}{4} \text{½} x_r; y_r; z_r]^T \tag{2}$$

$$q^{WC} \frac{1}{4} \text{½} q_1; q_2; q_3]^T \tag{3}$$

$$v^W \frac{1}{4} \; v_x; \; v_y; \; v_z \;^T \tag{4}$$

$$w^C \frac{1}{4} \; w_x; \; w_y; \; w_z \;^T \tag{5}$$

For every feature in the map, the following state vector in the form of inverse depth parameterisation (Civera, Davison, and Montiel 2008):

$$f_i^W \frac{1}{4} \; x_i^W; y_i^W; z_i^W; \alpha_i; \beta_i; \rho_i \;^T \tag{6}$$

Where α_i β_i represent the azimuth and the elevation respectively for the defining the unit directional vector $m(\alpha_i; \beta_i)$; the point's depth along the ray d_i denotes by its inverse $\rho_i \frac{1}{4} 1/d_i$. The following is the unit directional vector:

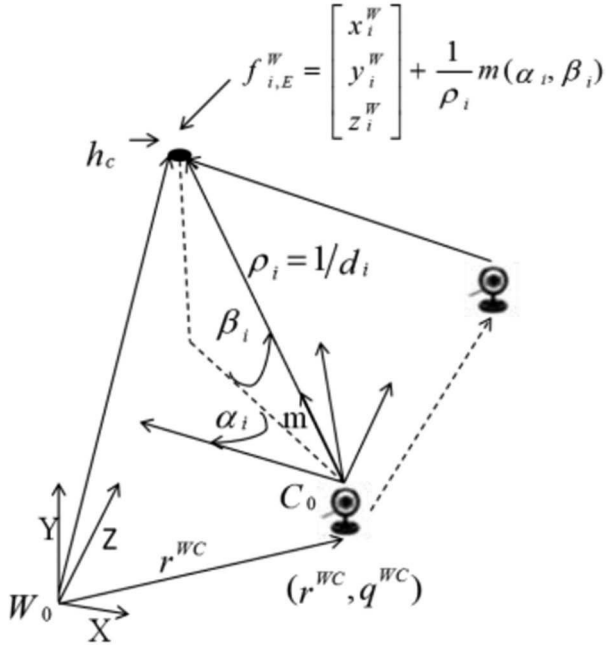


Figure 2. System parameterisation.

$$m(\alpha_i, \beta_i) = [\cos\beta_i \sin\alpha_i, -\sin\beta_i, \cos\beta_i \cos\alpha_i]^T \quad (7)$$

Furthermore, substitute the above 6D features into the 3D Euclidean point in the map:

$$f_{i,E}^{W} = \begin{bmatrix} x_i^W \\ y_i^W \\ z_i^W \end{bmatrix} = \begin{bmatrix} x_i^W \\ y_i^W \\ z_i^W \end{bmatrix} + \frac{1}{\rho} m(\alpha_i, \beta_i) \quad (8)$$

The next step, the system state vector included the following parameters:

$$x = [x_c, f_1^W, f_2^W, f_3^W, \ldots, f_N^W]^T \quad (9)$$

Meanwhile, the covariance matrix is as follow.

$$P = \begin{bmatrix} P_{xx}, P_{xf} \\ P_{fx}, P_{ff} \end{bmatrix}_T \quad (10$$

3.2. Map management

Map management handles adding new features or removing unwanted features that have been found to affect the results of the SLAM system. To provide a better performance, we propose a grid-based feature detecting strategy and modify the feature deleting criterion in the map.

3.2.1. The grid-based feature detection algorithm

Figure 3 shows the flowchart of our feature detection algorithm. We separate the image into M grids, and the ORB detector is applied to detect minimum features into the selected grid. The features are extracted randomly until many features are satisfied. The expression of a selection of the grid sub-region in the image is as follows:

$$grid_k = \begin{cases} randperm(1, 2, \ldots, M)_1 & k \ldots N \\ randperm(min_E(f_{k1}, f_{k2} \ldots f_{kM}))_1 & k > N \end{cases} \quad (11)$$

where $grid_k$ stands for the index of gird selected in the time k; randperm() is a function that can generate a random integer permutation, we need to select the first value from the permutation as the grid index; f_{kj} represents the number of features in each grid.

In Davison and Civera's research, they take the random region detection method to generate landmarks. Our method has a better distribution for the features over the whole image. The deletion criterion includes the ratio of the number of correct matching and matching attempts. If the ratio is below a threshold, the feature map deletes the landmark. However, this criterion has two shortcomings. Some occluded landmarks are removed using the ratio deletion criterion, and it will take the extra cost to re-initialise them when observed they again. It is impossible to recognise previous mapped areas for loop closing tasks is another shortcoming which will lead mapping to divergence if removed old features. We make some improvements to avoid the drawback of the ratio criterion. The experiment section shows improvements in the results.

