Improved Loop Closure Detection Algorithm for VSLAM with Spatial Coordinate Index

Weipeng Li, Guoliang Zhang, Jun Xu, Erliang Yao

Hi-Tech Institution of Xi’an, Xi’an, China

ABSTRACT: This paper presents an improved loop closure detection algorithm for VSLAM with spatial coordinate index in perceptual aliasing scenes. Firstly, we model the visual odometry accumulative error with Gaussian mixed distribution. Secondly, we eliminate the false loop closures by limiting the detection regions with accumulative error. Thirdly, the accumulative error is corrected by the appearance of loop closures for reducing the detection regions. Besides, we construct a dynamic octree for spatial coordinate index to accelerate the search of detection regions. Results demonstrate that, compared with IAB-MAP, FAB-MAP and RTAB-MAP, our algorithm has a better precision rate and a fine real-time property.

1 INTRODUCTION

Simultaneous localization and mapping (SLAM) is the key technology for autonomous robot to update the map and localize themselves in unknown environments. As an important part of SLAM, loop closure detection is significant for reducing the accumulative error of pose estimation and mapping uncertainty. With the development of sensor technology, RGB-D cameras were combined with loop closure detection and widely used in SLAM. Bag of Visual Word (BoVW), referenced from textual index [2] is widely used in visual loop closure detection algorithms. Based on BoVW, many modified algorithms appeared. Cummins described the correlation of words approximately with Chow-Liu tree and created Fast Appearance-Based Mapping (FAB-MAP) [4-5]. Angeli discussed the construction of incremental visual dictionary and presented Incremental Appearance-Based Mapping (IAB-MAP) [1]. Real-time Appearance-Based Loop Closure Detection (RTAB-MAP) [7-8] was an epoch. With a memory management mechanism, it keeps loop closure detection at a high speed in real-time. However, all of these algorithms have the same problem. They intend to provide false loop closures in perceptual aliasing scenes.

Our interest lies in improving the loop closure detection algorithm that can deal with perceptual aliasing scenes and keeping a fine real-time property in large-scale operation. First, we model the accumulative error of visual odometry. Then we use the Kalman Filter in updating the 3D coordinate of features. To limit the locations used in loop closures detection, detection regions are restricted by pose estimation uncertainty. And the coordinates of locations are indexed to speed up the selection of detection regions. As a result, the false loop closures are eliminated. Which improves the precision of loop closures detection. And the algorithm runs in a real-time standard as well. The experiment shows that our improved loop closure detection algorithm has a higher precision and real-time property compared with IAB-MAP, FAB-MAP in normal indoor environment, and a higher precision compared with RTAB-MAP in perceptual aliasing scenes.

2 UNCERTAINTY MODEL OF VISUAL ODOMETRY

Based on an uncertainty model of the Kinect, Dryanovski presented a frame-to-model registration in visual odometry [6]. They combined the features from sequential key frames into a single model to improve the consistency of features. But they left an incremental amount of feature matching. So we change the registration method from frame-to-model to frame-to-frame while persisting the relationship between adjacent key frame features. By this way, we reduce the calculation of registration. Then we model the visual odometry accumulative error based on feature coordinate uncertainty. Furthermore, we use the Kalman Filter to improve the mapping consistency when updating the camera pose and 3D coordinates of features.
2.1 3D uncertainty of sparse features

A 3D uncertainty of sparse features has been presented in [6]. Given a pixel with image coordinates \((u,v)\), \(d\) is the raw depth measurement of the RGB-D camera, the 3D point coordinate in camera frame is:

\[
x = \frac{z}{f_x} (u-c_x)
\]

\[
y = \frac{z}{f_y} (u-c_y)
\]

\[
z = d
\]

The parameters are mentioned as follow:

\((f_x, f_y)\): focal distance of the IR camera
\((c_x, c_y)\): IR image optical center

Let the pixel \((u,v)\) obeys the normal distribution with following approximate Gaussian kernel:

\[
G = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}
\]

Then we can obtain \(\sigma_u\) and \(\sigma_v\). Furthermore, the mean and variance of \(z\) are:

\[
\begin{align*}
\mu_z &= \sum_{i,j} G_{ij} \hat{z}_{i,j} \\
\sigma_z^2 &= \sum_{i,j} G_{ij} \left( \hat{z}_{i,j}^2 + \hat{\mu}_{i,j}^2 \right) - \mu_z^2
\end{align*}
\]

Combined with (1) and (2), the mean of 3D coordinate is:

\[
\mu = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} = \begin{bmatrix} \frac{\mu_x}{f_x} \\ \frac{\mu_y}{f_y} \\ \mu_z \end{bmatrix}
\]

