

Parallax Angle Parametrization in Incremental SLAM

Ellon Paiva Mendes and Simon Lacroix
LAAS-CNRS
Université de Toulouse, CNRS
Toulouse, France
Email: [ellon.mendes, simon.lacroix]@laas.fr

Joan Solà
Institut de Robòtica i Informàtica Industrial
CSIC-UPC
Barcelona, Spain
Email: jsola@iri.upc.edu

Abstract—The lack of depth information in camera images has triggered much work on their use for localization and mapping in robotics. In particular, specific landmark parametrizations that isolate the unknown depth in one variable, and that allows to handle the associated large uncertainties have been proposed. Recently, an innovative parametrization (Parallax Angle) has shown to outperform the others in the context of a Bundle Adjustment approach. This paper investigates the way to exploit this parametrization in an incremental graph-based SLAM approach, in a robotics context in which motions measures can be incorporated in the overall estimation. It presents the factors required to initialize landmarks and manage their observations. Simulation results show that the proposed algorithms are able to incrementally incorporate observations, and a discussion analyzes how the incremental updates on ISAM2 are affected by these new factors.

I. INTRODUCTION

Among the sensors available for robots, cameras remain the smallest, lightest and the most power efficient. The possibility to detect and track environment features during motion make them a solution of choice for the localization problem. Yet the loss of depth information due to the perspective projection challenges the position estimation approaches, all the more when the camera is facing towards the main direction of motion.

For instance, the cameras exploited for localization on board micro-UAVs are always oriented downwards (see *e.g.* [1]), so as to yield a good observability of depth, the UAVs evolving mostly in horizontal directions. On the contrary, autonomous ground robots are usually equipped with forward looking cameras, or endowing a UAV with a visual sensing and avoidance capacity requires the use of forwards facing cameras. In open areas, landmarks that lie far ahead are good to correct orientation errors, but hardly provide information about translation. On the other hand, close landmarks can be used to correct translation errors *if* their depth from the image is known, an information that is not available from a single image and may not be easily recoverable when the robot is moving along the camera focal axis.

Ideally, one should be able to handle both close and far landmarks in a unique manner. One of the way to achieve this is to find an alternate way to represent a point in space than the Euclidean XYZ coordinates. Several different parametrizations for monocular SLAM are compared in [2]. Among them, the

inverse depth parametrization (IDP) [3], is one of the most well known parametrizations for point landmarks. It allows *undelayed landmark initialization* by isolating the unknown depth in one parameter (the inverse of depth $\rho = 1/\text{depth}$), initializing it with large uncertainty and estimating it in an Extended Kalman Filter framework, along with an anchor position and the azimuth and elevation angles that encode the vector from the anchor to the point. In the context of a Bundle Adjustment (BA) approach, [4] proposed the Parallax Angle Parametrization (PAP), which constitutes an improvement over IDP, especially for features lying in the direction of the camera motion. The authors show that this parametrization avoids ill-conditioned equations and yields fast convergence, the objective function having no small gradients with respect to the introduced parameters to describe the features location.

This paper presents a way to exploit the Parallax Angle Parametrization in an Incremental Graphical SLAM approach. Indeed since the PAP has shown to well handle situations where landmarks have very low observability in an optimization solution, it is worth to adopt it in an incremental context. Yet exploiting alternate landmark parametrizations, and in particular parametrizations that exploit anchors, and hence induces additional dependencies, require a careful definition of the factors and variables to be introduced in the factor graph formulation.

The paper is organized as follows: Section II briefly describes the concepts behind PAP. Section III then explains how the monocular SLAM problem with PAP can be modeled as a factor graph, and depicts the various mechanisms required to handle this parametrization in an incremental manner. In particular, it proposes a way to re-anchor landmarks so as to yield a better observability when new observations occur. The overall incremental algorithm is given in Section IV. Section V analyses the integration of the proposed approach within the ISAM2 optimization back-end [5], on the basis of a simulation.

II. PARALLAX ANGLE PARAMETRIZATION

The Parallax Angle Parametrization (PAP) was first presented in [4] and further analyzed in [6]. Its main characteristic is that it uses *two* anchors (instead of one for the IDP), and the depth is encoded in a *parallax angle* between vectors from these anchors to the point. This section summarizes the

PAP definition and how initialization and observations are handled – readers interested in more details should refer to the aforementioned papers.

