Slam algorithm for aruco landmark array based on synchronizition optimization

B Y Xing, R N Dang, P Xu, C X Jiang and L Jiang

China North Vehicle Research Institute, Beijing, 100072, China

E-mail: feist201@qq.com

Abstract. High precision mapping algorithm is the core of two-dimensional(2D) landmark array( the ArUco landmark is used in this paper) localization application. In most of the existing mapping algorithms, the accumulated errors of online methods can not be eliminated, meanwhile, the manual calibration and off-line methods are inefficient and time-consuming. In order to overcome those shortcomings, a real-time simultaneous localization and mapping(SLAM) algorithm for landmark array based on synchronized optimization is proposed in this paper. The proposed algorithm updates and corrects the global map online, and the accumulated errors is optimized based on minimizing the re-projection error of the landmarks’ vertexes. In order to verify the proposed algorithm, several contrast experiments with other mapping systems are presented. The experimental results show that the proposed algorithm has the advantages of high map precision and small accumulative error.

1. Introduction

Landmark localization is the most commonly used method in machine vision. It is easier to identify and more reliable than the natural feature, meanwhile, its recognition algorithm is less computational and faster. Therefore, the real-time performance can also be guaranteed on low-cost embedded processor.

There are a lot of landmark systems proposed in the recent years, such as ArUco [1], AprilTag [2] and Whycon [3]. The landmark based localization system can be built in the following three methods: (1) Single landmark location is commonly used in UAV localization and autonomous landing, such as [4-5]. Although this system is simple and easy to realize, but it has the disadvantages of large dead zone and small location range. (2) Compound landmark location uses several landmarks tightly combined to realize the continuous localization in a wide range. In [6], the author adds two small size landmarks to the blank area of the ArUco landmark. In [7], the author uses several small size landmarks around a large one to provide reliable localization data. (3) Landmark array location realizes the precise localization in large space with dozens or hundreds of landmarks, which meets the demand of high precision and low cost localization system nowadays [8].

The core of landmark array localization is its precise map. The maps in [8] and [9] are obtained by manual calibration, but they are inefficient and time-consuming. Therefore, many researchers proposed to use the 3D reconstruction algorithm to realize the automatic mapping. The current automatic mapping system can be divided into offline and online methods. The off-line method generates high accuracy map with complete video and long processing time. For example, the system proposed in [10] can build a precise map based on the camera’s trajectory obtained from visual odometer. The system proposed in [11] effectively reduce the accumulated errors of map by
minimizing the reprojection of landmarks’ vertexes in multi-frame. The online method uses the real-
time recognition results of landmarks to update and correct the map in real time. For example, in [12],
an online mapping system based on extended Kalman (EKF) is implemented. However, its covariance
matrix inverse process requires a large amount of calculation. The system proposed in [13] estimates
the new landmark ’s pose based on the constraint in coordinate transformation. However, the
accumulated errors of map are relatively large due to the lack of global optimization. In [14], an on-
line mapping algorithm based on graph model is proposed, in which the new landmark’s pose can be
calculated by finding the shortest path in the graph. Although the algorithm is easy to implement, it
still cannot effectively reduce the accumulated errors of the map. In summary, there is still room for
further improvement in the current automatic mapping system. Therefore, a real-time mapping
algorithm is proposed in this paper, which is shown in figure 1.

![Figure 1. Synchronized Optimization Framework.](image)

Figure 1 shows the proposed synchronized optimization framework, in which contains three parts.
The online mapping updates and corrects the global map in real time, meanwhile, the key frame
sequence is captured during the online mapping process. The global optimization uses the map
generated by the online mapping tread as the initial pose graph to optimize the accumulated errors.
Finally, the optimized map is synchronized to the global map. In order to verify the performance of the
proposed algorithm, several comparison experiments are presented in this paper and the results show
that the proposed algorithm achieves high-precision map and real-time localization.

This paper is arranged as follows: The online mapping algorithm is presented in 2 Section , the
global optimization algorithm and the synchronized optimization algorithm is presented in 3 Section.
The experimental results and analysis are given in Section 4 and the summary of the paper is presented
in Section 5.