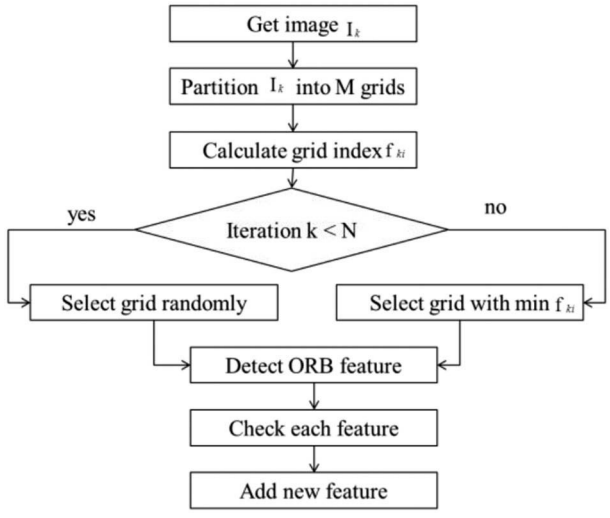


Figure 3. Flowchart of grid-based feature detection.

3.2.2. System state prediction with the camera motion model

The linear velocity and angular velocity (Davison et al. 2007) to describe the motion of a free movement visual sensor as the following:

$$x_{k+1} = g(x_k, u) = \begin{bmatrix} r_{k+1}^{WC} \\ q_{k+1}^{WC} \\ v_{k+1}^{W} \\ w_{k+1}^{C} \end{bmatrix} = \begin{bmatrix} r_k^{WC} + (v_k^W + n * V^W)\Delta t \\ q_{k+1}^{WC} \otimes q(w_k^C + n * W^C)\Delta t) \\ v_k^W + V^W \\ w_k^C + W^C \end{bmatrix} \tag{11}$$

Here V^W and W^C denoted as impulsive linear and angle velocity; $q(w_k^C + W^C)\Delta t)$ means the quaternion computed from the rotation vector$(w_k^C + W^C)\Delta t$; u denotes the changes in linear and angular velocity; m and n represent the dynamic parameters for acceleration velocities. Then we have

$$u = \begin{bmatrix} V^W \\ W \end{bmatrix} = \begin{bmatrix} a^W \Delta t \\ b^C \end{bmatrix} \tag{12}$$

here Δt represents the sampling time interval; a^W, b^C are all Gaussian variable with zero means.

Hence, the estimated system state for MVEKF is as follows.

$$\hat{x}_{k|k-1} = \begin{bmatrix} g(\hat{x}_{k-1|k-1}) \\ \hat{f}_1 \\ \cdots \\ \hat{f}_N \end{bmatrix} \tag{13}$$

$$P_{k|k-1} = G_x P_{k-1|k-1} G_x^T + G_u Q G_u^T \tag{14}$$

where G_x , G_u are the derivatives of motion model g, the camera state $\hat{x}_c$, and the control parameter u;Qmeans the noise of the covariance matrix.

3.2.3. Feature search and matching

In our Monocular SLAM system, the pinhole camera model with two parameters of radial distortion is also used to take the measurements of the landmarks. According to this model, we can project 6D features from inverse depth world coordinate frame W onto the camera coordinate frame C as the following equations:

$$h_i^C = \begin{bmatrix} x_i^C \\ y_i^C \\ z_i^C \end{bmatrix} = R^{CW}\left(q_i^{WC}\right)\left(\rho_i f_i^W - r_i^{WC}\right) + m(\alpha_i, \beta_i) \tag{15}$$

Where $R^{CW}(q_i^{WC})$ is a rotation matrix, and C is the frame from a camera, and the following is a pinhole model:

$$h_u = \begin{bmatrix} u_u \\ v_u \end{bmatrix} = \begin{bmatrix} u_0 - \frac{f x_i^C}{z_i^C dx} \\ v_0 - \frac{f y_i^C}{z_i^C dy} \end{bmatrix} \tag{16}$$

where f is a focal length of the camera and (u_0, v_0) is a centre coordinate of the image. *dx*, *dy* denote the unit size of each pixel. The transformation of the undistorted point with the radial distortion model and its distorted pixel coordinates are:

$$h_d = \begin{bmatrix} u_d \\ v_d \end{bmatrix} = \begin{bmatrix} u_0 + \frac{(u_u - u_0)(1 + k_1 r^2 + k_2 r^4)}{} \\ v_0 + \frac{(v_u - v_0)(1 + k_1 r^2 + k_2 r^4)}{} \end{bmatrix} \tag{17}$$

$$r = \sqrt{(d_x(u_d - u_0))^2 + (d_y(v_d - v_0))^2} \tag{18}$$

Section 3.2 describes the principle of active search and the predicted probability distribution for the system state, which can be used to find the correspondence search region automatically. From the function $h(x)$, an expected position $\hat{h}_i(u_i^E, v_i^E)$ of projection for the 6D feature f_i is the covariance matrix as:

$$\begin{aligned} \hat{h}_i &= h_i(\hat{x}_{k|k-1}) \\ S_i &= H_i P_{k|k-1} H_i^T + R_i \end{aligned} \tag{19}$$

Then, we obtain the bound of the search region as follow.

$$\begin{bmatrix} s_x \\ s_y \end{bmatrix} = \begin{bmatrix} 2n\sqrt{S_{i(1,1)}} \\ 2n\sqrt{S_{i(2,2)}} \end{bmatrix} \tag{20}$$

Where n is the number of the desired search region. To find a candidate z_i with a 95% probability is in the search region, as in Figure 4.