A. Definition

Let $C = [\mathbf{t} \ \mathbf{q}]^\top$ be a 3D camera frame, with position $\mathbf{t} \in \mathbb{R}^3$ and orientation represented as a quaternion $\mathbf{q} \in \mathbb{R}^4$. An image feature \mathbf{l}_j of a 3D point \mathbf{p}_j observed from two cameras C_m, C_a is represented in Parallax Angle Parametrization as:

$$\mathbf{l}_j = [\psi_j \ \theta_j \ \omega_j]^\top \quad (1)$$

where ψ_j and θ_j are the azimuth and elevation angles in a frame centered at \mathbf{t}_m and oriented as the global frame, and ω_j is the parallax angle, which is the angle between the vectors $\overrightarrow{\mathbf{t}_m \mathbf{p}_j}$ and $\overrightarrow{\mathbf{t}_a \mathbf{p}_j}$ (see Fig. 1 for a 2D example). The feature \mathbf{l}_j is anchored to the frames C_m and C_a , it has \mathbf{t}_m as the *main anchor*, and \mathbf{t}_a as the *associated anchor*.

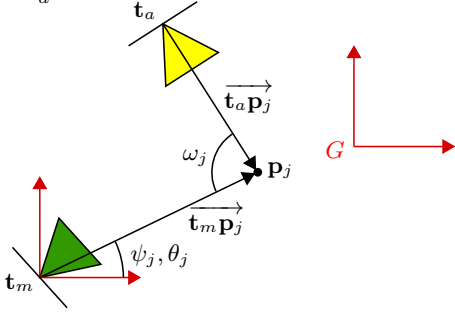


Fig. 1. Geometrical relations in the PAP parameters. For simplicity the relations are drawn in 2D, from a top view, and hence the angles ψ_j and θ_j are represented as the same arc. The triangles show the field of view of the cameras. The main anchor camera is in green, the associated anchor camera is in yellow, and the global frame is in red.

Throughout the paper, main anchor related variables are written as $(\cdot)_m$, associated anchor related variables as $(\cdot)_a$, variables not related with any of the anchors as $(\cdot)_o$ (which stands for “other than anchors”), and landmark related variables as $(\cdot)_j$. Camera frames written as C_i represents any camera frame, possibly the main frame C_m or the associated frame C_a .

B. Observation Function

Given a general camera frame $C_i = [\mathbf{t}_i \ \mathbf{q}_i]^\top$ and a PAP representation \mathbf{l}_j of a 3D point \mathbf{p}_j , let \mathbf{v}_i be the vector in the global frame that goes from the camera center at \mathbf{t}_i towards \mathbf{p}_j (Fig. 2). \mathbf{v}_i is computed in two distinct ways:

$$\mathbf{v}_i = \begin{cases} \mathbf{v}_m & \text{if } C_i = C_m \\ \tilde{\mathbf{v}}_i & \text{if } C_i \neq C_m \end{cases} \quad (2)$$

When $C_i = C_m$, $\mathbf{v}_i = \mathbf{v}_m$ is the direction vector from the main anchor to the point, given by¹:

$$\mathbf{v}_m = v(\psi_j, \theta_j) = \begin{bmatrix} \cos \psi_j \cos \theta_j \\ \sin \psi_j \cos \theta_j \\ \sin \theta_j \end{bmatrix}. \quad (3)$$

¹Here we use the *Front-Left-Up* (FLU) frame convention [7, Section 1.2.1], for the XYZ axes, instead of *Right-Down-Forward* (RDF) convention of the original PAP papers.

When $C_i \neq C_m$, $\mathbf{v}_i = \tilde{\mathbf{v}}_i$ is the vector proportional to $\overrightarrow{\mathbf{t}_i \mathbf{p}_j}$, given by [4], [6]:

$$\begin{aligned} \tilde{\mathbf{v}}_i &\triangleq \sin \omega_j \overrightarrow{\mathbf{t}_i \mathbf{p}_j} \\ &= \sin(\omega_j + \varphi) \|\overrightarrow{\mathbf{t}_m \mathbf{t}_a}\| \mathbf{v}_m - \sin \omega_j \overrightarrow{\mathbf{t}_m \mathbf{t}_i} \end{aligned} \quad (4)$$

where φ is the angle between $\overrightarrow{\mathbf{t}_m \mathbf{t}_a}$ and \mathbf{v}_m , and can be computed as²

$$\varphi = \text{atan2} \left(\|\mathbf{v}_m \times \overrightarrow{\mathbf{t}_m \mathbf{t}_a}\|, \mathbf{v}_m \cdot \overrightarrow{\mathbf{t}_m \mathbf{t}_a} \right) \quad (5)$$

The image projection $\mathbf{z}_i = [u_i \ v_i]^\top$ of a point \mathbf{p}_j to a camera C_i can then be computed as

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \propto \mathbf{K} \mathbf{R}(\mathbf{q}_i)^\top \mathbf{v}_i \quad (6)$$

with \mathbf{K} being the camera calibration matrix, and $\mathbf{R}(\mathbf{q})$ being the rotation matrix representation of the quaternion \mathbf{q} . A 2D representation of the variables that define this observation function can be seen in Fig. 2.

