Vision-based Target Tracking and Ego-Motion Estimation using Incremental Light Bundle Adjustment

Michael Chojnacki

Vision-based Target Tracking and Ego-Motion Estimation using Incremental Light Bundle Adjustment

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics and
Autonomous Systems

Michael Chojnacki

Submitted to the Senate
of the Technion — Israel Institute of Technology
Shevat 5777 Haifa February 2017

This research was carried out under the supervision of Asst. Prof. Vadim Indelman from the Faculty of Aerospace Engineering and under co-supervision of Prof. Ehud Rivlin from the Faculty of Computer Science, as part of the Technion Autonomous Systems Program at the Technion - Israel Institute of Technology

M. Chojnacki and V. Indelman. Vision-based target tracking and ego-motion estimation using incremental light bundle adjustment. In *Israel Robotics Conference*, 2016.

Contents

Abstract

This work presents a vision-based, computationally-efficient method for simultaneous robot motion estimation and dynamic target tracking, while operating in GPS-denied unknown or uncertain environments. While numerous vision-based approaches are able to achieve simultaneous ego-motion estimation along with detection and tracking of moving objects (DTMO), many of them require performing a bundle adjustment (BA) optimization, which involves the estimation of the 3D points observed in the process. One of the main concerns in robotics applications is the computational efforts required to sustain extended operation. The BA process is performed incrementally as new camera poses and new measurements arrive, constantly increasing the computational complexity of the problem. Considering applications for which the primarily interest is highly-accurate on-line navigation rather than mapping, the number of involved variables can be considerably reduced by avoiding the explicit 3D structure reconstruction and consequently save processing time. We take advantage of the light bundle adjustment (LBA) method [16], which allows for ego-motion calculation without the need for 3D points on-line reconstruction, and thus, significantly reduce computation time compared to BA. The proposed method integrates the target tracking problem into the LBA framework, yielding a simultaneous ego-motion estimation and tracking process, in which the target is the only explicitly on-line reconstructed 3D point. Furthermore, our method makes use of the recently developed Incremental Smoothing and Mapping (iSAM) [20] technique, which allows for re-use of calculations in order to further reduce the computational cost. Our approach is compared to BA and target tracking in terms of accuracy and computational complexity using simulated aerial scenarios and real-imagery experiments performed at the Autonomous Navigation and Perception Lab (ANPL) at the Technion.

List of Symbols

x, v	A generic point; a generic vector; a scalar; a variable
X	A matrix; a set of continuous random variables
$\hat{x}$	A unit vector
$\tilde{x}$	A vector with homogeneous coordinates
$f(x, y)$	Factor involving variables x and y
$\mathbb{R}^n$	The space of all n-tuples of Real numbers
$h(\cdot)$	A scalar function
$\bar{x}$	Linearization point for variable x
$t_{k \to l}$	Translation vector from the k^{th} view to the l^{th} view
R_G^C	Rotation matrix from frame C to frame G

Abbreviations and Notations

ANPL	:	Autonomous Navigation and Perception Lab
BA	:	Bundle Adjustment
DOF	:	Degree Of Freedom
DTMO	:	Detection and Tracking of Moving Objects
EKF	:	Extended Kalman Filter
GPS	:	Global Positioning System
iLBA	:	Incremental Light Bundle Adjustment
iSAM	:	Incremental Smoothing And Mapping
LBA	:	Light Bundle Adjustment
LOS	:	Line Of Sight
MAP	:	Maximum A Posteriori
RANSAC	:	Random Sample Consensus
RMSE	:	Root Mean Square Error
SFM	:	Structure From Motion
SLAM	:	Simultaneous Localization And Mapping
SLAMMOT	:	Simultaneous Localization, Mapping and Moving Objects Tracking
SVD	:	Single Value Decomposition
PDF	:	Probability Density Function