Although the feature matching process may be robust and reasonable with the active search principle, there still exist some improvements to enhance the robust and efficiency of feature matching by our experiments. Figure 5 is our feature matching method.

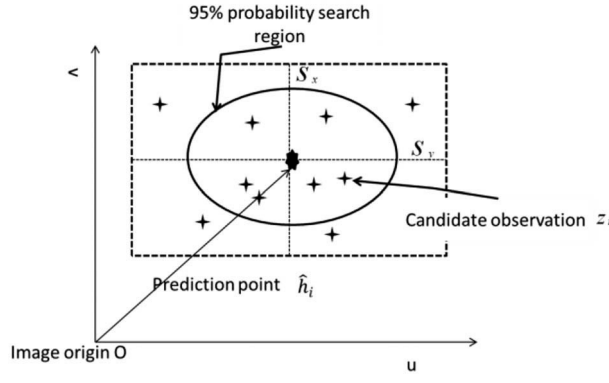


Figure 4. Feature search with active vision.

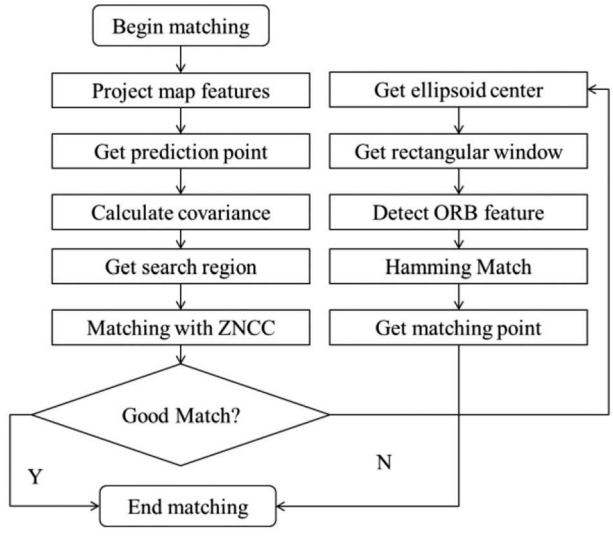


Figure 5. The flowchart of feature matching.

Depending on the situation of matching between system map features and new observations, the system will decide whether to use Hamming match with the descriptors extracted by ORB. This is the main improvement compared with the other matching method. When a camera has a drastic movement, the ratio of correct feature matches decrease largely, and sometimes even up to zero. In this case, it may cause the filter failure, and the results of the evaluation are necessary for Monocular SLAM.

3.3. System state update

The modified covariance EKF (MVEKF) (Guo 2003) compares with other usual filtering methods, ex: EKF, MGEKF, and IEKF. The idea of MVEKF to re-compute the Jacobi matrix H_k^b.

$$H_k^{\flat} = \frac{\partial h(x^{\flat})}{\partial x}\Big|_{x=\hat{x}_{kj}} \tag{21}$$

For state covariance matrix, it is updated by

$$H_k^{\flat} \; P_{k\,k-1}^{\flat} \quad 0 \tag{22}$$

When the expected observations are ready, then the parameters are updated as follows:

$$K_k = P_{kjk-1} H_k^T S_i^{-1} \tag{23}$$

$$\hat{x}_{kjk} = \hat{x}_{kjk-1} + K_k(z_i - \hat{h}_i) \tag{24}$$

$$K_k = P_{kjk-1} H_k^{\flat T} \left(H_k^{\flat} P_{kjk-1} H_k^{\flat T} + R_k\right)^{-1} \tag{25}$$

$$P_{kjk} = P_{kjk-1} - K_k S_i K_k^T \tag{26}$$

$$P_{kjk} = (P_{kjk} + P_{kjk}^T)/2 \tag{27}$$

Table 1 describes the whole motion estimation algorithm as follow.