2. Online Mapping Algorithm

2.1. Transform matrix mean value method based on nonlinear weighting
Define the landmark coordinate system as \( \{m\} \), define the global coordinate system as \( \{n\} \). The rotation
matrix \( R \) and translation vector \( t \) of the camera or the landmark can be represented as the pose matrix
\( T = [R | t] \) in SE(3). Thus, the any landmark’s pose matrix can be estimated based on the closed principle
of coordinate system transformation(CPT) in ideally. In order to improve the map accuracy, remove the
outliers and decrease the errors caused by PNP algorithm, a matrix array is built in this paper, which
stores all the estimated poses matrixes from different landmarks, and the mean value of the array is used
to correct the global map. In order to simplify the algorithm, the mean value of rotation matrix and
translation vector is calculated respectively. For the translation vector, the mean value can be obtained
by calculating its arithmetic mean:

\[
\bar{t}_n = \frac{1}{n} \sum_{i=1}^{n} t_i
\]  

(1)
For the rotation matrix array \( \mathbf{R} = \{ \mathbf{R}_i, i = 1, \ldots, n \} \) only its \( L_2 \) mean value \( \mathbf{R}_m \) is used, which can be obtained by solve the following unconstrained minimization problem:

\[
\arg\min_{\mathbf{R} \in \text{SO}(3)} \sum_{i=1}^{n} \| \mathbf{R}_i \mathbf{R}_m \|_2
\]

(2)

In order to solve the above problem, the correction vector \( \mathbf{r} \) is calculated in the \( \text{SO}(3) \) and obtains its relative rotation matrix \( \mathbf{R}_m \) with the Rodrigues transform method [15]. The vector \( \mathbf{r} \) can be calculated as follows:

\[
\mathbf{r} = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{w_i}{\sum_{k=1}^{n} w_k} \arcsin \left( \frac{\| \mathbf{y} \|}{\| \mathbf{y} \|_2 + \sigma} \right) \right)
\]

(3)

Where \( \sigma \) is a regularization parameter, and the vector \( \mathbf{y} = (y_1, y_2, y_3) \) can be calculated by Eq. (4).

\[
\mathbf{R}_m \mathbf{R}_i - \mathbf{R}_m = \begin{bmatrix} 0 & -y_1 & y_2 \\ y_1 & 0 & -y_3 \\ -y_2 & y_3 & 0 \end{bmatrix}
\]

(4)

Where \( d_i \) is the distance between the camera and the landmark. \( w_i = e^{-d_i^2/(2\sigma^2)} \) is the nonlinear weight, which ensures the results have higher fusion weight when \( d_i \) is small. The mean value algorithm \( \text{Mean}_{L_2}(\cdot) \) is given in table 1.

**Table 1. Nonlinear Mean Value Algorithm.**

| Input: | \( T_i (i = 1, 2, \ldots, n) \): Transformation matrix array \( \delta \): Threshold \( M \): Maximum number of iterations |
|---------|---------------------------------------------------------------------------------------------------|
| Output: | \( T_m \): Mean value |

/*---*---Split the Transformation Matrix---*---*/

for \( i = 1 \) to \( n \) do

Obtaining the \( \mathbf{R}_i \) and \( t_i \) from \( T_i = \begin{bmatrix} \mathbf{R}_i & t_i \\ 0_{3,1} & 1 \end{bmatrix} \subseteq \text{SO}(3) \);

end for

/*---*---Calculating the Mean Value of Translation Vector---*---*/

\[
\mathbf{t}_m = \frac{\sum_{i=1}^{n} t_i}{n} ;
\]

/*---*---Calculating the Mean Value of Rotation Matrix---*---*/

Initial the Mean Value: \( \mathbf{R}_m = \mathbf{R}_i \);

for \( i = 1 \) to \( M \) do

Calculating the correction vector \( \mathbf{r} \) based on Eq. (3);

if \( \| \mathbf{r} \| < \delta \) then

return \( \mathbf{R}_m \);

else

\( \mathbf{R}_m = \mathbf{R}_m \exp(\mathbf{r}) \);

end if

end for
2.2. Online mapping algorithm

In order to realize the online mapping and correction the following algorithm is proposed here. The coordinate system \( \{n\} \) is initialized by the first recognized landmark. The new landmark’s pose is estimated based on the CPT method. Ideally, the camera’s pose \( Tc^* \) in \( \{n\} \) calculated by any landmarks should be consistent. Thus, the pose \( T_j \) of the landmark \( j \) can be estimated by the existing landmark \( i \)’s pose \( T_i \) when they are recognized in one frame:

\[
Tc^* = T_i Tc^n_i = T_i Tc^n_j
\]