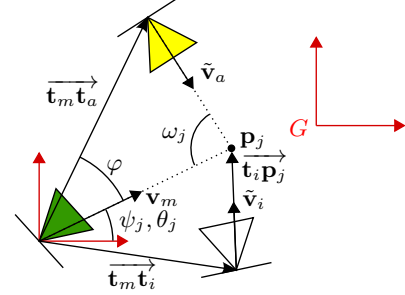


Fig. 2. Variables involved in the observation function in PAP (again, the relations are drawn in 2D, from a top view). The white triangle represents the field of view of a camera other than the main and associated cameras. The vector \mathbf{v}_m , the vector $\tilde{\mathbf{v}}_i$ and the angle φ are respectively computed using equations (3), (4), and (5). The vector $\tilde{\mathbf{v}}_a$ can be also computed from (4) when $i = a$.

C. Initialization From Measurements

Given a feature \mathbf{l}_j first observed from a camera at C_m , its azimuth and elevation angles in PAP are computed as:

$$\begin{aligned} \psi_j &= \text{atan2}(\hat{y}_m, \hat{x}_m) \\ \theta_j &= \text{atan2} \left(\hat{z}_m, \sqrt{(\hat{x}_m)^2 + (\hat{y}_m)^2} \right) \end{aligned} \quad (7)$$

where $\hat{\mathbf{v}}_m$ is the vector computed by back-projecting the image measurement $\mathbf{z}_m = [u_m \ v_m]^\top$ taken from C_m :

$$\hat{\mathbf{v}}_m = \begin{bmatrix} \hat{x}_m \\ \hat{y}_m \\ \hat{z}_m \end{bmatrix} = \mathbf{R}(\mathbf{q}_m) \mathbf{K}^{-1} \begin{bmatrix} u_m \\ v_m \\ 1 \end{bmatrix}. \quad (8)$$

If the feature is re-observed from a camera at C_a , then the parallax angle ω_j can be initialized as

$$\omega_j = \text{atan2}(\|\hat{\mathbf{v}}_m \times \hat{\mathbf{v}}_a\|, \hat{\mathbf{v}}_m \cdot \hat{\mathbf{v}}_a) \quad (9)$$

where $\hat{\mathbf{v}}_a$ is the vector computed by back-projecting the new measurement \mathbf{z}_a , with an expression akin to $\hat{\mathbf{v}}_m$ in (8).

²In the original PAP papers this angle was computed using arccos, we opt for a more stable version using atan2

III. PAP IN INCREMENTAL GRAPHICAL SLAM

A. Factor Graphs

This section is a brief description of graph-based SLAM – interested readers may find more information on the subject in the graph-based SLAM tutorial [8].

A *factor graph* [9] is a bipartite graph $\mathcal{G} = (\mathcal{F}, \mathcal{V}, \mathcal{E})$ with factor nodes $f_r \in \mathcal{F}$ and variable nodes $v_s \in \mathcal{V}$. Edges $e_{rs} \in \mathcal{E}$ always connect factor nodes and variable nodes. The factor graph defines factorization of a function $g(\mathcal{V})$ as:

$$g(\mathcal{V}) = \prod_r f_r(\mathcal{V}_r) \quad (10)$$

with \mathcal{V}_r being the set of variables v_s adjacent to f_r (i.e. variables connected to f_r through an edge e_{rs}). Thus each factor f_r is a function of the variables \mathcal{V}_r . The set \mathcal{E} can be implicitly defined by \mathcal{V}_r and is generally omitted.

Each factor f_r encodes a measurement function $h_r(\mathcal{V}_r)$ and a measurement z_r . When assuming Gaussian measurement models, f_r is defined as:

$$f_r(\mathcal{V}_r) \propto \exp\left(-\frac{1}{2}\|h_r(\mathcal{V}_r) - z_r\|_{\Sigma}^2\right) \quad (11)$$

with $\|e\|_{\Sigma}^2 = e^T \Sigma^{-1} e$ being the Mahalanobis distance with covariance matrix Σ . Then, finding the configuration for the variable nodes \mathcal{V}^* that maximizes (10) is the same as solving the nonlinear least-squares problem:

$$\mathcal{V}^* = \arg \min_{\mathcal{V}} \frac{1}{2} \sum_i \|h_r(\mathcal{V}_r) - z_r\|_{\Sigma}^2 \quad (12)$$

A typical factor graph for monocular SLAM with landmarks parametrized as XYZ coordinates is shown in Fig. 3.