List of Figures

Chapter 1

Introduction

Ego-motion estimation and target tracking are core capabilities in a wide range of applications. While motion estimation is essential to numerous robotics tasks such as autonomous navigation [37, 39, 7, 12] and augmented reality [3, 43], target tracking has been a key capability, amongst others, for video surveillance [26] and for military purposes [4]. Although researched for decades, target tracking methods have mostly assumed a known or highly predictable sensor location. Recent robotics applications such as autonomous aerial urban surveillance [42] or indoor navigation require the ability to track dynamic objects from platforms while moving in unknown or uncertain environments. The ability to solve simultaneously the ego-motion and target tracking problems becomes therefore an important task. Furthermore, interest has grown for cases in which external localization systems (e.g. GPS) are unavailable and the estimation process must be performed using on-board sensors only. In particular, the capability to perform those tasks based on vision sensors has gain great attention in the past two decades, mostly thanks to the ever-growing advantages these sensors present [30].

Vision-based ego-motion estimation is typically performed in a process known as bundle adjustment (BA) in computer vision, or simultaneous localization and mapping (SLAM) in robotics, where the differences between the actual and the predicted image observations are minimized. Therefore, the combined process of SLAM and moving object tracking usually involves an optimization over the camera's motion states, the target's navigation states, and the observed structure (3D points/landmarks). This non-linear optimization is performed incrementally as new camera poses, new target states and new surrounding features are observed, constantly increasing the computational complexity of the problem. One of the main challenges in extended operation is thus keeping computational efforts to a minimum despite the constantly growing number of variables involved in the optimization.

However, many robotics applications do not require actual on-line mapping of the environment. Conceptually, avoiding the 3D structure reconstruction would allow to reduce the number of involved variables, and therefore, improve processing performances. Several "structure-less" BA approaches have been developed, where the optimization

satisfies constraints which do not involve 3D structure reconstruction. Such a method would therefore benefit the combined ego-motion and target tracking problem in terms of processing time. Moreover, these methods use batch optimization, which performs the whole estimation process from scratch at every step. Instead, incremental optimization methods can be used to update the solution using partial calculations, allowing for further computational savings.

1.1 Related Work

The simultaneous ego-motion and dynamic object tracking relates to numerous works on SLAM and target tracking, both individually and combined. The SLAM problem consists of building a map of an unknown environment while simultaneously localizing the mapping platform. While the problem's origins are found in the photogrammetry community, its application for robotic purposes appeared first in the late 1980's [34],[29], where techniques for estimating relative spatial relationships between different frames were presented. The early 1990's have seen further development towards practical implementations [25] and already recognized the need for computational efficiency. Early approaches used the Extended Kalman Filter (EKF) to solve the SLAM problem [6, 34], but were eventually overtaken by other techniques due to their quadratic computational complexity, which limits them to relatively small environments or to relatively small state vectors. Numerous SLAM methods have been proposed to overcome computational complexity, for example, by exploiting the sparsity of the involved matrices [27, 22], or by approximating the full problem with a reduced non-linear system [23]. A more recent technique, used in the frame of this work, performs incremental smoothing [20] to recover the solution while recalculating only part of the variables at each optimization step, reducing significantly the computational cost. Still, full BA methods involve the reconstruction of the 3D observed structure, which in case on-line mapping is of no interest, increases unnecessarily the number of estimated variables. Different structure-less BA methods have been proposed to address this issue. Rodriguez et al. [33] use epipolar constraints between pairs of views while Steffen et al. [35] utilize trifocal tensor constraints. The recently developed LBA method [16], used in this work, applies two kind of multiview constraints: the two-view and three-view constraints. Pose-SLAM techniques [8, 14] avoid explicit mapping by maintaining the camera trajectory as a sparse graph of relative pose constraints, which are calculated using the landmarks as a separate process.