Table 1. Motion estimation algorithm.

algorithm:	MVEKF State Update for Monocular SLAM
input:	System state variable $\hat{x}_{k-1|k-1}$ system state covariance matrix $P_{k-1|k-1}$
output:	System state variable $\hat{x}_{kjk}$ system state covariance matrix $P_{k\,k\,j}$
Step 1:	state prediction with MVEKF filter $\hat{x}_{kjk-1} = g(\hat{x}_{k-1|k-1})$ $P_{kjk-1} = G_x P_{k-1|k-1} G_x^T + Q$
Step 2:	Calculate the Jacobi matrix of the measurement function to system state $H_k = \frac{\partial h(x)}{\partial x}\Big\|_{x=\hat{x}_{kjk-}}$ Calculate Kalman coefficient $K_k = P_{k\,k-1} H_k^T \left(H_k P_{k\,k-1} H_k^T + R_k\right)^{-1}$
Step 3:	System state update with MVEKF filter $\hat{h}_i = h_i(\hat{x}_{kjk-1})$ $S_i = H_i P_{kjk-1} H_i^T + R_i$ $\hat{x}_{kjk} = \hat{x}_{kjk-1} + K_k(z_i - \hat{h}_i)$ $P_{kjk} = P_{kjk-1} - K_k S_i K_k^T$
Step 4:	Modify the measurement Jacobi matrix $H_k^{\flat} = \frac{\partial h(x)}{\partial x}\Big\|_{x=\hat{x}_{kjk}}$
Step 5:	Repeat step 3, adjust system state matrix as a symmetric matrix $K_k = P_{kjk-1} H_k^T S_i^{-1}$ $\hat{x}_{kjk} = \hat{x}_{kjk-1} + K_k(z_i - \hat{h}_i)$ $K_k = P_{k\,k-1} H_k^{\flat} H_k^{\flat} P_{k\,k-1} H_k^{\flat} + R_k^{-}$ $P_{kjk} = P_{kjk-1} - K_k S_i K_k^T$ $P_{kjk} = (P_{kjk} + P_{kjk}^T)/2$
Step 6:	return new $\hat{x}_{kjk}$ and P_{kjk}

4. Evaluation

All the experiments in the following were run on the Pentium(R) Dual-Core T4500 processors at 2.30GHz. We have implemented three Monocular SLAM algorithms for testing their performance. The first algorithm is called 1PRMSLAM (short for 1-point RANSAC for EKF SLAM), which is proposed by (Civera, Davison, and Montiel 2008). Their main contribution is to propose a combination of one-point RANSAC, which allows the minimal sample size to be reduced to one, resulting in large computational savings without the loss of discriminative power for outlier rejection. The second algorithm is called CSEKFMSLAM, which uses SIFT to detect the image features as useful landmarks, and the other blocks are the same as the 1PRMSLAM. The purpose of building this algorithm is to test the effect and make a comparison with other algorithms when we change image feature detectors. The MVMSLAM denotes our proposed Monocular SLAM algorithm. This algorithm combines with the above improvements, and we will make a comprehensive comparison with the other two SLAM algorithms.

4.1. The experiment in an office

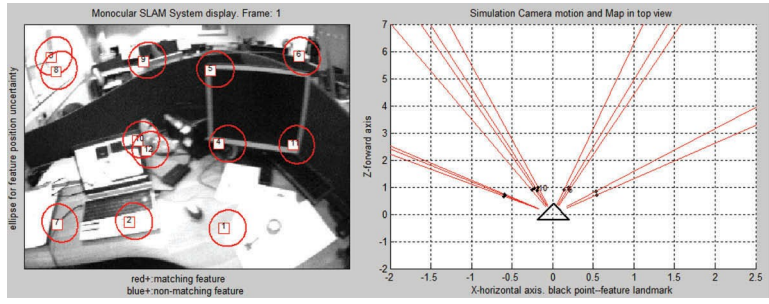
The experiment uses the videos published by the Civera (xxxx) and per second has 15 sequence images with the size 320×240. In this experiment with the same parameters, we selected 200 frames to test 1PRMSLAM, SEKFMSLAM, and our method-MVMLAM. Figure 6. displays the running results of our system. In the right side, the trajectory of the camera with a black line, and each feature (landmark) in the map is a black point with a corresponding red ellipse that represents the uncertainty of its position. From Figure 6, the uncertainties of most features are decreasing gradually, and the red ellipses are collapsed to the black points finally. The proposed architecture is feasible to solve the SLAM problem, and the whole estimation trajectory of the camera approximates the true trajectory of the camera.

Three algorithms use the public image datasets as the benchmark for evaluation. From Figure 7, it is obvious that the time that 1PRMSLAM cost is almost three times more than our system. Furthermore, we record their average time for each frame. As Table 2 shows, the average time of our system is 0.1799 s, while 1PRMSLAM and SEKFMSLAM take 1.5931s, 0.1901s respectively.