(5)

\[
\hat{T}_{j,i} = (Tc^n_j)^t = \sum_{k=1}^{N} w_k \xi_k
\]

(6)

Where \( Tc^n_j \) and \( Tc^n_i \) is the pose of the camera under each landmark’s \( \{n\} \) system. According to Eqs. (5)–(6), the pose of new landmark can be corrected and updated online. In order to improve the accuracy of the map and eliminate the distortion caused by false recognition, a matrix array \( \xi_j \) stores all the \( \hat{T}_{j,i}(t_j) \) calculated by another landmarks is build here, and its mean value \( T_n(j) \) is used to realize the local correction. Taking landmark \( j \) as an example, the outliers are eliminated by using Grubbs criterion. In order to ensure \( \xi_j \) can store the historical correction information, an amplitude variable \( f_j \) is constructed based on the Frobenius norm of \( T_n(j) \). After sorting up \( T_n(j) \) with \( f_j \), the first \( 1/\delta \) elements in \( T_n(j) \) are retained as the historical information, meanwhile, the landmark is fixed in the map when \( f_j \) is less than the threshold \( \tau \). Thus, the proposed online mapping algorithm is shown in figure 2.

![Flow Chart of the Online Mapping Algorithm](image)

Figure 2. Flow Chart of the Online Mapping Algorithm.

For simplicity, the camera’s pose \( Tc^*(k = 1, ..., N) \) in \( \{n\} \) are represented in 3D position and Euler angle \( \xi e^*(k = 1, ..., N) \). Meanwhile, the pose of the camera can be calculated with all the recognized landmarks:

\[
\xi e^* = \sum_{k=1}^{N} \theta_k \xi e^*_k / N
\]

(7)

Where \( \theta_k = e^{-d_k/(2\epsilon^3)} \) is the nonlinear weight, \( \epsilon \) is the adaptive parameter, and \( d_k \) is the distance between camera and landmark. The key frame is captured after every move a fixed distance.

3. Global Optimization and Synchronized Algorithm

In order to optimize the accumulate errors in global map. The following global optimization and synchronized algorithm is proposed when the key frame sequence is full, which improve the map
accuracy by minimizing the re-projection error of landmarks’ vertexes. Define the dimension of the landmark $j$ as $s$, and its vertexes in $\{m\}$ system can be seen in figure 3.

Define $\delta = (f, f_c, e_c, e, k_1, \ldots, k_4)$ is the inner parameters of the camera, which includes the focal length, the centre of light position and the distortion parameter. The projection pixel position of each vertex is:

$$u_j = \{\psi(\delta, T_j, p^{'j}), i = 1, 2, 3, 4\}$$

Where $\psi(\cdot)$ is the re-projection function, the objective of the global optimization is to minimize the re-projection error of all vertexes in the key frame sequence, thus the optimization functions is:

$$f(x) = \sum_{j=1}^{m} \sum_{i=1}^{n} e_i$$

$$e_i = \sum_{k=1}^{4}(\psi(\delta, T_j, p^{'j}) - \hat{u}_j), k = 1, 2, 3, 4$$

Where $m$ is the number of key frames, $n$ is the number of landmarks, and $\hat{u}_j$ is its pixel projection. The global optimization algorithm is given as following:

3.1. Initial Pose Graph

The algorithm directly uses of key frame sequence overflow time to construct the initial pose graph $G(t)$, which is shown in figure 4.

