Auton Robot (2017) 41:1107–1118
DOI 10.1007/s10514-016-9596-7

A bionic autonomous navigation system by using polarization navigation sensor and stereo camera

Zhiwen Xian[1] · Xiaofeng He[1] · Junxiang Lian[1] · Xiaoping Hu[1] · Lilian Zhang[1]

Received: 12 November 2015 / Accepted: 6 July 2016 / Published online: 15 July 2016

Abstract Inspired by the excellent navigational behavior of animals, this paper presents a bionic autonomous navigation (bio-navi) system by using a stereo camera and a polarization navigation sensor (POLNS). A topological graph consisting of nodes and edges forms the core of the system, which is similar to the simultaneous localization and mapping (SLAM). Like most of the SLAM approaches, in the bio-navi system, a node contains pose (position and attitude) and visual template information while an edge represents a constraint between two nodes. The difference is that the POLNS is introduced into the system. The yaw angle obtained from POLNS is attached to a node as a new property and also forms a new edge which restricts the direction of the related node. Besides, an absolute place constraint edge is added to the current node when recognizing a prior known place. To improve the accuracy and robustness of the place recognition, a POLNS-assist loop closure detection method is presented and to correct the pose error of constructed nodes, an improved graph optimization algorithm is derived in detail which is able to make full use of all the constraints. To demonstrate the performance of the system, an outdoor vehicle test has been done, and the results show its effectiveness and feasibility.

Keywords Bio-inspired · Autonomous navigation · Stereo camera · Polarization navigation sensor

✉ Xiaofeng He
xf.he.nudt@gmail.com

Zhiwen Xian
xianzw2012@gmail.com

[1] College of Mechatronics and Automation, National University of Defense Technology, ChangSha, China

1 Introduction

Navigation plays a vital role in our daily life. Global positioning system (GPS) and other global navigation satellites system (GNSS) technologies have been applied more and more widely; it seems that the unavailability of GNSS will make us unsettled or even grumpy when we step into a new and unknown environment. In fact, the GNSS does not and cannot work in every circumstance, since GNSS signal is deficient when facing obstruction and interference. Therefore, use of non-GNSS sources for autonomous navigation has increased significantly.

Spatial navigation is a core cognitive ability in animals. Many animals (including humans) can navigate over thousands of miles. Birds such as songbirds migrate between continents twice each year and return to the same breeding sites year after year with a voyage of up to 25,000 km (Cochran et al. 2004). Migratory songbirds can orient by compass information from polarized skylight pattern, the earths magnetic field, and the memorization of spatial cues en route. Insects such as bees and wasps are capable of learning landmarks in their environment, and of using these in navigation (Collett and Graham 2004). Etienne at al. (Etienne and Jeffery 2004) suggested that many animals have the ability of self-motion (also known as path integration). Besides, there are others navigational cues used by animals, such as olfaction (Gagliardo 2013) and echolocation (Geva-Sagiv et al. 2015). However, in this paper, we focus on the use of polarized skylight and vision.

The polarized skylight is the natural resource of the earth, which is caused by the scattering of sunlight in the earth's atmosphere and is partially linearly polarized (Lambrinos et al. 1997). It is suggested that skylight polarization pattern is used as a preferred compass that has no bias error and is used for calibrating all of their other compasses (Muheim et al. 2006). The main advantages of POLNS are that it has no drift when comparing with gyro compass and that it will not be interfered by magnetic sources when comparing with a magnetic compass. As for the vision, it can be used for landmarks identification, path integration, and 'cognitive maps' construction. By path integration, an animal keeps a continuously updated record of its current direction and distance from some reference point as it moves away from that place (Collett and Graham 2004). During path integration, an array of visually defined places have been attached to the path, what is often called a 'cognitive map'. When displaced to some recognizable location, the animal can retrieve the place's path integration coordinates and correct the animals currently estimated path integration position (O'keefe and Nadel 1978). Currently, how the 'cognitive map' is formed and corrected in a brain is largely unknown (Geva-Sagiv et al. 2015). Nevertheless, from a robot navigation point of view, this information is very helpful to design a bionic navigation system.

According to the mechanism of processing of polarized-light in certain animals (like desert ant), Lambrinos (Lambrinos et al. 2000) presented an original polarization compass model and method for extracting compass information. Chu (Chu et al. 2008) constructed a POLarization Navigation Sensor (POLNS). A novel angle computation and calibration algorithm of the POLNS is presented by Xian (Xian et al. 2014) and a camera based polarization measurement device is designed by Wang (Wang et al. 2014). However, most of the studies on the POLNS concentrate on the sensor itself, for example, the mechanical construction, angle computation, and sensor calibration. As for real navigation applications, they are rarely reported. In this paper, the POLNS will be used and fused with visual information for an actual navigation task.

In order to mimic the navigational strategy of animals using visual information, researchers have done lots of works (Srinivasan et al. 1999; Garratt and Chahl 2008; Baird et al. 2013). A bio-inspired snapshot based navigation scheme was applied to the guidance of unmanned aerial system (Denuelle and Srinivasan 2015). An area of particular relevance to visual navigation is the Visual Odometry (VO) (Kitt et al. 2010; Geiger et al. 2011) and visual simultaneous localization and mapping (SLAM) (Davison et al. 2007; Lategahn et al. 2011). VO is the process of estimating the egomotion of an agent using only the input of a single or multiple cameras attached to it (Scaramuzza and Fraundorfer 2011), which can be used for visual path integration. SLAM is currently regarded as a viable solution for large, unknown and unstructured environments (Boal et al. 2014). VO, loop closure detection and graph correction are the key technologies of SLAM and have developed a lot. However, the information used by most of the loop closure detection approach (Cummins and Newman 2008; Milford and Wyeth 2008) only depends on visual cues, and the graph correction method (Olson et al. 2006; Grisetti et al. 2010) is designed to correct the relative pose error. What's more, the sensor used by SLAM mainly are cameras and (or) other relative detection sensors like sonar, laser, and encoder. To the best of our knowledge, the use of POLNS has not been studied.

Inspired by the excellent navigational behavior of animals and the drawing upon current SLAM approaches, the paper presents a bionic navigation (bio-navi) system by using a POLNS and a stereo camera. As most of the SLAM approaches, the bio-navi system creates a topological graph when exploring an unknown environment. The graph is the core of the system, which contains an array of nodes and edges among the nodes. A node contains pose (position and attitude) and visual template information while an edge denotes a constraint between two linked nodes. Contrast to the SLAM, the POLNS which is able to provide yaw angle is introduced into the system. On the one hand, the yaw angle is added as a new property of each node and then used to assist visual template for loop closure detection. The use of both visual template and yaw angle in loop closure detection approach not only provides more information for matching but also reduces the searching space of visual template during visual matching, resulting a more efficient and reliable result comparing with the visual-only solution. On the other hand, the yaw angle of each node is used to form a new type of edge which constraints the direction of the related node. Besides, to mimic animals ability to use known landmarks to correct their position, an absolute place constraint edge is added the current node when recognizing a prior known place. In order to amend the pose error of each node accumulated by path integration, an improved graph optimization algorithm is presented in detail, which is able to make full use of the edges coming from path integration, loop closure detection, POLNS yaw angle constraints and absolute place constraints. To evaluate the performance of the proposed approach, a real outdoor vehicle test has been carried out, and the results validate its feasibility and accuracy.

The remainder of the paper is organized as follows: the system overview and related preliminaries are given in Sect. 2. Graph construction and optimization are detailed in Sect. 3 and 4. In Sect. 5, an outdoor vehicle test is carried out, and results are shown. Finally, main conclusions and future work are presented in Sect. 6.