The target tracking problem, referred more generally as *detection and tracking of moving objects* (DTMO) [41] in the robotics literature, has been extensively studied for several decades [1, 13]. The combined SLAM and DTMO problem, which is assessed in our work, can be regarded as an optimization process in which the inputs are the same as in SLAM (e.g. visual observations, environment scans, etc.), but the output includes the map of the environment, robot poses and the state of the observed dynamic

objects. This problem has attracted considerable attention in the recent years, mostly in order to improve SLAM accuracy, which can be greatly degraded by the presence of dynamic objects in the environment, if the latter are considered as static [28]. The first mathematical framework to the combined process of simultaneous localization, mapping and moving object tracking (SLAMMOT) was presented by Wang [40], who decompose the problem into two separate estimators, one for the SLAM problem given the static landmarks, and another for the tracking problem. Occupancy grid-based approaches were proposed later by Vu et al. [38] and Vu [37], who solved SLAM by calculating the maximum likelihood of occupancy grid maps. Ortega [31] introduced a geometric and probabilistic approach to the vision-based SLAMMOT problem, providing a comparison between the different kinds of optimization methods while Hahnel et al. [10] used sampled-based joint probabilistic data association filter (JPDAF) to track people, and occupancy grids for static landmarks.

An extensive overview of the literature concerning SLAM and DTMO is presented by Pancham et al. [32].

1.2 Contribution

This work presents a computationally efficient approach for simultaneous camera ego-motion estimation and target tracking, while operating in unknown or uncertain GPS-deprived environments. "Ego-motion" refers to the camera's motion expressed in terms of a relative 6DOF pose, i.e. relative translation and orientation, with respect to a reference frame (e.g. the first camera), while target "tracking" refers to the estimation of its position and velocity. Our focus lies on robotic applications for which on-line 3D structure reconstruction is of no interest, although recovering the latter off-line from optimized camera poses is always possible [19]. We propose to take advantage of the recently developed incremental light bundle adjustment (iLBA) [16, 18, 19] framework, which uses multi-view constraints to algebraically eliminate the (static) 3D points from the optimization, allowing the dynamic target to become the only explicitly reconstructed 3D point in the process. The reduced number of variables involved in the optimization allows therefore for substantial savings in computational efforts.

Throughout this work, we formulate the problem's probability distribution function (pdf), over which we calculate the corresponding maximum a posteriori (MAP) estimate. Incremental smoothing and mapping (iSAM) [20] technique is applied to re-used calculations, allowing to further reduce running time, in a similar fashion to the static-scene oriented iLBA approach [19]. We demonstrate, using simulations on synthetic datasets and real-imagery experiments, that while our methods provides similar levels of accuracy than full BA and target tracking, it compares favorably in terms of computational complexity.

This report is structured as follows: Chapter 2 introduces a few of the technical notions which will be used throughout this report. In Chapter 3, we formulate the

simultaneous ego-motion estimation and moving object tracking problem. Chapter 4 reviews the LBA method [18], which is then extended to address the mentioned problem. Chapter 5 focuses on the optimization process and Chapter 6 describes simulations and experimental results, comparing our method with full BA in terms of processing time and accuracy. We conclude in Chapter 7 and share thoughts about further possible developments.

Chapter 2

Background

2.1 The Pinhole Camera Model

The most common model for perspective camera assumes a pinhole projection system. The model describes the camera aperture as a point, through which the light rays pass, mapping the environment onto the *image plane* (See Figure 2.1).

Let $l_C = [x, y, z]^T$ be a landmark (i.e. a scene point) in the camera reference frame and $\tilde{p} = [u, v, 1]^T$ its homogeneous projection on the image plane (in pixels), the mapping of l_C to the image plane is given by the perspective projection equation [11]

$$p = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K l_C = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.1}$$

where λ is the *depth factor* (i.e. the z coordinate of the image plane in the camera axis frame), α_u and α_v are the *focal lengths* and $[u_0, v_0]$ is the *principal point* (i.e. the optical center). These parameters are called the *intrinsic parameters* and the K matrix, the *calibration matrix*. In a global reference frame, the considered scene point is expressed as $l_C = R l_G + t$ where R is the rotation matrix from global reference frame to camera reference frame, and t is the translation from the global reference frame to the camera reference frame, expressed in the latter. Substituting l_C into Equation 2.1 allows to define the *projection operator* as

$$proj\,(x, l_G) = K[R|t] l_G \tag{2.2}$$

where x represents the camera pose in the global reference frame.