Next, the uncertainty of the 3D coordinate can be estimated:

\[
\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 \end{bmatrix}
\]

where

\[
\sigma_x^2 = \frac{\sigma_z^2 (\mu_u-c_x)^2 + \sigma_u^2 (\mu_x^2 + \sigma_x^2)}{f_x^2}
\]

\[
\sigma_y^2 = \frac{\sigma_z^2 (\mu_v-c_y)^2 + \sigma_v^2 (\mu_y^2 + \sigma_y^2)}{f_y^2}
\]

\[
\sigma_x = \sigma_z \frac{\mu_u-c_x}{f_x}
\]

\[
\sigma_y = \sigma_z \frac{\mu_v-c_y}{f_y}
\]

\[
\sigma_{xy} = \sigma_z \frac{(\mu_u-c_x)(\mu_v-c_y)}{f_x f_y}
\]

2.2 Registration

The pose of camera can be indicated as a \(4\times4\) matrix:

\[
T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}
\]

Then a point \(P^{(c)}\) in camera coordinate system can be transferred into world coordinate system \(P^{(w)}\):

\[
P^{(w)} = RP^{(c)} + t
\]

Next we define the dataset of features in camera coordinate system as \(c_i=\{c_i, i=1,...,n\}\), where each \(c_i\) is described as:

\[
c_i = \left\{ \mu^{[c]}, \Sigma^{[c]}, x^{[c]} \right\}
\]

The \(\mu^{[c]}\) and \(\Sigma^{[c]}\) are defined as (6) and (8).

In this paper, the feature \(x\) is defined as a 64 dimension SURF [3]. Combined with KNN searching algorithm, we can match the features between two adjacent key frames. The features matched successfully called inner points while the others called outer points.

Dryanovski noticed that the Euclidean distance cannot reflect the uncertainty of spatial coordinate, but Mahalanobis distance was an effective way to calculate the similarity of two unknown sample set, reference [6] used Mahalanobis distance in ICP registration. But we think that the Mahalanobis distance is not suitable to be used as a space distance because it enlarges the influence of micro disturbance and it lacks the directly geometric meanings. So we use the distance:

\[
dist(P, Q) = \sqrt{\Delta^T \left( \frac{\Delta + \Sigma^{(w)}}{2} \right)^{-1} \Delta}
\]

where
\[ \Delta_{PQ} = \bar{\mu}_{P}^{[W]} - R \bar{\mu}_{Q}^{[C]} - t \]
\[ \Sigma_{Q}^{[W]} = R \Sigma_{Q}^{[C]} R^{T} \]  

(13)

\[ \mu_{P}^{[W]} \] and \( \mu_{Q}^{[C]} \) are the means of coordinates for a couple of features \( P \) and \( Q \) respectively in world coordinate system \( W \) and camera coordinate system \( C \).

The (12) combines the Euclidean distance and Mahalanobis distance, if the covariances decrease to 0, (12) changes into Euclidean distance.

2.3 Accumulative error estimation

Based on the 3D coordinate uncertainty of features, we can analyse the uncertainty of visual odometry. Considering into the uncertainty of visual odometry and coordinate transformation (11), the mean and covariance of \( P_{i}^{[W]} \) are:

\[ \begin{align*}
\bar{\mu}_{i}^{[W]} &= R \bar{\mu}_{i}^{[C]} + \mu_{i} \\
\Sigma_{i}^{[W]} &= R \Sigma_{i}^{[C]} R^{T} + \Sigma_{i}
\end{align*} \]  

(14)

Assuming that at time \( k \) the number of inner points is \( n \). From distance definition (12), the coordinate of camera can be estimated:

\[ t_{k} = \left[ \frac{1}{n} \sum_{i=1}^{n} (I + \Sigma_{p,k,i}^{[W]} + \Sigma_{p,k,i}^{[C]})^{-1} \right]^{-1} \]
\[ \sum_{i=1}^{n} \left( I + \Sigma_{p,k,i}^{[W]} + \Sigma_{p,k,i}^{[C]} \right)^{-1} \left( P_{i}^{[W]} - R_{i} P_{i}^{[C]} \right) \]  

(15)

But (15) has a huge amount of calculation. So we simplify (15) to an approximate formula when the covariances are not huge:

\[ t_{k} = \frac{1}{n} \sum_{i=1}^{n} \left( P_{i}^{[W]} - R_{i} P_{i}^{[C]} \right) \]  

(16)

Then we obtain the distribution of camera coordinate:

\[ t_{k} \sim N \left( \bar{\mu}_{k}, \Sigma_{k} \right) \]

\[ \begin{align*}
\bar{\mu}_{k} &= \frac{1}{n} \sum_{i=1}^{n} \left( \bar{\mu}_{i}^{[W]} - R_{i} \bar{\mu}_{i}^{[C]} \right) \\
\Sigma_{k} &= \left( 1 - \frac{1}{n} \right) \Sigma_{k-1} + \frac{1}{n} \sum_{i=1}^{n} \left( \Sigma_{p,k,i}^{[W]} + R_{i} \Sigma_{p,k,i}^{[C]} R_{i}^{T} \right) \\
&\quad + \sum_{i=1}^{n} \left( \bar{\mu}_{i}^{[W]} - \bar{\mu}_{i}^{[C]} \right) \left( \bar{\mu}_{i}^{[W]} - \bar{\mu}_{i}^{[C]} \right)^{T} \right)
\end{align*} \]  