2 System overview and preliminaries

2.1 system overview

In this section, we present an overview of the system architecture. As shown in Fig. 1, the bio-navi system is

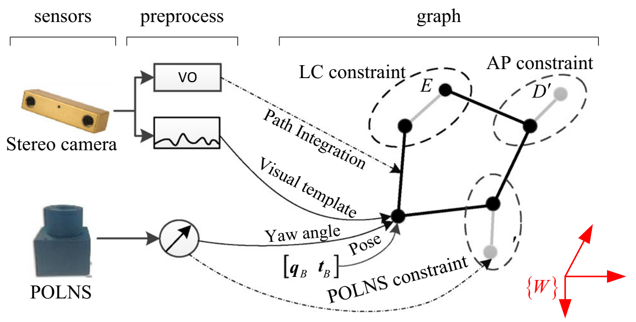

Fig. 1 The system overview of the bionic navigation system. In the graph, there are two kinds of nodes, the real node (such as A, B, C, D, E) and virtual node (such as C', D'). The goal of the system is to determine the pose of the real node with respect a global world frame W. The virtual nodes contain information about yaw angle (such as C') or positions information (such as D') with respect to the world frame W, and they are created only for the purpose of constraining the pose of their linked real node. The edges in the graph are also divided into two categories basing on the information contained in the edge, namely, relative constraint edge (such as $\overline{AB}$ and $\overline{AE}$) and absolute constraint edge(such as $\overline{CC'}$ and $\overline{DD'}$). And they show different contributions in the graph optimization process

divided by functionality into three modules, namely, sensors module, data preprocess module and topological graph module. The sensors used in this system are a stereo camera and a POLNS. Image sequences from the stereo camera are used for VO and visual template. The output of VO is relative rotation and translation between two consecutive time instants and is employed to do path integration. The visual template is the identity of a local view, which is used as the main reference of place identification and recognition. As for the POLNS, it is able to obtain the yaw angle of a carrier with respect to the earth true north. After data preprocessing, all the information is fed into the main module of the system, called topological graph.

The *graph* is the core of the bio-navi system, which acts as the cognitive map in animals hippocampus. It organizes the flow information from path integration, visual template and yaw angle into a set of related spatial *nodes*. An individual node (likes node B as shown in Fig. 1) contains the pose (rotation and translation), local visual template and yaw angle from the POLNS. When one or more of the three properties of current node change, a new node is created and is linked to the previous node. The link between two nodes is called *edge*, which encodes not only relative pose but also its covariance. An edge is a pose constraint in the graph, and most of the edges come from the path integration. Besides, there are three other types of edges, namely, loop closure (LC) constraint edge, POLNS constraint edge and absolute place (AP) constraint edge. When loop closure occurs, a LC constraint edge ($\overline{AE}$ as shown in Fig. 1) is added into the graph. A POLNS edge ($\overline{CC'}$) is used to restrict the direction of a node by adding a virtual node linked to the node. As for AP edge ($\overline{DD'}$), it is created only when the agent recognizes a prior known place. Basing on the Once the graph is constructed, a graph optimization algorithm will be done to revise the accumulated pose error of each node caused by the path integration.

Raw data coming from the stereo camera and POLNS are preprocessed before feeding into the graph. The POLNS is a bio-inspired sensor that is able to provide yaw angle, for details can refer to (Xian et al. 2014). The stereo camera is used for two purposes: path integration and local visual view extraction. To do stereo image sequences based path integration, an accurate and fast stereo VO method presented by (Kitt et al. 2010) is employed.

As for the local view extraction, only the left image of the camera is used at each step. The local visual template is represented by a scanline intensity profile formed from the image (the same as the profile used by Milford (Milford and Wyeth 2008)). The scanline intensity of a one-dimensional vector formed by summing the intensity values in each pixel column, and then normalizing the vector. The profile used by Milford has multiple purposes; here we only use it to compare the current image with previously stored images to perform place identification and recognition.

2.2 Preliminaries

In order to represent the pose and edge in the graph, a basic entity encoding attitude and position in the 3D space is defined as follows

$$\mathbf{x}_{ij} = \begin{bmatrix} \mathbf{q}_{ij} \\ \mathbf{t}_{ji} \end{bmatrix} \tag{1}$$

where $\mathbf{q}_{ij} = [q_0, q_1, q_2, q_3]^T$ is a unit quaternion (Diebel 2006) which denotes a rotation from frame j to frame i, and $\mathbf{t}_{ji}$ is the translation from frame i to frame j. The $\mathbf{x}_{ij}$ actually represents the transformation from frame j to frame i, which is equal to a homogeneous transformation, and they are interchangeable. With the definition of $\mathbf{x}_{ij}$ the related homogeneous transformation can be represented as follows

$$\mathbf{H}_{ij} = \begin{bmatrix} \mathbf{R}\left(\mathbf{q}_{ij}\right) & \mathbf{t}_{ji} \\ 0 & 1 \end{bmatrix} \tag{2}$$

where $\mathbf{R}(\mathbf{q})$ is the rotational matrix corresponding to the unit quaternion $\mathbf{q}$.

According to the definition of $\mathbf{x}_{ij}$, the inverse operator can be obtained as

$$\mathbf{x}_{ji} = \mathbf{x}_{ij}^{-1} = \begin{bmatrix} \mathbf{q}_{ij}^{-1} \\ -\mathbf{q}_{ij}^{-1}\left(\mathbf{t}_{ji}\right) \end{bmatrix} \tag{3}$$

where the $\mathbf{q}^{-1}$ is the inverse of the quaternion, and $\mathbf{q}\left(\mathbf{t}\right) = \mathbf{R}\left(\mathbf{q}\right)\mathbf{t}$ is rotation of the vector $\mathbf{t}$ according the the quaternion $\mathbf{q}$. Motion composition operator $\oplus$ is commonly used in this paper, which is defined as

$$\mathbf{x}_{wj} = \mathbf{x}_{wi} \oplus \mathbf{x}_{ij} = \begin{bmatrix} \mathbf{q}_{wi} \otimes \mathbf{q}_{ij} \\ \mathbf{t}_{iw} + \mathbf{q}_{wi}\left(\mathbf{t}_{ji}\right) \end{bmatrix} \tag{4}$$

where $\otimes$ is quaternion multiplication operator.

The pose of a node is the transformation with respect to an initially known frame. Assuming that the initial world frame is W, and then the pose of node i can be denoted as $\mathbf{x}_{wi}$. For simplicity, we only use $\mathbf{x}_i$ to represent the pose of node i. As for the edge, it represents the relative pose information between two nodes. An edge is fully characterized by its mean measurement $\mathbf{z}_{ij}$ and an information matrix (the inverse of the covariance matrix) $\mathbf{\Omega}_{ij}$ that accounts for its uncertainty.

3 Graph construction

The graph is a topological map composed of a set of individual nodes, nd, connected by edges. Each node nd_i corresponds to three properties, they are pose $\mathbf{x}_i$, local visual template vector V_i and yaw angle Y_i. A node can then be defined as a 3-tuple

$$nd_i = \{\mathbf{x}_i, V_i, Y_i\} \tag{5}$$

The first node is created at a known pose, and subsequent nodes build out from the first node over time.

3.1 Node creation

When one or more of the three properties is sufficiently different from a stored node, a new node is created. The current pose, local view and yaw angle are compared to that of the latest node through a score metric S.

$$S = \mu_x\left|\mathbf{x}_i - \mathbf{x}\right| + \mu_v\left|V_i - V\right| + \mu_Y\left|Y_i - Y\right| \tag{6}$$

where μ_x, μ_v, μ_Y are the weights of the three properties. When the score exceeds a threshold S_{th} , a new node is created and is linked to the previous node. The edge information which associates the relative transformation coming from the path integration cues.