In figure 4, the global map $M(t)$ stores the transformation matrix $T_i$ of all the existed landmarks. The key frame is interval sampling during the mapping process, thus the sequence may not contain all the landmarks in the global map $M(t)$. As can be seen in figure 3, the key frame sequence has 3 frames and the $M(t)$ containing No. 1–4 landmarks. However, landmark No.4 isn’t recognized in the sequence. Therefore, the initial pose graph $G(t)$ can only build based on the recognition result. Thus, frame 1 can construct the transformation matrix $T_{1,3}$, frame 2 can construct the $T_{2,3}$ and $T_{2,1}$. Finally, the initial
pose graph \( G(t) \) can be build and used for the global optimization. The transformation matrix between two existed landmarks can be calculated with their pose in \( M(t) \) and the following formula:

\[
\begin{align*}
T_{i,j} &= g(T_i, T_j) = (T_j)^{-1} T_i \quad (i, j = 1, 2, 3) \\
T_{j,i} &= (T_i)^{-1} T_j \quad (i, j = 1, 2, 3)
\end{align*}
\]  

(11)

3.2 Global optimization algorithm

For the global optimization, the \( G(t) \) is optimized by minimizing the re-projection error of the landmarks’ vertexes. In ideal case, the loop shown in Fig. 3 should satisfy the CPT method.

\[
I_{4s} = T_{k1} T_{32} T_{21}
\]  

(12)

However, an error matrix \( E \) exists in the actual rotation matrix.

\[
I_{33} = R_{31} R_{12} R_{23} E
\]  

(13)

In order to simplify the optimization, the rotation matrix and the translation vector are independently optimized. For the rotation matrix the error matrix \( E \) is assigned into the loop with different weight [11]:

\[
R_{31} E_{31}^{e1} R_{12} E_{12}^{e2} R_{23} E_{23}^{e3} = I_{33}
\]  

(14)

\[
E_{ij}^{e} = \exp(\alpha \ln E)
\]  

(15)

Where \( \alpha_{ij} (0 < \alpha_{ij} < 1) \) is a certain weight, which is the mean value of two nodes’ re-projection error. Thus, the larger the projection error is, the larger the correction is. The weight \( \alpha_{ij} \) can be calculated by the following formula:

\[
\alpha_{ij} = \frac{e_{ij}}{\sum_{k=2,3,4} e_{kk+1}} \quad e_{ij} = (e_i + e_j)/2
\]  

(16)

According to the graph theory, the total error of one loop can be allocated to its sub-loops[11]. Therefore, the modified rotation matrix between two landmarks \( i, j \) is \( R^*_{ij} = R_{ij} E^{e}_{ij} \). For the translation vector optimization, a decoupled point \( c_{ij} \) is used here. Ideally, the pose of \( c_{ij} \) should satisfy the following formula:

\[
R^*_{ij} c_{ij} + t_{ij} = R_{ij} c_{ij} + t_{ij}
\]  

(17)

\[
t_{ij} = (R_{ij} - R^*_{ij}) c_{ij} + t_{ij}
\]  

(18)

The optimized translation vector needs to minimize the error between \( \hat{t} \) and \( \hat{t}^* \), meanwhile, for any point \( v_k \) still satisfies the CPT method in loop, thus the optimization problem is given as follows:

\[
\begin{align*}
\min & \sum |\hat{t} - \hat{t}^* | \\
\text{s.t.} & \quad v_k = R^*_{kk} (R^*_{k+1} c_{k+1} + t_{k+1}^*) + \cdots + R^*_{k+1} (R^*_{k+2} v_{k+2} + t_{k+2}^*) + \cdots + t_{k+1}^* + t_{k+1}^*
\end{align*}
\]  

(19)

In the same way, the above optimization problem can be built through all the loops in \( G(t) \) and constitutes a constrained linear quadratic optimal problem [11]. The LM method is used here to get the optimal \( \hat{t}^* \). Finally, the optimized transformation matrix \( \hat{T} \) can be calculated and generated the optimized map \( G^*(t) \).
3.3. Synchronized Optimization

In order to solve the mismatch between \( G'(t_i) \) and \( M(t_i) \) after the global optimization, a synchronized optimization algorithm is proposed in this paper. The first detected landmark \( \mu \) in the key frame sequence is selected as the node, then the \( G'(t_i) \) is rotated and aligned with \( M(t_i) \).

\[
T_i^* = (T_i')^{-1}T_i^\prime
\]  
(20)