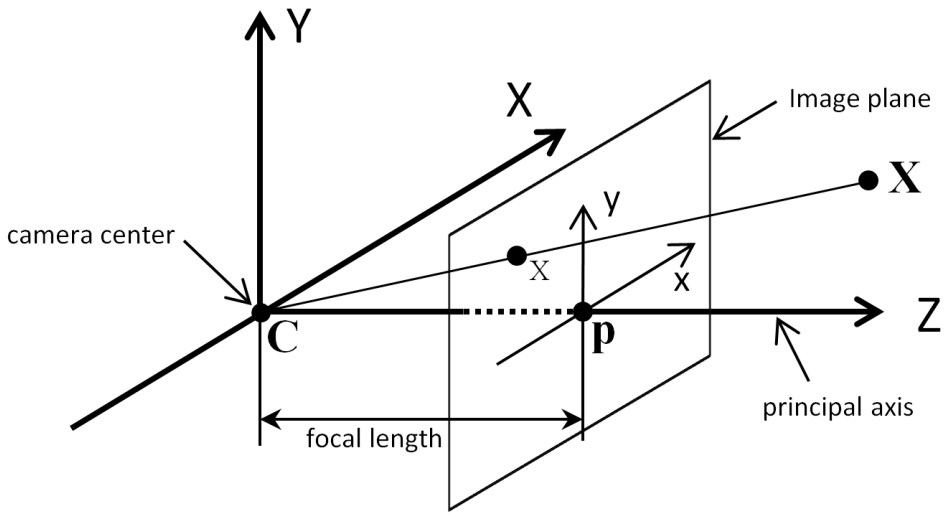


Figure 2.1: Pinhole Camera Geometry

2.2 The Tree-View Constraints [15]

The *three-view constraints* represent geometrical relations between three camera poses from which the same 3D landmark is observed. These constraints will be used in this work as part of the pose estimation process.

Consider three camera poses, x_k, x_l, x_m, from which the same 3D point l_1 is observed. As shown in Figure 2.2, we denote $t_{k\to l}$ and $t_{l\to m}$ the translation vectors from the k^{th} view to the l^{th} view and from the l^{th} view to the m^{th} view, respectively, and define q_k, q_l and q_m as the line of sight (LOS) vectors from the camera to the landmark. The position of the 3D landmark relative to view x_k expressed in some reference axis system G can be written

$$R_G^{C_k} q_k^{C_k} = R_G^{C_k} t_{k\to l}^{C_k} + R_G^{C_l} q_l^{C_l} \tag{2.3}$$
$$R_G^{C_k} q_k^{C_k} = R_G^{C_k} t_{k\to l}^{C_k} + R_G^{C_l} t_{l\to m}^{C_l} + R_G^{C_m} q_m^{C_m}, \tag{2.4}$$

where $R_G^{C_i}$ denotes the rotation matrix from the camera system at x_i to the reference frame. Subtracting Equation 2.4 from Equation 2.3 and re-writing Equation 2.3 yields

$$0 = R_G^{C_k} q_k^{C_k} - R_G^{C_l} q_l^{C_l} - R_G^{C_k} t_{k\to l}^{C_k} \tag{2.5}$$
$$0 = R_G^{C_l} q_l^{C_l} - R_G^{C_m} q_m^{C_m} - R_G^{C_l} t_{l\to m}^{C_l}. \tag{2.6}$$

We define scales parameters λ_i such that $q_i = \lambda_i \hat{q}_i$ where $\hat{q}_i$ is the unit vector in the LOS's direction. Equations 2.5 and 2.6 can then be re-written into the matrix form as