Assuming that the latest node is i, the newly created node is j, and the measurement from VO between the two nodes is $\mathbf{z}_{ij}$ with its information matrix being $\mathbf{\Omega}_{ij}$. Then the pair$\left\langle \mathbf{z}_{jj}, \mathbf{\Omega}_{jj} \right\rangle$ is defined as an edge, and the initial pose guess of node j is estimated by

$$\breve{\mathbf{x}}_j = \breve{\mathbf{x}}_i \oplus \mathbf{z}_{ij} \tag{7}$$

If the POLNS is available when a node is creating, then a POLNS constraint edge $\left\langle \mathbf{z}_{jj}, \mathbf{\Omega}_{jj} \right\rangle$ is also linked to the newly created node. The POLNS edge has the same form as that of path integration edge, but different mean value and information matrix, they are computed as follows

$$\mathbf{z}_{jj} = \begin{bmatrix} \mathbf{q}\left(\mathbf{R}\left(0, 0, Y_j\right)\right) \\ [0, 0, 0]^T \end{bmatrix} \tag{8}$$

$$\mathbf{\Omega}_{jj} = \begin{bmatrix} \left(\mathbf{J}_q diag\left(inf, inf, \delta_Y^2\right)\mathbf{J}_q^T\right)^{-1} \\ \mathbf{0} \end{bmatrix} \tag{9}$$

where $\mathbf{R}\left(r, p, y\right)$ is the rotation matrix of three Euler angles, namely roll, pitch and yaw and $\mathbf{J}_q$ is the Jacobin matrix of a quaternion with respect to three Euler angles. For details about the computation of Euler angles to a rotation matrix and related derivatives can refer to (Diebel 2006). It is worth mentioning that even though the mean value $\mathbf{z}_{jj}$ contains all the pose information, only the yaw angle will constraint the current node pose since all the elements unrelated to yaw angle in the information matrix are zero.

3.2 POLNS assist loop closure detection and absolute place recognition

In order to detect the loop closure and absolute place recognition, both the local view and yaw angle are used. The loop closure or absolute place recognition occurs when current local view and orientation angle after a change in graph sufficiently matches a stored local view and related orientation. When this occurs, a new edge will be added to the existing graph. The matching method is similar to (6) but only use visual template vector and yaw angle. From implementation program point of view, a vector matching is more time-consuming than a scalar value comparison, so yaw angle is compared first and then do the visual template matching. By using the two-step matching implementation method, the matching speed will be faster and more accurate than that of visual-only approach.

Note that there are differences between loop closure and absolute place recognition. When an agent revisits a stored place, we call it loop closure when the pose of the stored place is local which obtained by path integration, and we call it absolute place recognition when the absolute position with respect to the global frame is prior known. Another difference is that they form different edges. Loop closure edge is connected to existing real node. By contrast, when the absolute place recognition takes place, a virtual node with the prior information is created and linked to the current node.

4 Graph optimization

4.1 Problem

The primary goal of the bio-navi system is to estimate poses of the whole trajectory. All the nodes and edges organized in the graph is for better pose solution. The poses of nodes are the parameters which need to be optimized while all the edges are seen as constraints on the parameters. The edges in our graph can be classified into two categories, one is the relative constraint (path integration constraint and LC constraint), and the other is the absolute constraint (POLNS constraints and AP constraint). Once the graph is constructed, the graph optimization problem is to find a configuration of the nodes that is maximally consistent with relative and absolute pose measurements. That involves solving a large error minimization problem.

Let $\mathbf{x} = \left(\mathbf{x}_1^T, \mathbf{x}_2^T, ..., \mathbf{x}_n^T\right)^T$ be the vector of parameters, where $\mathbf{x}_i$ is the pose of node i. Let $\left\langle \mathbf{z}_{ij}, \boldsymbol{\Omega}_{ij} \right\rangle$ and $\left\langle \mathbf{z}_{jj}, \boldsymbol{\Omega}_{jj} \right\rangle$ be respectively the relative constraint edge and absolute constraint edge respectively. Then the optimization problem can be solved by finding the minimum of the function in this form

$$
\begin{aligned}
F(\mathbf{x}) = & \sum_{<i,j>\in C_{ij}} \underbrace{\mathbf{e}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right)^T \boldsymbol{\Omega}_{ij} \mathbf{e}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right)}_{\mathbf{F}_{ij}} \\
& + \sum_{k\in C_{kk}} \underbrace{\mathbf{e}(\mathbf{x}_k, \mathbf{z}_{kk})^T \boldsymbol{\Omega}_{kk} \mathbf{e}\left(\mathbf{x}_k, \mathbf{z}_{kk}\right)}_{\mathbf{F}_{kk}} \qquad (10)
\end{aligned}
$$

$$
\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\, F(\mathbf{x}) \qquad (11)
$$

where C_{ij} and C_{kk} are the index collections of relative and absolute constraint edges respectively; $\mathbf{e}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right)$ is a relative measurement error function that measures how well the pose $\mathbf{x}_i$ and $\mathbf{x}_j$ satisfy the constraint $\mathbf{z}_{ij}$, and similarly $\mathbf{e}\left(\mathbf{x}_k, \mathbf{z}_{kk}\right)$ is an absolute measurement error function.

For simplicity of notation, in the rest of this paper, we will encode the measurement in the indices of the error function, and the two error function are defined as follows

$$
\mathbf{e}_{ij}(\mathbf{x}) = \mathbf{e}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}\right) = \left(\mathbf{z}_{ij}^{-1} \oplus \left(\mathbf{x}_i^{-1} \oplus \mathbf{x}_j\right)\right)_{[2:7]} \qquad (12)
$$

$$
\mathbf{e}_{kk}(\mathbf{x}) = \mathbf{e}\left(\mathbf{x}_k, \mathbf{z}_{kk}\right) = \left(\mathbf{z}_{kk}^{-1} \oplus \mathbf{x}_k\right)_{[2:7]} \qquad (13)
$$

where the operator $(\bullet)_{[2:7]}$ selects the last 6 elements of its vector argument.

4.2 Solution

An initial guess $\breve{\mathbf{x}}$ of the pose can be computed from the path integration, the numerical solution of (11) can be obtained by using Least Squares optimization. Here we use an alternative approach to solving the problem, which is called Least Square on a Manifold in (Grisetti et al. 2010). The idea is to optimize the increments $\Delta\mathbf{x}$ and then compensate the initial guess instead of optimizing $\mathbf{x}$ directly. Contrast to the approach in (Grisetti et al. 2010), the optimization has been improved in order to deal with both relative and absolute constraints.

Let $\Delta\mathbf{x}_i$ be the perturbation around the current variable $\breve{\mathbf{x}}_i$, which is defined as

$$
\Delta\mathbf{x}_i = \left[\Delta\boldsymbol{\theta}_i^T, \Delta\mathbf{t}_i^T\right]^T \qquad (14)
$$

Note that $\Delta\mathbf{x}_i$ is a 6D vector, in which $\Delta\boldsymbol{\theta}$ is the vector part of the unit quaternion representing the 3D rotation and $\Delta\mathbf{t}$ denotes the translation. For the quaternion increment, we have

$$
\Delta\mathbf{q} = \left[\sqrt{1 - \|\Delta\boldsymbol{\theta}\|^2}, \Delta\boldsymbol{\theta}^T\right]^T \qquad (15)
$$

A new value of a variable $\mathbf{x}_i^*$ after the optimization can be obtained by

$$
\mathbf{x}_i^* = \breve{\mathbf{x}}_i \oplus \Delta\mathbf{x}_i = \begin{bmatrix} \breve{\mathbf{q}}_i \\ \breve{\mathbf{t}}_i \end{bmatrix} \oplus \begin{bmatrix} \Delta\mathbf{q}_i \\ \Delta\mathbf{t}_i \end{bmatrix} \qquad (16)
$$