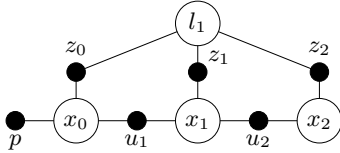


Fig. 3. Typical factor graph of the monocular SLAM when landmarks are parametrized as XYZ coordinates. Variable nodes are shown in large circles and factor nodes as small solid circles ($\mathcal{V} = \{x_0, x_1, x_2, l_1\}$, $\mathcal{F} = \{p, u_1, u_2, z_0, z_1, z_2\}$). The factors represent a prior information p about the origin, odometry measurements u , and landmark measurements z . Note that every factors z here are of the same type, connecting only to the landmark and the camera pose from which the measurement was taken.

B. Projection Factors for PAP

When using PAP in a graphical SLAM, the observation function described in Section II-B involves variables in up to four different nodes (main anchor, associated anchor, camera pose, and landmark), and is represented by three different factors, as shown in Fig. 4. The ordinary projection factor is shown Fig. 4(c), and uses equation (4). The other two are special cases when the measurements are taken from one of the anchor frames. The factor shown in Fig. 4(a) is called *main factor* and uses the equation (3), thus it only depends on x_m . The factor on Fig. 4(b) is the special case of equation (4)

when $i = a$, and it is called *associated factor*. Note that for each fully initialized landmark there is always one main and one associated factor, all other factors for the same landmark being of the ordinary type.

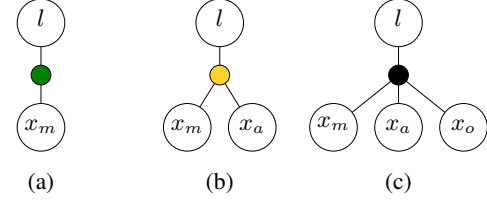


Fig. 4. Three possible projection factors for a PAP feature. (a) main factor; (b) associated factor; and (c) ordinary factor. The subscripts for the camera variables represents the anchors: m for the main anchor frame, a for the associated anchor frame, and o for all the other cameras from where the landmark was observed.

Fig. 5 shows a monocular SLAM factor graph using these specific PAP factors. The main difference between this graph and the one on Fig. 3 is the increased number of edges due to the use of projection factors that connect to more than one camera frame.

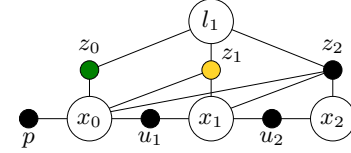


Fig. 5. Reformulation of the SLAM problem using the specific PAP factors. The color code is the same as in the former figures: the measurements from the main and associated anchors are respectively in green and yellow. Here the landmark l_1 is anchored in the camera frames represented by x_0 and x_1 .

C. Incremental Anchor Selection

PAP was originally conceived to be used in *Bundle Adjustment* (BA). In this context, initial estimates of the camera frames are computed from two-view geometry techniques, like the eight point algorithm. The parallax angle is computed using the initial camera frame estimates for each pair of measurements to a landmark: the camera frames that provides the maximum parallax angle or a parallax angle larger than a threshold are selected as landmark anchors. This threshold is used to stop the anchor search when the parallax is good enough, avoiding unnecessary computation. The anchors are then kept fixed during the optimization phase.

For PAP to perform well it is important to choose anchors that provide a good observability of the landmark depth, encoded in the parallax angle. In an incremental SLAM solution, it is not possible to know in advance which camera frames will be the best anchors for a landmark: it is necessary to allow the anchors to be actively selected when new observations are made. This is done in two steps: first the landmark should be *anchored*, and then possibly be *re-anchored* if better anchor frames are found.

To *anchor* a landmark it is necessary to decide which frames are adequate. Indeed the anchors should not define vectors \mathbf{v}_m (equation (3)) and $\mathbf{t}_m \mathbf{t}_a$ that are collinear, otherwise the associated and ordinary factors of this landmark will not be

able to compute the expected image measurement using (6), because $\tilde{\mathbf{v}}_i$ obtained from (4) will be zero due to coefficients $\sin(\omega_j + \varphi)$ and $\sin \omega_j$. Fig. 6(a)-(b) shows the cases to avoid when anchoring a landmark. This problem is more likely to happen when the camera is moving along its focal axis and observing a landmark situated on the focal point. Note that cases when the point is very far away (zero or near zero initial parallax) but the anchors are not collinear should not cause problems in PAP (see Fig. 6(d)).