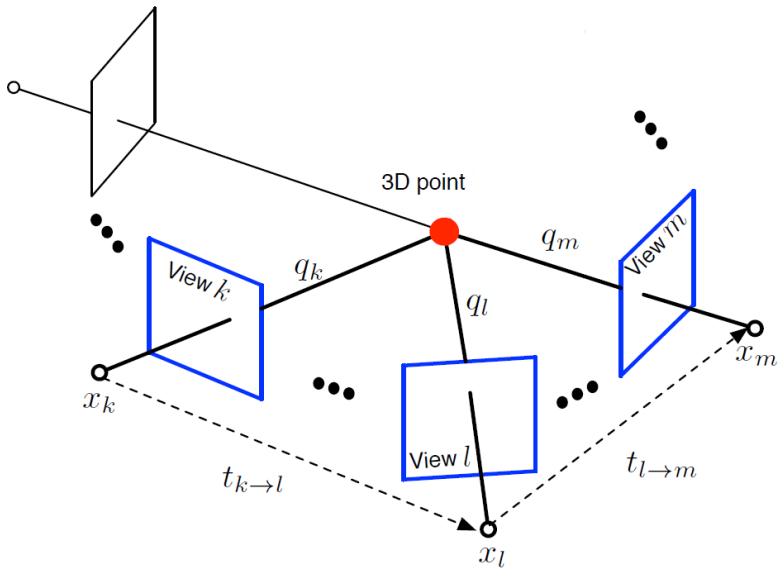


Figure 2.2: Three view geometry for frames k, l and m observing a landmark l_1. Image from [19]

$$\underbrace{\begin{bmatrix} \hat{q}_k & -\hat{q}_l & 0_{3\times 1} & -t_{k\to l} \\ 0_{3\times 1} & \hat{q}_l & -\hat{q}_m & -t_{l\to m} \end{bmatrix}}_{A} \begin{bmatrix} \lambda_k \\ \lambda_l \\ \lambda_m \\ 1 \end{bmatrix} = 0_{6\times 1}, \tag{2.7}$$

where $\hat{q}_i$ are expressed in the reference frame (i.e. $\hat{q}_i = R_G^{C_i} \hat{q}_i^{C_i}$).

Since the elements of $\begin{bmatrix} \lambda_k & \lambda_l & \lambda_m & 1 \end{bmatrix}^T$ are non-zero, it follows that $\text{rank}\,(A) < 4$, which is possible if and only if the following conditions are satisfied [15]

$$q_k \cdot (t_{k\to l} \times q_l) = 0, \tag{2.8}$$
$$q_l \cdot (t_{l\to m} \times q_m) = 0, \tag{2.9}$$
$$(q_l \times q_k) \cdot (q_m \times t_{l\to m}) = (q_k \times t_{k\to l}) \cdot (q_m \times q_l). \tag{2.10}$$

The two first equations are called *two-view constraints* and correspond to the *epipolar constraint*, also reviewed in Section 2.5, which relates between a 3D point and its projection at two different camera positions. Since Equation 2.8 and 2.9 are homogeneous, the translations $t_{k\to l}$ and $t_{l\to m}$ can only be found up to scale. Equation 2.10 connects between the magnitudes of these translations. As a result, given one of the translation scales, the second can be calculated.

2.3 Probabilistic Representation of Estimation Problems

Troughout this work, we use Bayesian inference to recover the posterior probability distribution function over variables of interest such as camera poses, target states and (possibly) landmark locations, based on available information such as features from camera-captured images. In this section, however, we first provide the necessary background in probabilistic inference considering some variable X.

Let X denote some continuous random variable and x a specific value that X may be equal to. Thus, the expression

$$p(X = x), \tag{2.11}$$

abbreviated $p(x)$, denotes the probability that X has the value x and is called the *probability density function* (pdf). Assuming normal Gaussian distribution, as is in most SLAM literature, the pdf of a one dimensional variable (i.e. x is a scalar) with mean μ and variance σ^2 is given by

$$p\left(x\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{\left(x-\mu\right)^2}{\sigma^2}\right), \tag{2.12}$$

and is abbreviated $x \sim N\left(\mu, \sigma^2\right)$. In the case where x is a vector, the normal distribution is called *multivariate* and is given by

$$p\left(x\right) = \frac{1}{\sqrt{\det\left(2\pi\Sigma\right)}} \exp\left(-\frac{1}{2}\left\|x-\mu\right\|_{\Sigma}^2\right), \tag{2.13}$$

where Σ is the covariance matrix and $\left\|x-\mu\right\|_{\Sigma}^2 \doteq \left(x-\mu\right)^T\Sigma^{-1}\left(x-\mu\right)$ is called the *squared Mahalanobis norm.*