Now the previous error (12) and (13), are the function of $\Delta\mathbf{x}$. Let's rewrite the error equations and expand them by their first order Taylor series approximation.

$$
\begin{aligned}
\mathbf{e}_{ij}\left(\breve{\mathbf{x}} + \Delta\mathbf{x}\right) &= \mathbf{e}_{ij}\left(\breve{\mathbf{x}}_i + \Delta\mathbf{x}_i, \breve{\mathbf{x}}_j + \Delta\mathbf{x}_j\right) \\
&\approx \mathbf{e}_{ij} + \mathbf{J}_{ij}\Delta\mathbf{x} \qquad (17)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{e}_{kk}\left(\breve{\mathbf{x}} + \Delta\mathbf{x}\right) &= \mathbf{e}_{kk}\left(\breve{\mathbf{x}}_k + \Delta\mathbf{x}_k\right) \\
&\approx \mathbf{e}_{kk} + \mathbf{J}_{kk}\Delta\mathbf{x} \qquad (18)
\end{aligned}
$$

Here, $\mathbf{J}_{ij}$ and $\mathbf{J}_{kk}$ are the Jacobian matrix of $\mathbf{e}_{ij}(\mathbf{x})$ and $\mathbf{e}_{kk}(\mathbf{x})$ with respect to $\Delta\mathbf{x}$ computed in $\Delta\mathbf{x} = \mathbf{0}$

$$
\mathbf{J}_{ij} = \left.\frac{\partial \mathbf{e}_{ij}\left(\breve{\mathbf{x}} \oplus \Delta\mathbf{x}\right)}{\partial \Delta\mathbf{x}}\right|_{\Delta\mathbf{x}=\mathbf{0}} \qquad (19)
$$

$$
\mathbf{J}_{kk} = \left.\frac{\partial \mathbf{e}_{kk}\left(\breve{\mathbf{x}} \oplus \Delta\mathbf{x}\right)}{\partial \Delta\mathbf{x}}\right|_{\Delta\mathbf{x}=\mathbf{0}} \qquad (20)
$$

Substituting (17) and (18) in (10), we obtain

$$\begin{aligned} F_{ij}\left(\breve{\mathbf{x}}+\Delta\mathbf{x}\right) &= \mathbf{e}_{ij}\left(\breve{\mathbf{x}}+\Delta\mathbf{x}\right)^T \boldsymbol{\Omega}_{ij}\mathbf{e}_{ij}\left(\breve{\mathbf{x}}+\Delta\mathbf{x}\right) \\ &\approx \left(\mathbf{e}_{ij}+\mathbf{J}_{ij}\Delta\mathbf{x}\right)^T \boldsymbol{\Omega}_{ij}\left(\mathbf{e}_{ij}+\mathbf{J}_{ij}\Delta\mathbf{x}\right) \\ &= \underbrace{\mathbf{e}_{ij}^T\boldsymbol{\Omega}_{ij}\mathbf{e}_{ij}}_{c_{ij}} + 2\underbrace{\mathbf{e}_{ij}^T\boldsymbol{\Omega}_{ij}\mathbf{J}_{ij}}_{\mathbf{b}_{ij}^T}\Delta\mathbf{x} \\ &\quad + \Delta\mathbf{x}^T\underbrace{\mathbf{J}_{ij}^T\boldsymbol{\Omega}_{ij}\mathbf{J}_{ij}}_{\mathbf{H}_{ij}}\Delta\mathbf{x} \\ &= c_{ij} + 2\mathbf{b}_{ij}^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_{ij}\Delta\mathbf{x} \end{aligned} \tag{21}$$

Similarly,

$$F_{kk}\left(\breve{\mathbf{x}}+\Delta\mathbf{x}\right) \approx c_{kk} + 2\mathbf{b}_{kk}^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_{kk}\Delta\mathbf{x} \tag{22}$$

Then the cost function can be rewritten as

$$\begin{aligned} &F\left(\breve{\mathbf{x}}+\Delta\mathbf{x}\right) \approx \\ &\quad \sum_{<i,j>\in C_{ij}} \left(c_{ij} + 2\mathbf{b}_{ij}\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_{ij}\Delta\mathbf{x}\right) \\ &\quad + \sum_{k\in C_k} \left(c_{kk} + 2\mathbf{b}_{kk}\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_{kk}\Delta\mathbf{x}\right) \\ &\quad = c + 2\mathbf{b}^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}\Delta\mathbf{x} \end{aligned} \tag{23}$$

where $c = \sum c_{ij} + \sum c_{kk}$, $\mathbf{b} = \sum \mathbf{b}_{ij} + \sum \mathbf{b}_{kk}$, and $\mathbf{H} = \sum \mathbf{H}_{ij} + \sum \mathbf{H}_{kk}$. It can be minimized in $\Delta\mathbf{x}$ by solving the linear system

$$\mathbf{H}\Delta\mathbf{x}^* = -\mathbf{b} \tag{24}$$

where $\mathbf{H}$ is the information matrix of the total system.

To solve (24), the information matrix $\mathbf{H}$ and the vector $\mathbf{b}$ should be constructed first. Every constraint will contribute to the system with an added term. The structure of this addend depends on the Jacobian of the error function. Since the error function of one constraint only depends on the values of one or two nodes, the Jacobian in (19) and (20) have the following form:

$$\mathbf{J}_{ij} = \left(\cdots \quad \underbrace{\mathbf{A}_{ij}}_{node\, i} \quad \cdots \quad \underbrace{\mathbf{B}_{ij}}_{node\, j} \quad \cdots\right) \tag{25}$$

$$\mathbf{J}_{kk} = \left(\cdots \quad \underbrace{\mathbf{C}_{kk}}_{node\, k} \quad \cdots\right) \tag{26}$$

where $\mathbf{A}_{ij}$ and $\mathbf{B}_{ij}$ are respectively the derivatives of the error $\mathbf{e}_{ij}\left(\breve{\mathbf{x}}+\Delta\mathbf{x}\right)$ with respect to $\Delta\mathbf{x}_i$ and $\Delta\mathbf{x}_j$, and $\mathbf{C}_{kk}$ is the derivatives of the error $\mathbf{e}_{kk}\left(\breve{\mathbf{x}}+\Delta\mathbf{x}\right)$ with respect to $\Delta\mathbf{x}_k$. The form of $\mathbf{A}_{ij}$ is a little complex while the form of $\mathbf{B}_{ij}$ and $\mathbf{C}_{kk}$ are very simple, that are

$$\mathbf{A}_{ij} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \tag{27}$$

where

$$\begin{aligned} \mathbf{A}_{11} &= A_{22} = -\mathbf{R}\left(\mathbf{q}_j^{-1}\right)\mathbf{R}\left(\mathbf{q}_i\right) \\ \mathbf{A}_{21} &= 2\mathbf{R}\left(\mathbf{q}_j^{-1}\right)\lfloor \mathbf{t}_j \times \rfloor \mathbf{R}\left(\mathbf{q}_i\right) - 2\mathbf{R}\left(\mathbf{q}_j^{-1}\right)\mathbf{R} \\ &\qquad \left(\mathbf{q}_i\right)\left\lfloor \mathbf{q}_i^{-1}\left(\mathbf{t}_i\right)\times \right\rfloor \\ \mathbf{B}_{ij} &= \mathbf{C}_{ij} = \mathbf{I}_6 \end{aligned} \tag{28}$$

where $\mathbf{I}_n$ is the identity matrix with the dimension of n.