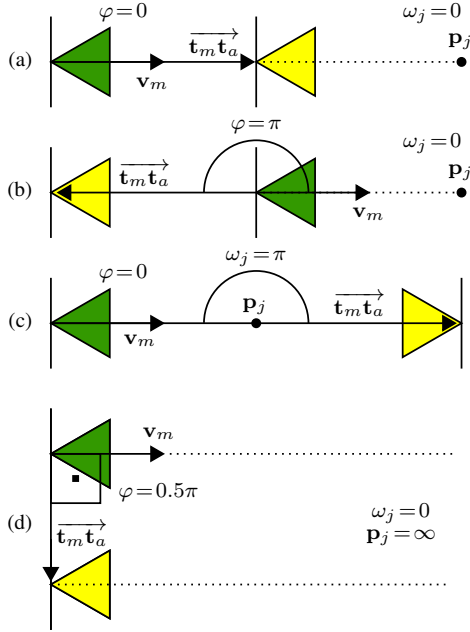


Fig. 6. Particular geometrical configurations of anchors, φ and ω_j (again presented in 2D, compare with Fig. 2): (a) The associated anchor is between the main anchor and point; (b) The main anchor is between the associated anchor and the point; (c) The point is in the middle of anchors; (d) The point is at infinity and the anchors are not aligned. Configurations (a) and (b) should be avoided because they produce a vector $\tilde{\mathbf{v}}_i = \mathbf{0}$ in Equ. (4) that cannot be properly projected by Equ. (6). (c) will hardly occur in practice because the re-anchoring process stops once the anchors provide good parallax angle for the landmark (not to mention that visual features can hardly be matched in such conditions). (d) shows that PAP can handle points at infinity as long as the anchors are not aligned with the point.

To *re-anchor* a landmark, a re-anchoring test is performed *after the optimization step*, whenever an anchored landmark is observed from a new camera frame and the current parallax is below a threshold: equation (9) is used to compute a new parallax angle for each main-new and associated-new frame pairs, using the current frame values and the respective measurements. If any of the pairs provides a better parallax, the elements of this pair are selected as the new anchors. Then, to re-anchor the landmark:

- If the main-new pair is selected, we only change the associated anchor. For this all landmark's projection factors but the main factor are linked to the new associated anchor, and the landmark's ω is updated to reflect the new anchor;
- If the associated-new pair is selected, both anchors change. All landmark's projection factors are updated, and the landmark variable is fully updated as well (ψ , θ should

reflect the new main anchor, and ω the new anchor pair).

- If none of the pairs are selected, the landmark measurement is kept as an ordinary projection factor.

This re-anchoring procedure is shown in Fig. 7. Note that if re-anchoring is performed, the type of some of the projection factors will be updated, implying that they will use different measurement functions to compute the error.

IV. INCREMENTAL PAP ALGORITHM

The incremental optimization with PAP is described in algorithm 1. The initialization step goes from lines 1 to 4. Here the origin and the prior factor are normally set to the identity frame. The prior covariance is set to be very small, but not zero (line 2). Setting it to zero would introduce a hard constraint to the system that can cause problems during optimization. In lines 3 and 4 we store a frame-measurement pair for each landmark observed from the origin as main anchor information for that landmark. The stored anchor information is used later to initialize the landmark variable and to create its projection factors.

The main loop of the algorithm starts at line 5. For each new camera frame and odometry measurement we initialize a new frame variable by composing the previous frame with the odometry measurement (lines 6 and 7), and we add a odometry factor between subsequent camera frames to the graph (line 8).

Then, from lines 9 to 20 we handle the landmark measurements from this new frame: main anchor information is stored for new landmarks (lines 10 and 11) and azimuth and elevation angles for the new landmark are computed (line 12); landmarks being seen that are not yet anchored test if the new frame provides a good associated anchor (line 14) and if the test succeeds they have their associated anchor information stored (line 15), their parallax angle computed (line 16), their anchor factors created and added to the graph (line 17), and they are considered anchored and initialized (line 18); and observed landmarks that were already anchored receive a new ordinary projection factor (lines 19 and 20).

Next, the current factor graph is solved (line 21) and the variables are updated with the optimization solution (line 22).

In the final part of the main loop (lines 23 to 27) the re-anchoring test is performed for every landmark observed from the new frame, and updates in the factor graph and the landmark variables are performed if needed, as described in Section III-C. Note that the re-anchoring step is performed only *after* the optimization step. This is needed to guarantee that the tested parallax angles use frame values compatible with the anchor frame values, *i.e.* that all frame values used in the test come from the solution of the optimization.