Let Y denote a second random variable involved in the problem and y a specific value that Y may take on. The *joint* pdf of variables X and Y is defined $p(x, y)$ and, if the two variables are independent, is given by

$$p(x, y) = p(x)p(y). \tag{2.14}$$

However, variables often carry information about other variables, in which case one probability will be calculated *given* the probability of the other one. In the case of X and Y, the probability that X's value is x given Y's value is y, is written

$$p\left(x|y\right) = \frac{p\left(x, y\right)}{p\left(y\right)}. \tag{2.15}$$

From Equation 2.15, we can formulate a relation between the two conditionals $p(x|y)$ and $p(y|x)$, known as the *Bayes rule*:

$$p\left(x|y\right) = \frac{p\left(y|x\right)p\left(x\right)}{p\left(y\right)}. \tag{2.16}$$

In probabilistic robotics, Bayes rule allows to compute the conditional probability $p(x|y)$ over the variable X given Y, called the *posterior probability distribution*, as a function of the inverse conditional probability $p(y|x)$ and the prior $p(x)$. $p(y)$ being independent of x, it can be considered as constant and thus, Equation 2.16 can be written

$$p(x|y) = \eta p(y|x)p(x), \tag{2.17}$$

where $\eta = p(y)^{-1}$.

Eventually, the goal is to obtain an optimal estimate for the set of unknowns, given available information. Referring to Equation 2.17, this is done by calculating the *maximum a posteriori* (MAP) estimate x^*, given y:

$$x^* = \arg\max_x p\left(x|y\right). \tag{2.18}$$

2.4 Factor Graph

A joint probability distribution function can be represented by a graphical model called *factor graph*, which can be used to perform efficient incremental inference. A factor graph is a bipartite graph which represents a specific factorization of a joint pdf $p(\Theta)$

$$p(\Theta) \propto \prod f_i(\Theta_i). \tag{2.19}$$

Here, Θ_i is the subset of variables Θ involved in the *factor* f_i, which represents the constraint between the involved variables, such as motion models, measurement models and priors. A factor graph is composed of vertices, which picture the variables, and of nodes, representing the factors. An edge between a factor node f_i and a variable vertice $x_j \in \Theta$ exists only if the factor f_i expresses a constraint involving the variable x_j. A simple example is shown in Figure 2.3

Referring to Section 2.3, for the Gaussian case, f_i is then defined

$$f(\Theta) \doteq \exp\left(-\frac{1}{2}||g_i(\Theta_i) - t_i||^2_{\Sigma_i}\right), \tag{2.20}$$

where g is a model function, t a measurement or pose, and $||e||^2_\Sigma = e^T\Sigma^{-1}e$ is the squared Mahalanobis distance with covariance Σ_i.

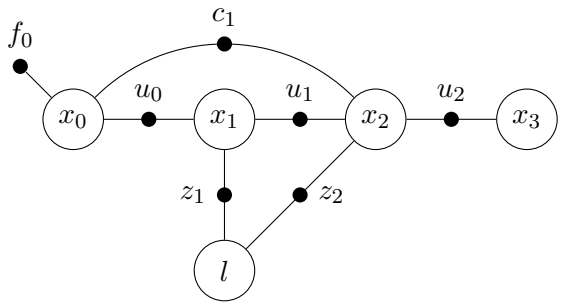


Figure 2.3: Example of factor graph for $f(\Theta) = f(x_0) f(x_0, x_1) f(x_0, x_2) f(x_1, x_2) f(x_1, l) f(x_2, l) f(x_2, x_3)$.

As will be reviewed in Chapter 5, by operating directly on the factor graph, calculations from previous optimization steps can be re-used to improve computational efficiency.