From (21) and (22), the structure for the block of the matrix $\mathbf{H}$ and vector $\mathbf{b}$ can be obtained as follow

$$\mathbf{H}_{ij} = \begin{pmatrix} \ddots & & & & \\ & \mathbf{A}_{ij}^T\boldsymbol{\Omega}_{ij}\mathbf{A}_{ij} & \cdots & \mathbf{A}_{ij}^T\boldsymbol{\Omega}_{ij} & \\ & \vdots & \ddots & \vdots & \\ & \boldsymbol{\Omega}_{ij}\mathbf{A}_{ij} & \cdots & \boldsymbol{\Omega}_{ij} & \\ & & & & \ddots \end{pmatrix} \tag{29}$$

$$\mathbf{b}_{ij} = \begin{pmatrix} \vdots \\ \mathbf{A}_{ij}^T\boldsymbol{\Omega}_{ij}\mathbf{e}_{ij} \\ \vdots \\ \boldsymbol{\Omega}_{ij}\mathbf{e}_{ij} \\ \vdots \end{pmatrix} \tag{30}$$

$$\mathbf{H}_{kk} = \begin{pmatrix} \ddots & & \\ & \boldsymbol{\Omega}_{kk} & \\ & & \ddots \end{pmatrix} \tag{31}$$

$$\mathbf{b}_k = \begin{pmatrix} \vdots \\ \boldsymbol{\Omega}_{kk}e_{kk} \\ \vdots \end{pmatrix} \tag{32}$$

For simplicity of notation, the zeros blocks are omitted.

4.3 Pseudo code

For a detailed description of the graph optimization algorithm flow, the pseudo-code is given as in Algorithm 1.

Algorithm 1 Graph optimization algorithm

```
INPUT: initial pose guess $\breve{\mathbf{x}}$; edges $C_{ij} = \{\langle \mathbf{z}_{ij}(\bullet), \boldsymbol{\Omega}_{ij} \rangle\}$ and $C_{kk} = \{\langle \mathbf{z}_{kk}(\bullet), \boldsymbol{\Omega}_{kk} \rangle\}$
OUTPUT: optimized $\mathbf{x}^*$
while not converged do
    $\mathbf{H} = \mathbf{0}, \mathbf{b} = \mathbf{0}$                                  ▹ init H, b
    for All $\langle \mathbf{z}_{ij}(\bullet), \boldsymbol{\Omega}_{ij} \rangle \in C_{ij}$ do          ▹ relative edges
        $\mathbf{e}_{ij} = \mathbf{e}\left(\breve{\mathbf{x}}_i, \breve{\mathbf{x}}_j, \mathbf{z}_{ij}\right)$                  ▹ compute error
        $\mathbf{A}_{ij} = \left.\frac{\partial \mathbf{e}_{ij}\left(\breve{\mathbf{x}} \oplus \Delta\mathbf{x}\right)}{\partial \Delta \mathbf{x}_i}\right|_{\Delta\mathbf{x}_i = \mathbf{0}}$     ▹ compute Jacobian
                                                          ▹ construct H, b
        $\mathbf{H}_{[ii]} += \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij}, \mathbf{H}_{[ij]} += \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij}$
        $\mathbf{H}_{[ji]} += \boldsymbol{\Omega}_{ij} \mathbf{A}_{ij}, \mathbf{H}_{[jj]} += \boldsymbol{\Omega}_{ij}$
        $\mathbf{b}_{[i]} += \mathbf{A}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}, \mathbf{b}_{[j]} += \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}$
    end for
    for All $\langle \mathbf{z}_{kk}(\bullet), \boldsymbol{\Omega}_{kk} \rangle \in C_{kk}$ do          ▹ absolute edges
        $\mathbf{e}_{kk} = \mathbf{e}\left(\breve{\mathbf{x}}_k, \mathbf{z}_{kk}\right)$
        $\mathbf{H}_{[kk]} += \boldsymbol{\Omega}_{kk}$
        $\mathbf{b}_{[kk]} += \boldsymbol{\Omega}_{kk} \mathbf{e}_{kk}$
    end for
    $\Delta\breve{\mathbf{x}} = \mathrm{lscov}(\mathbf{H}, -\mathbf{b})$                   ▹ solve the linear system
    $\breve{\mathbf{x}} = \breve{\mathbf{x}} \oplus \Delta\breve{\mathbf{x}}$                        ▹ compensate the error
end while
$\mathbf{x}^* = \breve{\mathbf{x}}$
return $\mathbf{x}^*$
```

5 Experiment and results

5.1 Experiment setup

To evaluate the performance of the proposed bio-navi system, a real test has been done. The sensors used in the test contain a PointGrey Bumblebee2 stereo camera, a POLNS, a GPS receiver and a navigational grade Inertial Navigation System (INS). As shown in Fig. 2, the stereo, POLNS, and GPS were mounted on the top of a land vehicle while the INS was in the vehicle. The stereo and POLNS were used for our bio-navi system while the INS and GPS system were used as ground truth, as they can provide open sky position/attitude precision as high as 2 m/0.01 deg. The camera's field of view is 70 deg with a focal length of 3.8 mm, and the resolution of the image is 512×384 pixels. Stereo images are sampled at a rate of 12 Hz while the POLNS provides measurements at 10 Hz. The experiment was carried out in a typical urban environment in Changsha, China. The test vehicle moved around an area for two loops, with the total length being of about 2.5 km.

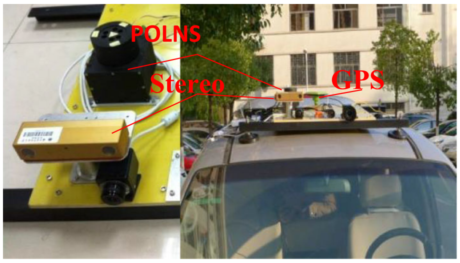

Fig. 2 The bio-navi system setup

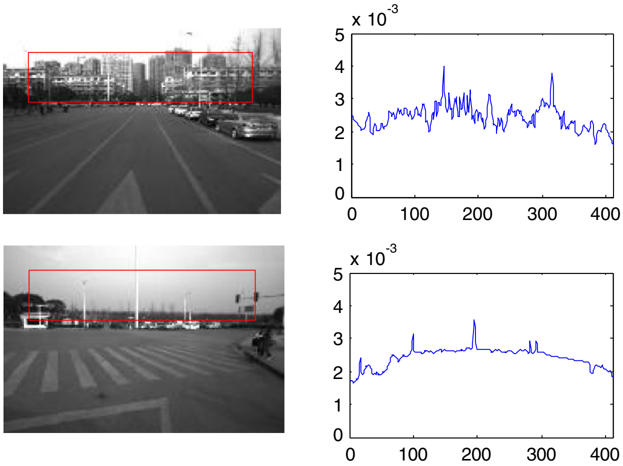

Fig. 3 Sample images (*left column*) and related visual templates (*right column*), the visual template comes from the sub-image region (*red rectangle*) of the image (Color figure online)

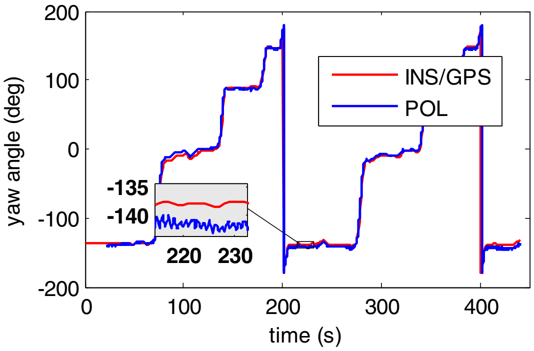

Fig. 4 POLNS yaw angle compared with that of INS/GPS

5.2 Results

5.2.1 *Visual template, POLNS yaw angle and loop closure detection*

Stereo image pairs were used as input of VO method for path integration while the sub-image of the left image was used to generate the local visual template. Two samples image and related visual templates are shown in Fig. 3. Clearly to see that the upper image regions are white sky and the lower ground region contains more moving objects (vehicles, pedestrians, and so on). These regions are not distinctive and so not chosen for the visual template. By contrast, the red rectangle region is more likely to contains distinct visual information, so this image area is used for scene recognition.