The mismatched landmarks in \( M(t_i) \) and \( G'(t_i) \) are manually added in the aligned graph \( G^*(t_i) \). Then the D* shortest path algorithm is used to find the path from \( \mu \) to the mismatched landmarks. Define \( L < \mu, ..., k > \) as the shortest path between the node and the mismatched landmark \( k \), then the transformation matrix of the mismatched landmark \( k \) in synchronized graph \( G(t_i) \) can be calculated as follows:

\[
\tilde{T}_k = \sigma^{-1}T_k^*, \quad \sigma = \prod_{i \in L} (T_i^*)^{-1}T_i
\]  
(21)

Above all, the mismatched landmark is optimized based on the accumulate corrections of the minimal re-projection path. The global optimization of all landmarks can be achieved through the continuous updating and incremental optimization of the map with the proposed synchronized optimization framework.

4. Experiment and Analysis

In order to verify the proposed SLAM algorithm, a \( 4 \times 4 \) m array contains \( 10 \times 10 \) ArUco landmarks is designed in this section. The array is arranged regularly and the ID is incremented. In the experiment, the precise map (Benchmark) of the array is obtained by manual calibration. The parameters are given as follows: the landmark size is 18 cm, array’s row spacing is 38 cm, the column spacing is 38.5 cm. The hardware system uses a PS3eye camera (60 frames, \( 320 \times 240 \) resolution) to capture real-time video, which is processed by an image processor (IntelRAtom 294i5-4300U 1.9 GHz CPU running Ubuntu14.04). The proposed algorithm is implemented by Opencv3.0, Eigen and g2o library.

4.1. Online Mapping Experiment

In order to verify the performance of the proposed algorithm, the experimental results are compared with the online mapping algorithm (Method1) proposed in [6] and the off-line mapping algorithm (Method2) proposed in [11]. The absolute center error (ACE) and the absolute trajectory error (ATE) is used to evaluate their performance. The experimental results are shown in figure 3. In figure 5, the red line is the real camera trajectory, and the blue line is the camera trajectory calculated by the each mapping algorithm. The proposed online mapping algorithm and the Method 2 algorithm have successfully built the map, while the map built by the Method1 algorithm is distorted.
Figure 5. Online Mapping Experiment.

Table 2 gives the detail comparison results of three algorithms. It can be seen that although the Method2 algorithm has the best mapping performance, but it is running off-line and spends 16.5 s. Although the Method1 algorithm implements online mapping, but it is easy to be disturbed by false detection. The proposed algorithm has a good balance between real-time, robust and accuracy. Although there are some accumulated errors, but it can be optimized by the proposed synchronized optimization algorithm latter.

Table 2. Performance Comparison of Mapping Algorithms.

| Algorithm | ACE (cm) | ATE (cm) | Number of Landmarks | Real Time |
|-----------|---------|---------|---------------------|-----------|
| Method1   | 36.0    | 45.0    | 62                  | Online    |
| Method2   | 2.5     | 4.1     | 100                 | Offline   |
| Ours      | 4.85    | 5.5     | 98                  | Online    |

4.2. Synchronization Optimization Experiment

In order to verify the synchronized optimization algorithm the following experiments are designed. The initial pose graph containing 42 landmarks and only 32 landmarks are recognized in the key frame sequence. The experimental results are shown in figure 6.

(a) Initial Pose Graph $G$  
(b) Optimized Graph $G'$
Figure 6(a) is the initial graph needed to be optimized, where the dotted line landmarks are Benchmark. The No.44, 45, 54, 55, 50, 60, 70, 94, 95 and 96 landmarks were not recognized in the key frame sequence. Figure 6(b) is the optimized graph, and the accuracy of the optimized graph is obviously improved. Figure 6(c) is the aligned graph which has added the mismatched landmarks. Figure 6(d) is the synchronized graph, the red landmarks are the optimized mismatched landmarks and the red dotted lines are the shortest path used for the synchronization. The comparison of the proposed optimization algorithm with Method2 mentioned before and Method3 proposed in [10] is shown in table 3.