V. ANALYSIS

The experiments were realized in a simulated environment using SLAMTB³, a SLAM Toolbox for Matlab. They consisted in a robot represented as a 3D pose moving in a plane, in an environment filled with 3D landmarks. The robot is equipped

³github.com/joansola/slamtb

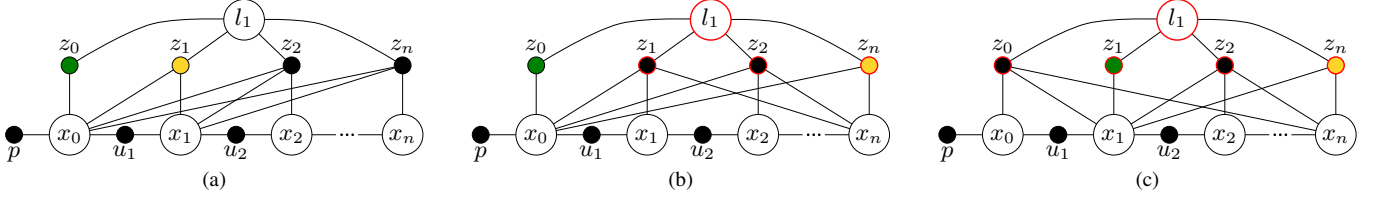


Fig. 7. Landmark re-anchoring after addition of a factor z_n . All affected factors and variables are highlighted in red. (a) New measurement does not provide better parallax or current landmark's parallax angle ω is above a threshold, so the measurement is kept as an ordinary factor; (b) A new measurement provides better parallax wrt. the main anchor: x_n becomes the associated anchor, the graph is updated, and landmark's ω is updated using (9); (c) A new measurement provides better parallax wrt. the associated anchor: the current associated anchor becomes the new main anchor, x_n becomes the new associated anchor, the graph is updated, and landmark's ψ, θ and ω are updated using (7) and (9) according to the new anchors.

Algorithm 1 Incremental PAP SLAM

```

1: Add camera frame  $C_0$  as origin  $x_0$  to  $\mathcal{V}$ 
2: Add prior about the origin as a factor  $p$  to  $\mathcal{F}$ 
3: for each measurement  $z_{0j}$  of landmark  $j$  from  $C_0$  do
4:   Store  $(C_0, z_{0j})$  as main anchor of landmark  $j$ 

5: for each new camera frame  $C_i$  and new odometry  $u_i$  do
6:   Use  $x_{i-1}$  in  $\mathcal{V}$  and  $u_i$  to initialize variable  $x_i$ 
7:   Add variable  $x_i$  to  $\mathcal{V}$ 
8:   Add  $u_i$  as odometry factor between  $x_i$  and  $x_{i-1}$  to  $\mathcal{F}$ 
9:   for each measurement  $z_{ij}$  of landmark  $j$  from  $C_i$  do
10:    if  $z_{ij}$  is first measurement of  $j$  then
11:      Store  $(C_i, z_{ij})$  as main anchor of landmark  $j$ 
12:      Compute  $\psi_j$  and  $\theta_j$  of  $l_j$  ▷ eq. (7)
13:    else if landmark  $j$  is not anchored then
14:      if  $C_{mj}$  and  $C_i$  are not collinear with landmark  $j$  then
15:        Store  $(C_i, z_{ij})$  as associated anchor of landmark  $j$ 
16:        Compute  $\omega_j$  of  $l_j$  ▷ eq. (9)
17:        Add anchor factors of  $l_j$  to  $\mathcal{F}$  ▷ fig. 5
18:        Add variable  $l_j$  to  $\mathcal{V}$ 
19:    else if landmark  $j$  is anchored then
20:      Add it as an ordinary factor of  $l_j$  to  $\mathcal{F}$  ▷ fig. 5

21: Solve optimization problem using  $\mathcal{F}$  and  $\mathcal{V}$  ▷ eq. (12)
22: Update  $\mathcal{V}$  with current solution

23: for each measurement  $z_{ij}$  of landmark  $j$  from  $C_i$  do
24:   if landmark  $j$  has three or more projection factors then
25:     Perform re-anchoring test for landmark  $j$  ▷ sec. III-C
26:   if landmark  $j$  should be reanchored then
27:     Update  $\mathcal{F}$  and  $\mathcal{V}$  to match new anchors ▷ fig. 7

```

with an odometry sensor and one camera, which is positioned with the focal axis looking forward. Key-frames were selected after a certain number of odometry measurements. These measurements were integrated and the result is used as odometry measurements in Algorithm 1, together with image measurements of landmarks visible from the key-frame.

The front-end used is similar to the one used in RT-SLAM [10]: we limit the number of measurements in each key-frame to the first N_{up} measurements to landmarks whose projections provides the best innovations. We always try to initialize N_{init} new landmarks every key-frame to deal with possible landmarks going out of the field-of-view.