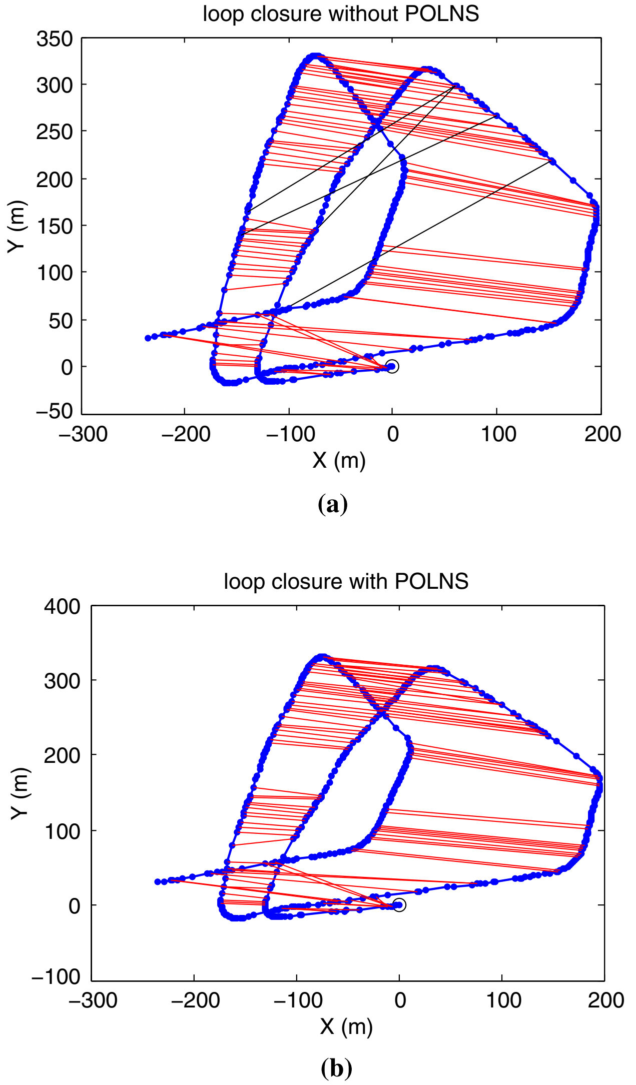

Fig. 5 Loop closure detection results without (**a**) and with (**b**) POLNS assistant. The circles in both **a** and **b** show the starting points. Positions (from visual odometry) at which a node was created are marked with a *blue dot*. The *blue lines* between two consecutive nodes are the relative constraint edge from visual odometry while the *red* (and *black*) *lines* between two nodes are the relative constraint edges from loop closure detections. Obviously, there are several false loop closure detections in (**a**) which are marked with *black lines* (Color figure online)

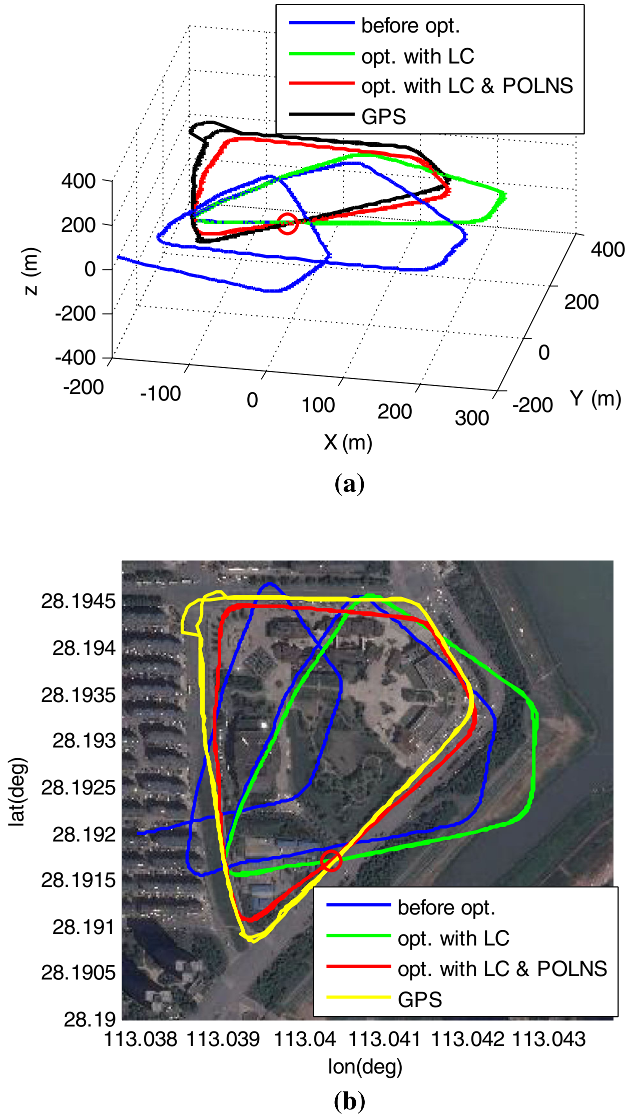

Fig. 6 Trajectories (**a**: 3d, **b**: 2d overlaid on the Google map) comparison of different solutions. Noting that the *red circle* denotes the starting points and that the jumps in the *top left* parts of the GPS trajectory in both 2d and 3d images were caused by GPS signal outage (Color figure online)

The direct outputs of the POLNS are polarization direction angle and degree. The accuracy of the polarization direction angle is able to reach about 0.2 deg. However, that is not the yaw angle's accuracy. In order to obtain the yaw angle of a carrier with respect to the earth truth north, we also need an atmosphere polarization model, an approximate location information and an accurate local time which is used to compute the celestial position of the sun. In practice, the major error comes from the difference between the theoretical and real models of the polarized skylight pattern. Currently, the best and most used model is Rayleigh sky model. However, it fits well with the real polarized sky model when the sky is clear and blue. So a cloudy sky will bring large yaw angle error, and it did happen in our test. As shown in Fig. 4, yaw angle estimated by POLNS is compared with the result of INS/GPS system. The POLNS yaw angle is noisy, and its error reaches about 5 deg sometimes. Even though the accuracy of POLNS yaw angle is poor, its advantage is obvious. As shown in Fig. 4, the error does not accumulate over time since the current value only depends on the currently input polarized skylight. That may also be the reason why so many animals use the polarized skylight as navigational cues.