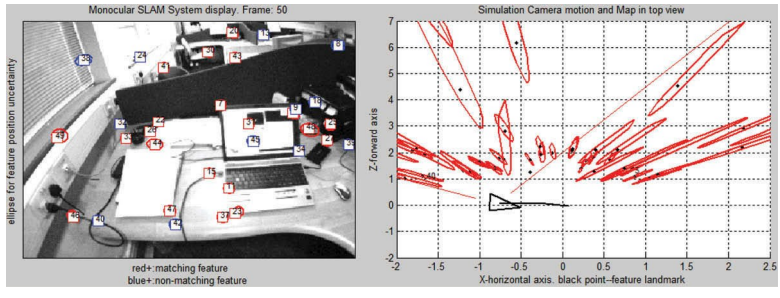
Figures 8 and 9 show the errors of camera trajectory estimation in our system. We calculate the root mean square error to show the estimation performance between them. Table 3 shows the error for 6D pose of our method SEKFMSLAM and MVMSLAM. The maximal space location error is 0.2005m while the minimal error is 0.0192m. Meanwhile, errors for camera orientation are all small, and these errors accepted by the 95% confidence regions. This estimation result proves that our algorithm, including SEKFMSLAM, is effective to track the unconstraint 6D motion. For a better performance showing, we have done a lot of test with our images. We will focus on the following experiment, which can test the final performance for the three methods.

4.2. Application for planetary rover

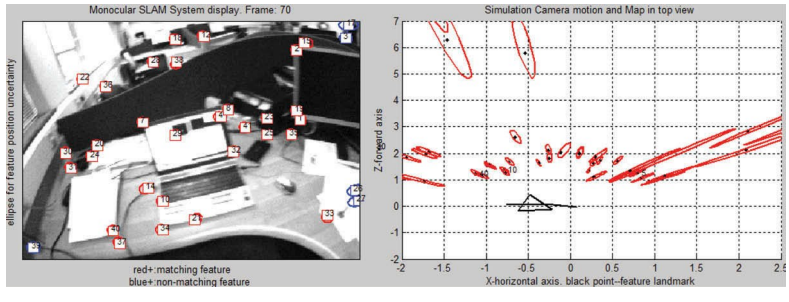
Visual SLAM (Zhou et al. 2015) and Planetary rover (Fallah, Yue, Vahid-Araghi and Khajepour et al. 2013) are both popular topics for vehicular Technology. Lunar rover is



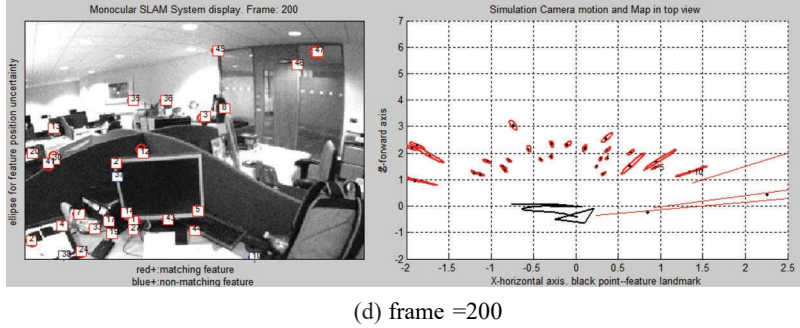
(a) frame =1



(b) frame =50



(c) frame =160



(d) frame =200

Figure 6. Monocular SLAM system display and camera motion in top view.

currently the most direct tool for lunar exploration, which mainly includes locating, path planning, obstacle avoidance, and motion control module. In the modules motioned above, localisation module is one of the most critical modules in all modules motioned

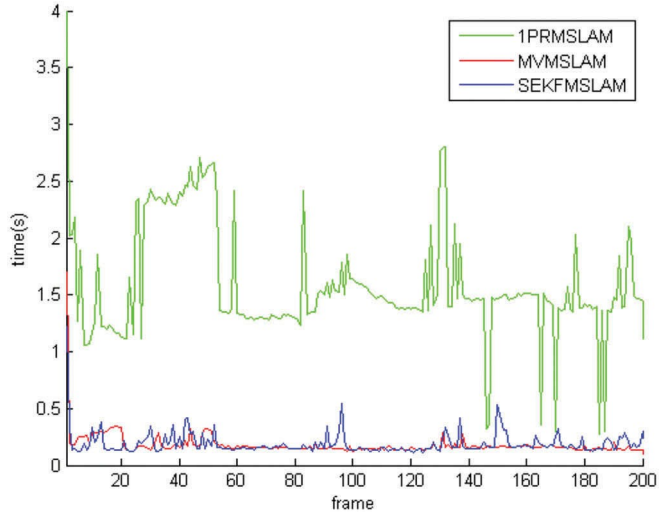


Figure 7. Time comparison of SLAM for frame processing.

Table 2. Average time for each frame.