The optimization back-end used ISAM2 [5] as it is implemented on *GTSAM 3.1.0*. The PAP variables with initialization methods and the different projection factors described in this work are implemented in GTSAM and are available

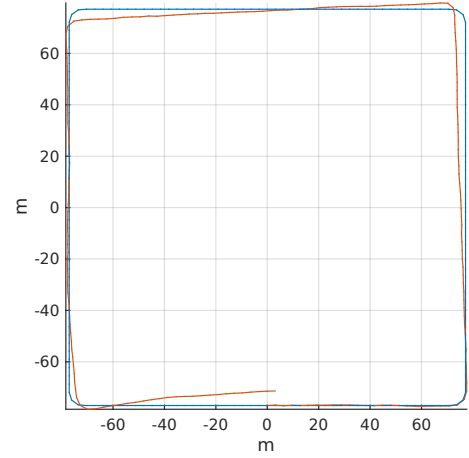


Fig. 8. Robot Trajectory for a long run: estimated in red, and simulated in blue

at github.com/Elton/gtsam. As a wrapper is used to access GTSAM libraries from Matlab, it is impossible to have a computational analysis of time performance of the algorithm, although some theoretical analysis are discussed below.

The algorithm described in this paper is placed in-between the SLAMTB front-end and the ISAM2 back-end. It decides which landmarks can be anchored and initialized and when, dealing also with projection factor creation and possible removals, modifications and re-insertions of these factors in the factor graph inside ISAM2 due to incremental re-anchoring. Contrary to the original PAP papers, here we estimate the robot frames instead of camera frames. While this increases the non-linearity of the system, it is the easiest way to fuse data from multiple sensors (odometry and camera in our case). The landmark anchors are taken from the composition of estimated robot frames with the camera frame wrt. the body frame.

The trajectory estimated for a long run along a squared path can be seen in Fig. 8.

A. Performance on ISAM2

The decision to use ISAM2 as the incremental optimization back-end was motivated by its popularity and reported performance when used in SLAM solutions. Here we present the results of using PAP with ISAM2.

The main features of ISAM2 are:

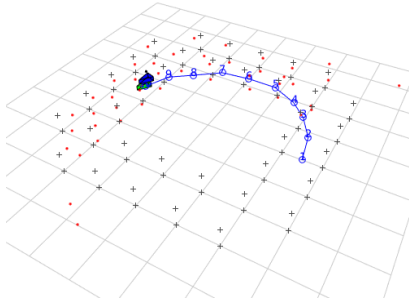


Fig. 9. Scenario used for testing of incremental updates. The polygons are the simulated robot: simulated in blue, and estimated (little below the simulated) in green. The crosses are the simulated landmarks. Red marks are estimated landmarks and the blue circles and line are the key-frames and trajectory.

TABLE I
AVERAGE SIZE OF FRONTAL VARIABLES AND SEPARATORS OF BAYES TREES

	Front Variables	Separator set
Euclidean	1.33	9.19
PAP	2.09	11.31

- 1) restructure the factor graph in a tree of cluster-nodes called Bayes Tree [11] in a way incremental tends to modify only a small part of the tree;
- 2) the ability perform partial updates of the variables to reduce the computational cost while recovering a nearly exact solution; and
- 3) track the validity of the linearization point for each variable *and* only re-linearize when needed.

1) *Incremental Updates:* We investigated the impact of the extra edges needed by the associated and ordinary projection factors from Section III-B on the incremental update. We compared the Bayes trees built for the monocular SLAM using Euclidean XYZ and using PAP, run over the scenario shown in Fig. 9. The differences between the average number of frontal variables and the sizes of the separator is shown in Table I. They suggest a denser fill-in of the system matrices when using PAP, which yields slower incremental updates. The Hessian matrices in Fig. 10 confirm the increase of the fill-in caused by the additional edges needed to implement PAP with factor graphs.

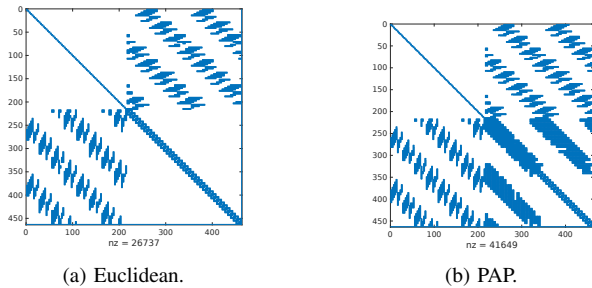


Fig. 10. Hessian matrices for the SLAM problem using Euclidean and PAP. The PAP hessian has a higher number of non-zeros (41649) wrt. the Euclidean solution hessian (26733).