Reliable loop closure is achieved by POLNS assist local view matching method. The results of loop closure detec-

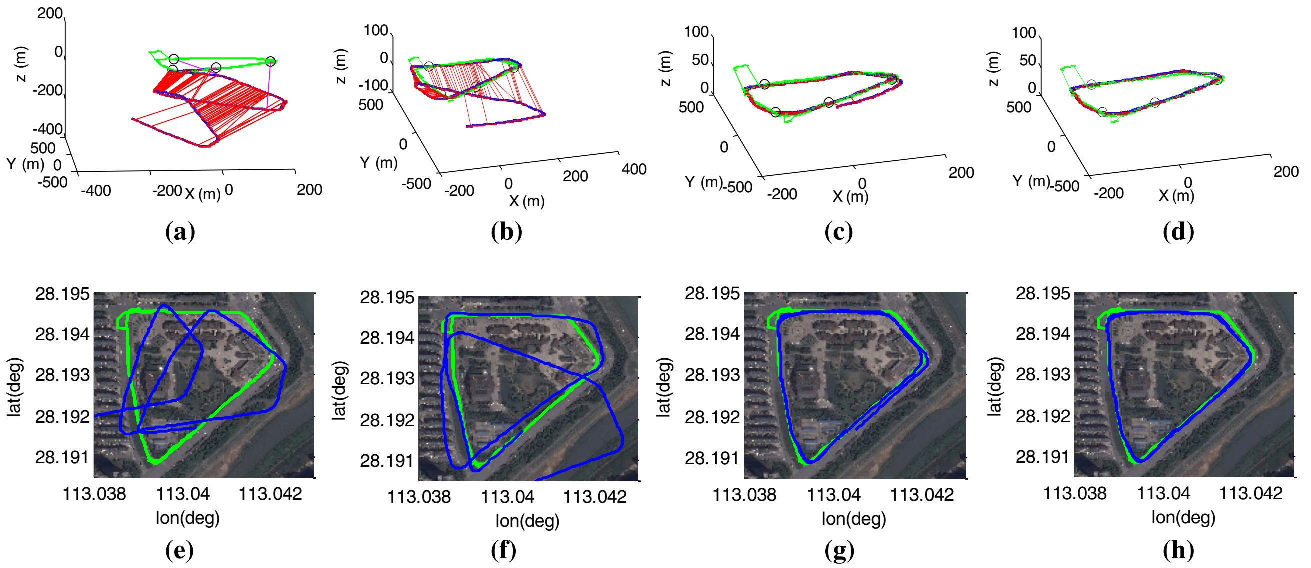

Fig. 7 The convergence process of graph optimization by using LC&POLNS&AP constraints. **a**–**d** show the 3d graph structure during the iterative process while **e**–**h** show the 2d trajectories overlaid on the Google map. The *green* and *blue lines* in all the figures denote the ground truth and estimated trajectory respectively. Again, the jumps of GPS solution in the figures were caused by GPS signal outage. Four *black circles* in (**a**)–(**d**) are priorly known places, and the *magenta lines* linked to the *circles* are the AP constraints. *Red lines* show in (**a**)–(**d**) denotes LC constraints (Color figure online)

tion with and without using POLNS are compared in Fig. 5. Apparently to see that there are several obvious outlier loop closure constraints (black lines in Fig. 5a) in the vision-only approach as shown in Fig. 5a. By contrast, the POLNS-assist approach has a better result. The comparison shows that the more information used in the loop closure detection, the better result will be achieved. Besides, the POLNS-assist approach reduced the search space of visual template at each time and then mostly improved the running speed.

5.2.2 Graph optimization with loop closure constraints and POLNS constraints

The graph constructed in this test contains 567 individual nodes and 1216 constraints in which there are 566 constraints created by path integration, 83 LC constraints, 566 POLNS constraints and one artificially added AP constraints to the first node for the purpose of keeping the first node fixed. The raw graph before optimization is shown in Fig. 5b.

To test the graph optimization performance, the initial attitude of the first node is not aligned with the ground truth but is added a yaw angle error of 30 deg. Two graph optimization approaches are done: one is to do the graph optimization with LC constraints (denoted as LC) and the other one using both LC and POLNS constraints (denoted as LC&POLNS). The trajectories estimated before optimization and after optimization are compared in Fig. 6. As can be seen in both the 3d (Fig. 6a) and 2d (Fig. 6b), the two solutions successfully recover the true shape of the track. The difference is that the LC&POLNS obtained not only the shape of the track but also a better orientation estimation that is close to the real direction of the ground truth as shown in Fig. 6b. The results indicate the feasibility of the graph optimization method and also show that the LC constraints are useful for the recovery of the track shape while the POLNS constraints contribute to the direction recover.

5.2.3 Graph optimization with absolute place constraints

In order to test the consequence of using absolute place constraints during graph optimization, four absolute place constraints (including the one connected to the first node) were artificially added. The iterative process is depicted in Fig. 7. As shown in both Fig. 7a, e, the initial graph has a large error in both position and direction. After one iteration, the estimated trajectory began to approach to the real value, as shown in Fig. 7b, f, especially the position and direction of the first loop. The reason why the first loop has faster convergence speed is that the AP constraints are connected to the node in the first loop, and they have higher weights during the optimization. The estimated trajectory is very close to the real value after three iterations, leaving only small position error, as shown in Fig. 7c, f. Finally, the estimated track converged at the 5th iteration, and fits well with the ground truth, as shown in Fig. 7d, h.

To compare the position and yaw angle estimated by using different constraints combination. The 3d position errors and

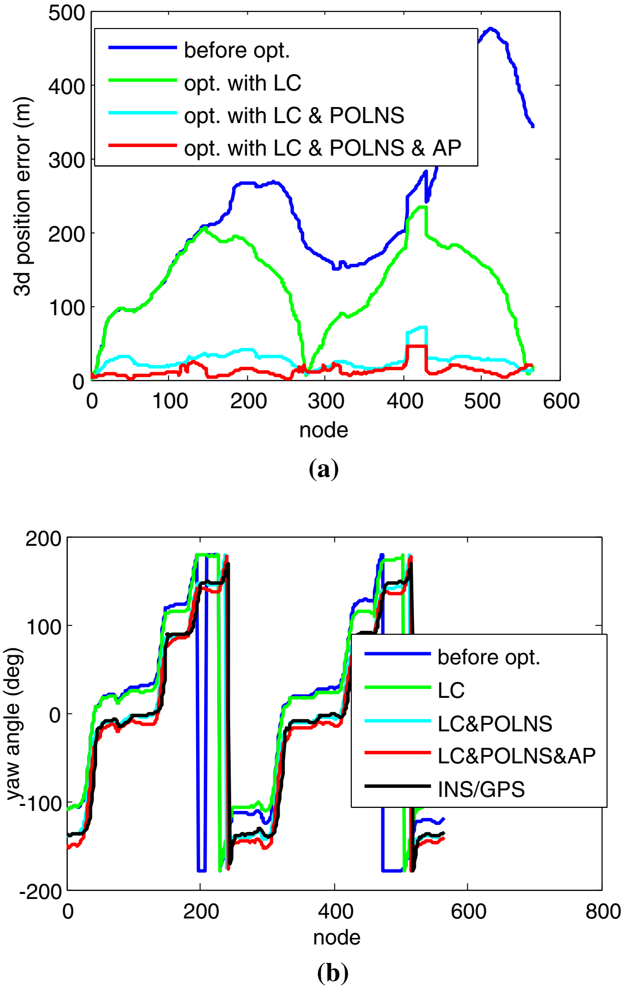

Fig. 8 3d position error (**a**) and yaw angle error (**b**) comparison of different approaches. Note that the large errors occurred in the nodes (about 400–430) is the results of incorrect GPS position in these areas

yaw angle errors are compared in Fig. 8. The track before optimization has large position error (as shown in Fig. 8a) and yaw angle error (as shown in Fig. 8b). By using the LC constraint, the position accuracy does not improve a lot, since the LC optimization cannot able to improve the direction of the trajectory. However, the yaw angle accuracy has largely improved by adding the POLNS constraints into the graph, and, therefore the position error is significantly reduced. The position accuracy gets a further improvement when adding the AP constraints into the graph, with the 3d position RMS error being less than 12 m. It is worthwhile to note that either the position error or yaw angle error do not grow over time.

6 Conclusions and future work

In this paper, we presented an alternative navigation strategy approach, called bio- navi system which mimics the navigation mechanism of animals. The core of the bio-navi system is a topological graph which consists of sequences of nodes and edges. The node contains the information of the position, attitude, POLNS yaw angle and local visual template. The edge which denotes the constraints in the graph comes from four sources, namely, path integration, LC, POLNS, and AP constraints. Stereo images are used for path integration and local visual template while the POLNS are used as an attribute of a node and also used for constraining the direction of the node and assisting loop closure detection. The details of how the bio-navi system works are depicted in this paper.