(17)

Considering into the coordinate transformation of features (14), we can obtain the update of feature coordinates:

\[ \Sigma_{p,k-1}^{[W]} = R_{k-1} \Sigma_{p,k-1}^{[C]} R_{k-1}^{T} + \Sigma_{k-1} \]  

(18)

Combining (17) and (18), we notice that the accumulative error of visual odometry model increases with registration time.

2.4 Kalman Filter update

In order to increase the consistency of feature coordinates in world system and to reduce the accumulative error of visual odometry. We use the Kalman Filter to update feature coordinates.

First, set the predicted value of feature coordinate in world system at time \( k \) as follow:

\[ \begin{align*}
\bar{\mu}_{k} &= \bar{\mu}_{k-1} \\
\Sigma_{k} &= \Sigma_{k-1}
\end{align*} \]  

(19)

Marking the observed feature coordinate as \( \tilde{P}_{k}^{[W]} \), we can obtain the updating of feature coordinate:

\[ \begin{align*}
K_{k} &= \Sigma_{k-1} \left( \Sigma_{k-1} + \tilde{P}_{k}^{[W]} - \bar{\mu}_{k-1} \right)^{-1} \\
\bar{\mu}_{k} &= \bar{\mu}_{k-1} + K_{k} \left( \tilde{P}_{k}^{[W]} - \bar{\mu}_{k} \right) \\
\Sigma_{k} &= \left( I - K_{k} \right) \Sigma_{k-1}
\end{align*} \]  

(20)

After Kalman Filter updating, the feature coordinate uncertainty declines, that reduces the visual odometry accumulative error.

3 REGION CONSTRAINTS FOR LOOP CLOSURE DETECTION

To solve the perceptual aliasing problem during loop closure detection while keeping the algorithm in a real-time speed, we need a region constraint to distinguish that whether the similar appearances come from the same location. The visual odometry accumulative error is a great constraint for loop closure detection. But it also faces the increasing amount of comparison when the accumulative error enlarges. So we build a spatial coordinate index to speed up the retrieval of locations. Besides, the accumulative error is corrected when the loop closure appears.

3.1 Spatial coordinate index

We describe the locations \( X \) with two parts. The first one is the coordinate \( \hat{T}(X) \), the second one is the vector of visual words \( W(X) \) [9]:

\[ D(X) = \{ \hat{T}(X), W(X) \} \]  

(23)

where \( \hat{T}(X) \) is described as (10), and

\[ W(X) = \{ w_{1}, w_{2}, \cdots, w_{i} \} \]  

(21)
\[
 w_i = \begin{cases} 
 \frac{m}{m_i} \log \frac{M}{M_i}, & X \text{ includes } \lambda_i \\
 0, & X \text{ doesn't include } \lambda_i 
\end{cases}
\]  

(22)

\( M \) is the total number of locations, \( M_i \) is the number of locations including the feature \( \lambda_i \), \( m_i \) is the number of feature \( \lambda_i \) included in current location, \( m \) is the number of features in total.

Then we build a dynamic octree for the index of location’s coordinates. We want to find out which region the current location might belong to. The octree only needs to index the regions but not the specific locations. Figure 1 illustrates the building procedure of dynamic octree.

![Dynamic Octree Building Procedure](image)

Figure 1. Caption of the building procedure of dynamic octree. The dynamic octree is built by this procedure. Then it will be used to index the detection regions.

In this procedure, we judge whether the current location belongs to the current base unit. If not, whether it belongs to the current second rank unit, then the third rank, until to the top one. If it doesn’t belong to the current top unit, create a higher unit over the top one and the current location belongs to one of the lower units. Else if it doesn’t belong to current unit at specific rank, but belongs to the upper one, create other units at specific rank so it belongs to one of them.

3.2 Loop-Closure preliminary selection

In this paper, we use the accumulative error of visual odometry to design the constraint in loop-closure preliminary selection. The range of constraint is 3 times of standard deviation around the current location estimated. It is reasonable that there is a shift between the current location and the location passed when the loop closure occurs. So the range of constraint needs to enlarge \( \beta \), where \( \beta \) is an appropriate shifting of coordinate. Then we obtain the loop closure preliminary selection function:

\[
 f(t) = \left[(t - \mu_h)^\top \Sigma_h^{-1}(t - \mu_h)\right]^{1/2} \left| t - \mu_h \right| \frac{\sqrt{2}}{2} \eta 
\]

(24)

where \( \eta \) is the length of max margin in base unit.