2) *Sensibility to ISAM2 parameters:* The properties “2” and “3” of ISAM2 are controlled respectively by two parameters α and β . They were found to have great impact on the results, especially β . Setting β to a high value trigger re-linearizations less often, providing faster updates. We suspect by estimating the landmarks values in the system we increase the non-linearity of the system, being necessary to keep the system Jacobians up to date to achieve a good solution. In the experiments we normally set this parameter to a very low value.

B. Projection Factors and Cheirality

The measurements associated to a projection factor are the image coordinates of the landmark (i.e. their projection on the camera). To evaluate the error contribution of a projection factor (and to derive the jacobians of the observation function) we use the equations from Section II-B. For a projection to be valid, the vector $\mathbf{R}(\mathbf{q}_i)^\top \tilde{\mathbf{v}}_i$ in (6) needs to be in the *front side* of the image plane of the camera at C_i , also named *cheirality* [12]. If the cheirality is negative, the point lays behind the camera, and then the projection factor should not be considered (the jacobians are set to zero). This can be dangerous for monocular vision systems because it may reduce the rank of the matrix used in the optimization step that is built from the factor graph, possibly leading to *ill- conditioned local systems* that cannot be solved.

When using PAP points, we noticed that sometimes the estimated points are indeed behind the camera. This mostly occurs with points lying on the ground and during the first few observations. Since the robot is always moving forward in our test case, a landmark estimated by a PAP that is close to the robot and with a reduced observability (for example, with only measurements from its anchors) may have big changes on its depth wrt. the main anchor on each iteration of the algorithm, and possibly move behind another camera that has a measurement to it. We noticed that the point became stable as more measurements are incorporated and the re-anchoring is applied.

VI. CONCLUSIONS

This work presented how to use Parallax Angle Parametrization (PAP) for 3D points in a incremental graphical SLAM setting. The main difference from the original PAP approach being the incremental selection of anchors, and the estimation of robot frames instead of camera frames. We presented the factors needed to project a PAP points into the cameras, and how to re-anchor then when needed to obtain a better parallax. Some analysis of the performance in the incremental estimation were presented, based on simulations. We used ISAM2 as the incremental optimization back-end and the projection factors for PAP implemented on GTSAM framework were made publicly available.

Possible future extensions to this work could be to find a better elimination order that increases the sparsity when using PAP, and performing the re-anchoring inside ISAM2 by only reconnecting edges, without the need to explicitly remove and reinsert the modified factors in the factor graph.

REFERENCES

- [1] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-Based Navigation for Autonomous Micro Helicopters in GPS-denied Environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, Nov. 2011.
- [2] J. Solà, T. Vidal-Calleja, J. Civera, and J. M. M. Montiel, "Impact of landmark parametrization on monocular ekf-slam with points and lines," *International Journal of Computer Vision*, vol. 97, no. 3, pp. 339–368, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s11263-011-0492-5>
- [3] J. Civera, A. Davison, and J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 932–945, Oct 2008.
- [4] L. Zhao, S. Huang, L. Yan, and G. Dissanayake, "Parallax angle parametrization for monocular SLAM," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 3117–3124.
- [5] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012. [Online]. Available: <http://ijr.sagepub.com/content/31/2/216.abstract>
- [6] L. Zhao, S. Huang, Y. Sun, L. Yan, and G. Dissanayake, "Parallaxba: bundle adjustment using parallax angle feature parametrization," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 493–516, 2015. [Online]. Available: <http://ijr.sagepub.com/content/34/4-5/493.abstract>
- [7] J. Solà, "Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach." Ph.D. dissertation, École Doctorale Systèmes, 2007.
- [8] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, no. 4, pp. 31–43, winter 2010.
- [9] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.
- [10] C. Roussillon, A. Gonzalez, J. Solà, J.-M. Codol, N. Mansard, S. Lacroix, and M. Devy, "Rt-slam: a generic and real-time slam architecture," in *International Conference on Vision Systems, Sophia Antopolis (France)*, Sept. 2011.
- [11] M. Kaess, V. Ila, R. Roberts, and F. Dellaert, "The bayes tree: An algorithmic foundation for probabilistic robot mapping," in *Algorithmic Foundations of Robotics IX*, ser. Springer Tracts in Advanced Robotics, D. Hsu, V. Isler, J.-C. Latombe, and M. Lin, Eds. Springer Berlin Heidelberg, 2011, vol. 68, pp. 157–173. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-17452-0_10
- [12] R. I. Hartley, "Cheirality invariants," in *In proc. DARPA Image Understanding Workshop*, 1993, pp. 745–753.