The proposed algorithm has been validated by using real outdoor data. The proposed POLNS-assist vision closure detection method was compared with the visual-only method, and the result shows the POLNS-assist method is more reliable that visual-only approach. The results of graph optimization with different constraints are also compared. The comparison shows that all of the different types of constraints are beneficial to recover the real trajectory, in which the LC constraints are useful for recover the topological structure, the POLNS constraints are helpful to regain the orientation of the graph and the AP constraints contribute to restoring the real global position of the each node in the graph. For the typical vehicle path of about 2.5 km, the 3d position RMS error is less than 12 m and does not accumulate over time. It is concluded that the proposed bio-navi strategy is effectiveness and feasibility and is able to be used in an unknown environment.

As for future work, several aspects of the bio-navi system should be further improved. First, more sensors will be introduced into the system, like inertial and magnetic sensors. Second, more sophisticated place recognition algorithm will be investigated to improve robust and reliable of loop closure detection. Third, how to deal with a large-scale environment also will be studied.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Nos. 61503403, 61573371) and the National University of Defense Technology Advanced Research Programs (Nos. JC14-03-04, JC14-03-06). We thank Yujie Wang and Tao Ma for helpful discussions.

References

Baird, E., Boeddeker, N., Ibbotson, M. R., & Srinivasan, M. V. (2013). A universal strategy for visually guided landing. *Proceedings of the National Academy of Sciences*, *110*(46), 18686–18691.

Boal, J., Sánchez-Miralles, A., & Arranz, A. (2014). Topological simultaneous localization and mapping: A survey. *Robotica*, *32*(05), 803–821.

Chu, J., Zhao, K., Zhang, Q., & Wang, T. (2008). Construction and performance test of a novel polarization sensor for navigation. *Sensors and Actuators A: Physical*, *148*(1), 75–82.

Cochran, W. W., Mouritsen, H., & Wikelski, M. (2004). Migrating songbirds recalibrate their magnetic compass daily from twilight cues. *Science*, *304*(5669), 405–408.

Collett, T. S., & Graham, P. (2004). Animal navigation: Path integration, visual landmarks and cognitive maps. *Current Biology*, *14*(12), R475–R477.

Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, *27*(6), 647–665.

Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1052–1067.

Denuelle, A., & Srinivasan, M. V. (2015). Snapshot-based navigation for the guidance of UAS. In *Australasian Conference on Robotics and Automation (ACRA 2015)*.

Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, *58*, 15–16.

Etienne, A. S., & Jeffery, K. J. (2004). Path integration in mammals. *Hippocampus*, *14*(2), 180–192.

Gagliardo, A. (2013). Forty years of olfactory navigation in birds. *The Journal of experimental biology*, *216*(12), 2165–2171.

Garratt, M. A., & Chahl, J. S. (2008). Vision-based terrain following for an unmanned rotorcraft. *Journal of Field Robotics*, *25*(4), 284.

Geiger, A., Ziegler, J., & Stiller, C. (2011). Stereoscan: Dense 3D reconstruction in real-time. In *2011 IEEE, Intelligent Vehicles Symposium (IV)* (pp. 963–968). IEEE.

Geva-Sagiv, M., Las, L., Yovel, Y., & Ulanovsky, N. (2015). Spatial cognition in bats and rats: From sensory acquisition to multiscale maps and navigation. *Nature Reviews Neuroscience*, *16*(2), 94–108.

Grisetti, G., Kümmerle, R., Stachniss, C., & Burgard, W. (2010). A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, *2*(4), 31–43.

Kitt, B., Geiger, A., & Lategahn, H. (2010). Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *2010 IEEE Intelligent Vehicles Symposium (IV)* (pp. 486–492). IEEE.

Lambrinos, D., Kobayashi, H., Pfeifer, R., Maris, M., Labhart, T., & Wehner, R. (1997). An autonomous agent navigating with a polarized light compass. *Adaptive Behavior*, *6*(1), 131–161.

Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., & Wehner, R. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, *30*(1), 39–64.

Lategahn, H., Geiger, A., & Kitt, B. (2011). Visual SLAM for autonomous ground vehicles. In *2011 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1732–1737). IEEE.

Milford, M. J., & Wyeth, G. F. (2008). Mapping a suburb with a single camera using a biologically inspired SLAM system. *IEEE Transactions on Robotics*, *24*(5), 1038–1053.

Muheim, R., Phillips, J. B., & Åkesson, S. (2006). Polarized light cues underlie compass calibration in migratory songbirds. *Science*, *313*(5788), 837–839.

O'keefe, J., & Nadel, L. (1978). *The hippocampus as a cognitive map* (Vol. 3). Oxford: Clarendon Press.

Olson, E., Leonard, J., & Teller, S. (2006). Fast iterative alignment of pose graphs with poor initial estimates. In *ICRA 2006, Proceedings 2006 IEEE International Conference on Robotics and Automation* (pp. 2262–2269). IEEE.

Scaramuzza, D., & Fraundorfer, F. (2011). Visual odometry [tutorial]. *IEEE Robotics & Automation Magazine*, *18*(4), 80–92.

Srinivasan, M. V., Chahl, J. S., Weber, K., Venkatesh, S., Nagle, M. G., & Zhang, S.-W. (1999). Robot navigation inspired by principles of insect vision. *Robotics and Autonomous Systems*, *26*(2), 203–216.

Wang, Y., Hu, X., Lian, J., Zhang, L., Xian, Z., & Ma, T. (2014). Design of a device for sky light polarization measurements. *Sensors*, *14*(8), 14916–14931.

Xian, Z., Hu, X., Lian, J., Zhang, L., Cao, J., Wang, Y., et al. (2014). A novel angle computation and calibration algorithm of bio-inspired sky-light polarization navigation sensor. *Sensors*, *14*(9), 17068–17088.

Zhiwen Xian received the B.S. degree in navigation and control from National University of Defense Technology (NUDT), Changsha, China, in 2010. From December 2014 to December 2015, he was a visiting scholar at Australian Centre for Field Robotics, the University of Sydney. He is currently pursuing his Ph.D. at NUDT, and his research topic is bio-inspired multiple sensors fusion for autonomous navigation.

Xiaofeng He received the B.E. and Ph.D. degrees from National University of Defense Technology (NUDT), China, in 2001 and 2009, respectively. From Januray to July 2015, he was a Visiting Scholar at the Institute of Computer Science, Christian-Albrechts-University of Kiel, Germany. He is currently a leturer with NUDT. His research interests include GNSS/IMU integrated system, and bio-inspired navigation.

Junxiang Lian received the B.E. and Ph.D. degrees from National University of Defense Technology (NUDT), China, in 2000 and 2007, respectively. From March to August 2012, he was a Visiting Scholar with the Division of Control, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently an associate professor with NUDT. His research interests include vision/inertial measuring unit fusion, and integrated navigation.

Xiaoping Hu received the B.Sc. and M.Sc. degrees in automatic control systems and aerocraft designing from the Department of Automatic Control, National University of Defense Technology (NUDT), Changsha, P. R. China, in 1982 and 1985, respectively. Currently he is with NUDT as a professor. His scientific interests include inertial and satellite navigation, aerocraft guidance and control and bio-inspired navigation.

Lilian Zhang is a lecturer at National University of Defense Technology (NUDT), China. He received his Dr. Degree at the Institute of Computer Science, Christian-Albrechts-University of Kiel, Germany in 2013 and his B.S. degree from NUDT in 2007. His current research is on image processing, structure from motion and vision navigation.