The unit with the smallest fitness in each rank will be found out by comparing the fitness of their center coordinate \( t_R \). The base units whose fitness less than \( 3|t - \mu_h| \) are the regions which might contain loop-closure hypotheses. Finally, the preliminary selected locations are selected by (24) within these regions.

![Loop-Closure Preliminary Selection](image)

Figure 2. Caption of Loop-Closure preliminary selection. The regions contain loop-closure hypotheses are found based on octree.

3.3 Matching

The selection of loop closures hypothesis is based on the appearance comparison between preliminary selected locations and current location. The function of matching score is:

\[
 S(X, Y) = \frac{1}{\|W(X) - W(Y)\| + 1} 
\]

(25)

This function is easy to compute. The higher the score is, the more similar the frames are. Then we choose the historical location with highest score as the loop-closure hypothesis.

3.4 Accumulative error correction

After the appearance of loop-closures, we need to correct the accumulative error. If location \( Y \) is the
loop-closure of current location, the covariance of visual odometry can be corrected:

\[
\Sigma_{l_i} = \left(1 - \frac{1}{n}\right) \Sigma_{t_y} + \frac{1}{n} \sum_{i=1}^{n} \left( \Sigma_{v_i}^{(W)} + R_i \Sigma_{v_i}^{(C)} R_i^T \right) \\
+ \frac{1}{n^2} \sum_{i=1}^{n} \left[ (\mu_{v_i}^{(W)} - \mu_{v_i}^{(W)}) (\mu_{v_i}^{(W)} - \mu_{v_i}^{(W)})^T \right]
\]  

(26)

Notice that the location \( Y \) has been already past and the error is accumulative, we have \( \Sigma_{t_y} < \Sigma_{l_i} \).
Then, we can obtain \( \Sigma_{l_i} < \Sigma_{l_{i+1}} \). As a result, the detection regions are reduced, and the time of loop closure detection is limited.

4 RESULT

The performance of loop closure detection algorithm is evaluated in terms of precision–recall metrics and real-time property. Using a laptop with a 2.5GHz Inter Core i5 and a 4GB RAM, the experiments are implemented under the community RGB-D datasets and real scenes. The feature we use is SURF.
As a comparison, IAB-MAP, FAB-MAP and RTAB-MAP also runs in the same scene. The time threshold of RTAB-MAP is set to 700ms.

4.1 Community datasets

The RGB-D dataset we used is freiburg_room. It is a complex scene containing perceptual aliasing images. Figure 3 shows the result of mapping. The processing time for loop closure detection are shown in Figure 4, and the precision–recall curves are shown in Figure 5.

The processing time of IAB-MAP and FAB-MAP are increasing with the increasing of images. RTAB-MAP has a time threshold, so its processing time stables around 0.7s after the early increase. Because of the correction of the accumulative error, the processing time of our algorithm decreases after the loop closures are detected.
All the 4 kinds of loop closure detection algorithm have good precision–recall curves because of the small scale of dataset. But we can see that each curve has a little difference from others.

4.2 Real-time passageway

The passageway we experimented in contains lots of perceptual aliasing images. The photos of passageway are shown in Figure 6.
From Figure 7, we can see that the algorithm provides true loop closures so the map is corrected.
Figure 8 shows the processing time for loop closure detection in passageway. As expected, our improved algorithm has a fine real-time property.
Figure 9 shows the precision–recall curves for the loop-closure detection in passageway. It is clear that the improved algorithm has a better precision–recall performance.
5 DISCUSSION

The results presented in Section 4 suggest that our improved loop-closure detection algorithm has a good precision-recall performances and a high speed in perceptual aliasing scenes.

Overall, it is a simple way to speed up the loop-closure detection and to eliminate the false loop-closures by combining the region constraints and spatial coordinate index together. On the one hand, the region constraints limit the range of loop closure detection. On the other hand, loop closures correct the accumulative error and reduce the detection regions. This structure ensures the high precision of loop closure detection and the limitation of processing time. But it also has a danger of leaving out the true loop closures when the estimation of accumulative error is too smaller than the actual one, especially in long term running without the loop closures.

6 CONCLUSIONS

In this paper, we presents an improved loop closure detection algorithm for VSLAM with spatial coordinate index. It shows that our improved algorithm is able to find out true loop closures in perceptual aliasing scenes and keep a fine real-time property. Based on the spatial coordinate index and loop-closure preliminary selection, the detection range and processing time are limited. Especially, when a true loop closure is confirmed, the accumulative error is corrected followed by the reduction of detection regions. In future work, we plan to combine the loop-closure detection with the achievement from cognitive psychology and bioinformatics, to enhance the robustness and adaptability of loop-closure detection.

REFERENCES