Table 3. Contrast Experiment of Global Optimization Results.

| Algorithm | Time Cost(s) | ACE(cm) | Mark Number |
|-----------|--------------|---------|-------------|
| Method2   | 7.25         | 2.4     | 32          |
| Method3   | 3.03         | 9.6     | 32          |
| Ours      | 4.10         | 3.2     | 42          |

It can be seen from table 3 that the proposed algorithm is the only one that establishes the complete map, it not only ensures the accuracy of the map, but also guarantees less optimization time.

4.3. Simultaneous Localization and Mapping Experiment

In order to verify the overall performance of the proposed algorithm three simultaneous localization and mapping experiments are carried out. The experimental results are shown in the figure 7.
The red landmarks are the newly added landmarks with the proposed online mapping algorithm, the purple landmarks are the fixed landmarks during online correction and the black landmarks are the optimized and synchronized landmarks. Figure 7(2) shows the mapping result of randomly positioned and orientation aligned landmark array. Figure 7(3) shows the mapping result of the randomly positioned and orientated landmark array. As can be seen in Figure 7, the accurate maps of three different landmark arrays are established, there is almost no absence of landmarks, and the cumulative error of the map is optimized with the proposed synchronized optimization framework. In the experiment of Figure 7(3), the UWB system is used as the ground truth of the camera. The ATE of the proposed algorithm is lower than 4.9 cm, which meet the location requirement of low-cost robot system such as quadrotor.

5. Conclusion
In this paper, an online SLAM algorithm for ArUco landmark array is proposed. The algorithm can update and correct the global map online. The accumulated error is eliminated by minimizing the re-projection error of the landmark vertex in key frame sequence. The synchronized optimization process is used to optimize the mismatched landmarks. By comparing with the existing mapping system, the experiment results verify that the proposed algorithm can realize online mapping and avoid landmark loss or map distortion while ensuring the accuracy of mapping.

Acknowledgments
This work has been supported by the National Natural Science Foundation (NNSF) of China under Grant 91748211
Reference
[1] Garrido-Jurado S and Madrid-Cuevas F J 2014 Pattern Recognition Automatic generation and detection of highly reliable fiducial markers under occlusion  6 pp 2280-92
[2] Olson E 2011 IEEE International Conference on Robotics and Automation  3400-7
[3] Nitsche M, Krajnik T and Cíek P 2015 IROS Workshop on Open Source Aerial Robotics WhyCon: an efficient, marker-based localization system
[4] Bi Y and Duan H 2013 Optik-International Journal for Light and Electron Optics 124 3296-300
[5] Huang Y H, Wu T H and Lin M H Initial experience in stereotactic image-guided system for small animal with an automatic robot.
[6] Benavidez P, Lambert J and Jaimes A 2014 World Automation Congress Landing of an Ardrone 2.0 quadcopter on a mobile base using fuzzy logic  803-12
[7] Diego L D I, Mendona P R S and Hopper A 2002 Personal & Ubiquitous Computing TRIP: a low-cost vision-based location system for ubiquitous computing  6 pp 206-219
[8] Godil, A, Tsai, RE and Hong, H 2013 Nistir Ground truth systems for object recognition and tracking p 229
[9] Krajnik T, Nitsche M and Faigl J 2013 International Conference on Advanced Robotics External localization system for mobile robotics pp 1-6
[10] Klopschitz M and Schmalstieg 2007 IEEE Computer Society Automatic reconstruction of wide-area fiducial Marker Models pp 1-4
[11] Muoz-Salinas R, Marin-Jimenez M J and Yeguas-Bolivar E 2017 Pattern Recognition Mapping and localization from planar markers
[12] Maidi M, Ababsa F and Mallem M 2009 Signal Image & Video Processing Vision-inertial tracking system for robust fiducials registration in augmented reality 9 pp 83-90
[13] Bacik J, Durovsky F and Fedor P 2017 Intelligent Service Robotics Autonomous flying with quadrocopter using fuzzy control and ArUco markers pp 1-10
[14] Shaya K, Mavrinac A and Herrera J L A 2012 International Conference on Intelligent Robotics and Applications A self-localization system with global error reduction and online map-building capabilities pp 13-22
[15] Hartley, Richard and Trumpf 2013 International Journal of Computer Vision Rotation averaging chapter 103 pp 267-305