Time	1PRMSLAM	MVMSLAM	SEKFMSLAM
Frame(s)	1.5931	0.1799	0.1901

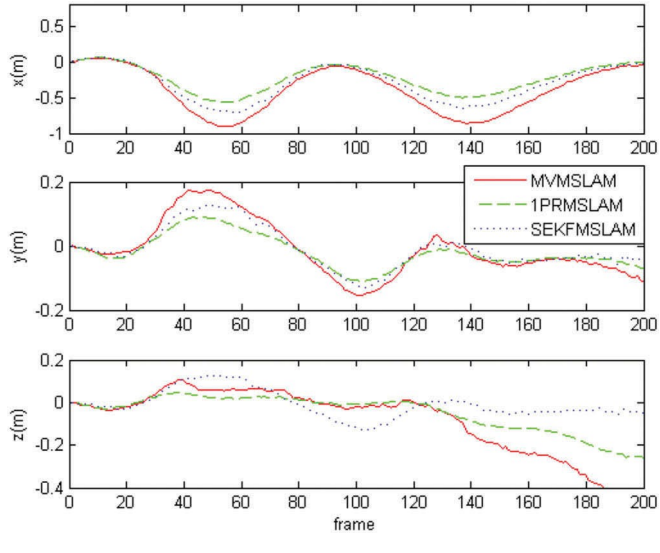


Figure 8. Error for camera location (XYZ).

above, its role is to calculate the real-time motion status, which indicates the position and attitude information of the rover. The rover position information is the premise of path planning and the important input data of obstacle avoidance and other motion control

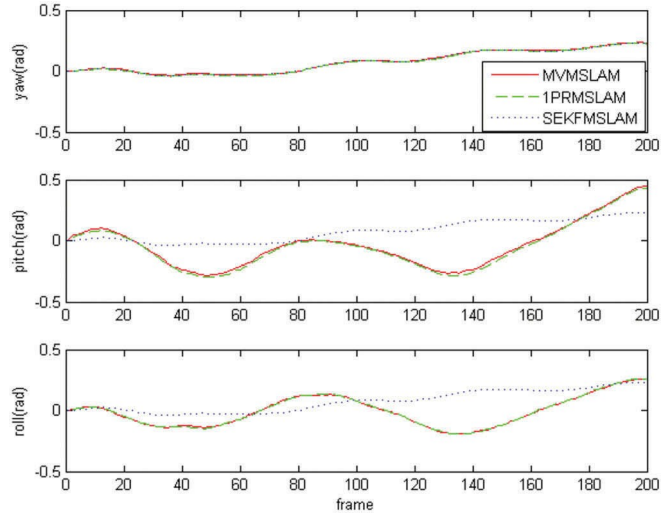


Figure 9. Error rate of monocular camera orientation.

Table 3. Error for 6D pose.

Pose	SEKFMSLAM	MVMSLAM
X(m)	0.0867	0.2005
Y(m)	0.0193	0.0367
Z(m)	0.0192	0.0854
Yaw(m)	0.0038	0.0049
Pitch(m)	0.1001	0.0968
Roll(m)	0.0064	0.0043

modules. Therefore, real-time rover locating has important research significance for the lunar exploration mission.

Currently, well-rounded navigation technologies are Dead reckoning, visual navigation, radio navigation. Dead reckoning often relies on the odometer or another unit for navigation, but the fatal drawback of dead navigating is the existence of cumulative errors increase as time goes on. Therefore, this location method isn't suitable for long-term Lunar exploration mission. One characteristic of the Moon is that it has no atmosphere. The Moon has no atmosphere. Ultrasonic sensors are useless. The magnetic field of the Moon is very weak in comparison to that of the Earth. A magnetic compass would not work on the Moon because the magnetic field of the Moon is very weak in comparison to that of the Earth. The magnetic compass for navigation is useless on the moon. The GPS system operates using 50 or so satellites in orbit around the Earth. They would not be useful on the Moon even though their signals were possibly received. The Earth's GPS positioning system cannot provide lunar rover navigation services. Thus, the most feasible method for Lunar rover locating is visual navigation. Based on our proposed MVMSLAM, we propose a lunar rover locating module, as shown in Figure 10.

A monocular camera as a sensor can percept the moon's surface environment. The real-time image sequence captured during the rover movement is used as the input of the module to realise the real-time position and attitude information calculation of the rover in an unknown environment. Meanwhile, Map, build a good environment map can be used as follow-up path planning and other high-level exploration mission services. The

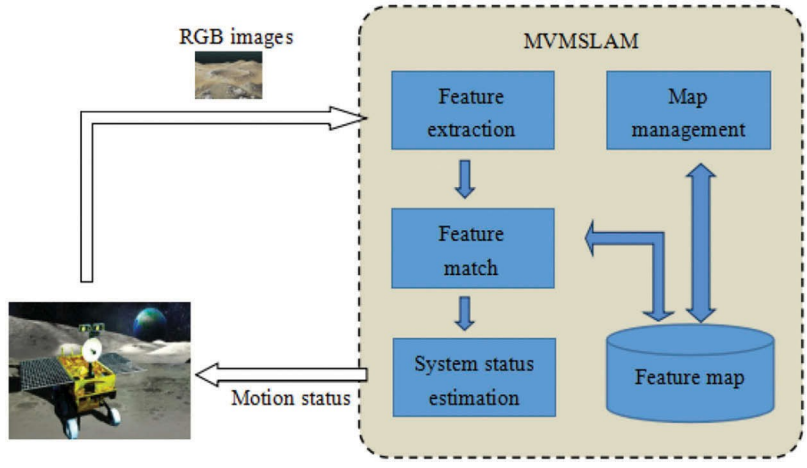


Figure 10. Lunar rover locating module.

module mainly includes improved grid-based feature extraction, rule matching, and optimised feature map management algorithm.

To validate the feasibility of the module, we perform a simulated experiment for the Lunar rover. Since the difficulty of obtaining an image sequence of the lunar surface environment, we adopt the desert landscape on earth to simulate the lunar surface environment. Desert topography visual field above the open, and less blocking within the visual field, the surface features are mostly gravel and undulating surface, these characteristics are more similar to the lunar environment. Using a head-mounted monocular camera, we took a 36-s long monocular sequence of 1140 frames and ran it on MVMSLAM. Figure 11 shows the results.

In Figure 11, a) the distribution of feature points corresponding to a certain frame in the image sequence, b) the distribution of the constructed map feature points c) the 3D trajectory map estimated by the module.

Experiments show that the lunar rover module based on our proposed MVMSLAM can provide rover with real-time location information and the feature map of an unknown environment. The output of the locating module mainly includes the motion trajectory and the feature map. The motion trajectory shows the motion route of the rover to return

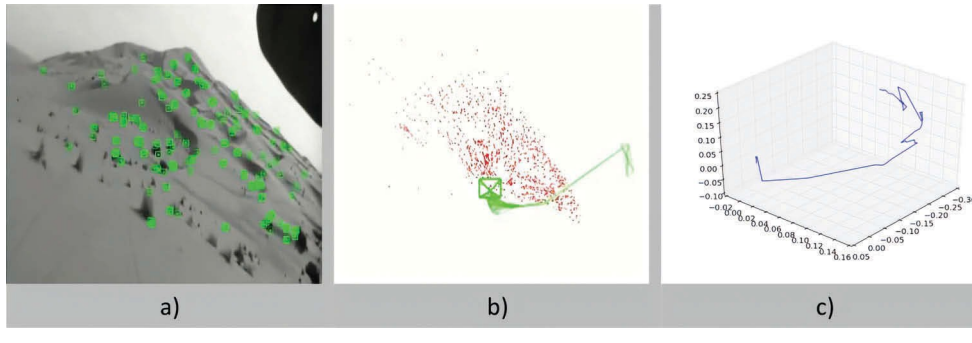


Figure 11. Experiment results.

the base station. The feature map can cooperate with the path planning module to accomplish more autonomous exploration tasks.

5. Conclusion and future work

This research presents a new approach MVMSLAM to design a visual SLAM system for the rover with a single camera, which is named MVMSLAM. In MVMSLAM system, firstly, a modified ORB is proposed to enhance the feature point extraction with grid-based strategy. Furthermore, to maintain the satisfying process speed for the SLAM application, the ellipse search algorithm is improved with a better feature detection approach. Besides, we do our best to optimise the MVEKF filter for the SLAM architecture. The experimental results show that our proposed new architecture is useful with satisfying performance and low error rate in the rover SLAM system.

In the end, we apply a Lunar rover location system with our proposed MVMSLAM. In the simulated experiment for the Lunar rover's locating task, the result shows the MVMSLAM should be capable of the Lunar rover exploration task.

In the future, with the development of autonomous driving rover, the demand for multi-task (MatariĆ et al. 2003) and multi-vision sensors (Civera 0000) is increasing. So how to design and optimise the SLAM architecture for more advanced autonomous driving rover in the future will be our future research focus.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Shenzhen Government [Grant[JCYJ20160531191837793, KQJSCX201707261040333]; Shenzhen Development and Reform Commission through Shenzhen Development and Reform [Grant(2016)889]. This research is also partially supported by a grant from the Department of Industrial and Systems Engineering at the Hong Kong Polytechnic University (project code: H-ZG3K).

References

Aulinas, J., Y. Petillot, J. Salvi, and X. Llado, "The SLAM Problem: A Survey," Artificial Intelligence Research and Development, In Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence, CCIA, October 22-24, Sant Martí d'Empúries, Spain. *Frontiers in Artificial Intelligence and Applications* 184: pp. 363–371. IOS Press. ISBN 978-1-58603-925-7

Civera, J. 0000. "Publications." http://webdiis.unizar.es/~jcivera/publications.html

Civera, J., A. J. Davison, and J. M. M. Montiel. 2008. "Inverse Depth Parametrization for Monocular SLAM." *IEEE Transactions on Robotics* 24 (5): 932–945. doi:10.1109/TRO.2008.2003276.

Civera, J., O. G. Grasa, A. J. Davison, and J. M. M. Montiel. 2010. "1-point RANSAC for EKF Filtering: Application to Real-Time Structure from Motion and Visual Odometry." *Journal of Field Robotics* 27 (5): pp. 609–631. doi:10.1002/rob.20345.

Davison, A. J. 2003. "Real-time Simultaneous Localisation and Mapping with a Single Camera." *International Conference on Computer Vision* 2: 1403–1410.

Davison, A. J., N. D. Molton, I. D. Reid, and O. Stasse. 2007. "Monoslam: Real-time Single Camera SLAM." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (3): 1052−1067. doi:10.1109/TPAMI.2007.1049.

Fallah, S., Yue, B., Vahid-Araghi, O. and Khajepour, A. 2013. "Energy Management of Planetary Rovers Using a Fast Feature-Based Path Planning and Hardware-in-the-Loop Experiments." In *IEEE Transactions on Vehicular Technology* 62(6): 2389−2401. DOI:10.1109/TVT.2013.2244624.

Fraundorfer, F., and D. Scaramuzza. 2011. *Visual Odometry: Part II Matching, Robustness, Optimization, and Applications*. IEEE Robotics & Automation Magazine.

Fucheng, G., Zhongkang, S. and Kan, H. 2003. "A Modified Covariance Extended Kalman Filtering Algorithm in Passive Location." *Proceeding of IEEE International Conference on Robotics, Intelligent Systems and Signal Processing* 1: pp. 307−311.

Harris, C., and M.Stephens. 1988. "A Combined Corner and Edge Detector." In *Proc. Alvey Vision Conf AVC 1988, Manchester, UK, September.* pp. 147−151. https://dblp.org/db/conf/bmvc/avc1988

Karlsson, N., E. D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. 2005. "The vSLAM Algorithm for Robust Localization and Mapping." *Proceedings of the 2005 International Conference on Robotics and Automation, Barcelona, Spain* pp. 24−29. doi: 10.1109/ROBOT.2005.1570091

Klein, G., and D. Murray. 2007. "Parallel Tracking and Mapping for Small AR Workspaces." In Proc. of Int. Symp. Mixed and Augmented Reality (ISMAR'07, Nara), 225−234.

Lemaire, T., C. Berger, I. Jung and S. Lacroix. 2007. "Vision-based SLAM: Stereo and Monocular Approaches." *International Journal of Computer Vision* 74(3): 343−364. doi:10.1007/s11263-007-0042-3.

Matarić, J. M., G. S. Sukhatme, H. Ostergaard, and Esben. 2003. "Multi-robot Task Allocation in Uncertain Environments." *Autonomous Robots* 14 (2−3): 255−263. doi:10.1023/A:1022291921717.

Moravec, H. 1980. "Obstacle avoidance and navigation in the real world by a seeing robot rover." Ph.D. dissertation, Stanford, CA: Stanford Univ

Rublee, E., V. Rabaud, K. Konolige, and G. Bradski. 2011. "ORB: An Efficient Alternative to SIFT or SURF." *International Conference on Computer Vision*, *Barcelona*, pp. 2564−2571. doi: 10.1109/ICCV.2011.6126544

Scaramuzza, D., and F. Fraundorfer. 2011. "Visual Odometry, Part I: The First 30 Years and Fundamentals. " *IEEE Robotics & Automation Magazine* 18(4): pp. 80−92. doi: 10.1109/MRA.2011.943233

Se, S., D. Lowe, and J. Little. 2002. "Mobile Robot Localization and Mapping with Uncertainty Using Scale Invariant Visual Landmarks." *International Journal of Robotics Research* 21(8): pp. 735−758. doi:10.1177/027836402761412467.

Strasdat, H., J. Montiel, and A. J. Davison. 2010a. "Scale Drift-aware Large scale Monocular SLAM." In *Proceeding of Robotics Science and Systems, Zaragoza, Spain.*

Strasdat, H., J. M. M. Montiel, and A. J. Davison. 2010b. "Real-time Monocular SLAM: Why Filter." *International Conference on Robotics and Automation, Anchorage, AK* pp. 2657−2664. doi: 10.1109/ROBOT.2010.5509636

Wang, Y.-T., D.-Y. Hung, and C.-H. Sun. 2011. "Improving Data Association in Robot SLAM with Monocular Vision[J]." *Journal of Information Science and Engeering* 27: 1823−1837.

Zhang, Z., Y. Huang, C. Li, and Y. Kang. 2008. ""Monocular Vision Simultaneous Localization and Mapping Using SURF." *World Congress on Intelligent Control and Automation* 1651−1656.

Zhou, H., D.Zou, L.Pei, R.Ying, P.Liu, and W.Yu. 2015. "Structslam: Visual Slam with Building Structure Lines." *IEEE Transactions on Vehicular Technology*64(4): 1364−1375. doi:10.1109/TVT.2015.2388780.