2.5 Essential Matrix and Pose Estimation

Camera poses are estimated as part of a bundle adjustment process, which requires an initial guess as a starting point for the optimization. In lack of supplementary information, this first guess will have to be calculated using the on-board camera measurements. One way is to extract the relative motion between the current and the last estimated poses, using an important property called the *epipolar constraint*. The epipolar constraint is a geometric relation which links between a 3D point and its projection on two different camera positions. Let the *epipolar plane* be the plane defined by the baseline (i.e. the line defined by C_{k-1} and C_k in Figure 2.4) and a 3D point X. The *epipolar lines* are then defined as the intersection of the epipolar plane with the image planes of the two cameras. let $\tilde{p}$ and $\tilde{p}'$ be the normalized projection of X on frame $k-1$ and k respectively. If $\vec{t}$ is denoted as the translation vector from C_{k-1} to C_k, then $\tilde{p}$, $\tilde{p}'$ and $\vec{t}$ are co-planar and therefore:

$$\left[\tilde{p}'\right]^T \cdot \left(t \times \tilde{p}''\right) = 0, \tag{2.21}$$

where $\tilde{p}''$ is the vector corresponding to $\tilde{p}$ in C_k's reference frame and $\tilde{p}'' = R\tilde{p}$, with R as the rotation matrix from C_{k-1} to C_k. From Equation 2.21, we get

$$\left[\tilde{p}'\right]^T [t]_\times R\,\tilde{p} = 0, \tag{2.22}$$

where $[t]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$.

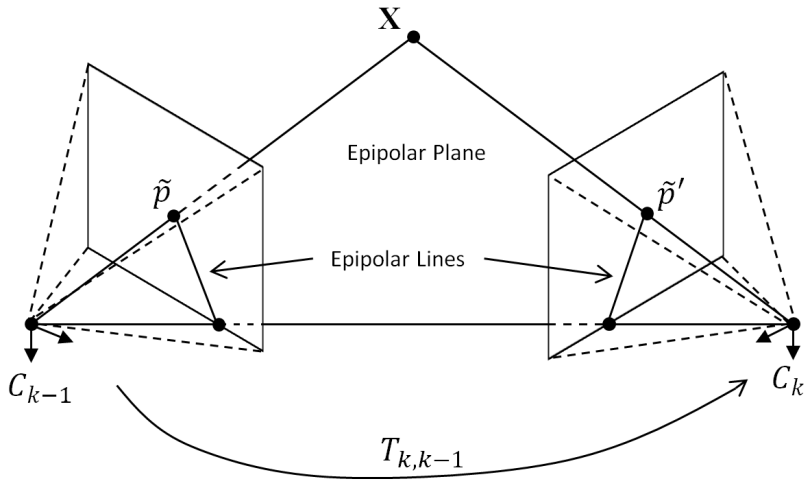


Figure 2.4: The *epipolar plane* is the plane defined by the baseline (i.e. the line defined by C_{k-1} and C_k) and a 3D point X. The *epipolar lines* are defined as the intersection of the epipolar plane with the image planes of the two cameras. $\tilde{p}$ and $\tilde{p}'$ are the normalized projection of X on frame $k-1$ and k respectively. If $\vec{t}$ is denoted as the translation vector from C_{k-1} to C_k, then $\tilde{p}$, $\tilde{p}'$ and $\vec{t}$ are co-planar. This is known as the *epipolar constraint*

$$E \simeq [t]_{\times} R \tag{2.23}$$

is called the *essential matrix*. Here, the symbol $\simeq$ denotes the fact that the equivalence is valid up to a multiplicative scalar. Incorporating the latter in Equation 2.22, we get

$$\left[\tilde{p}'\right]^T E\, \tilde{p} = 0. \tag{2.24}$$

This homogeneous equation defines the *epipolar constraint*.

Given data-association, or as part of a feature-matching algorithm such as RANSAC [9], the essential matrix resulting from Equation 2.24 can be estimated. Referring to Equation 2.23, the rotation and translation parts can be extracted using singular value decomposition (SVD). A valid essential matrix after SVD is $E = USV^T$. In general, there are four different solutions R, t for one essential matrix:

$$R = U\left(\pm W^T\right) V^T \tag{2.25}$$

$$\hat{t} = U\left(\pm W\right) SU^T \tag{2.26}$$

where $W^T = \begin{bmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. In order to identify the correct R, t pair, a single point is triangulated. The resulting z component of the reconstructed point must be positive

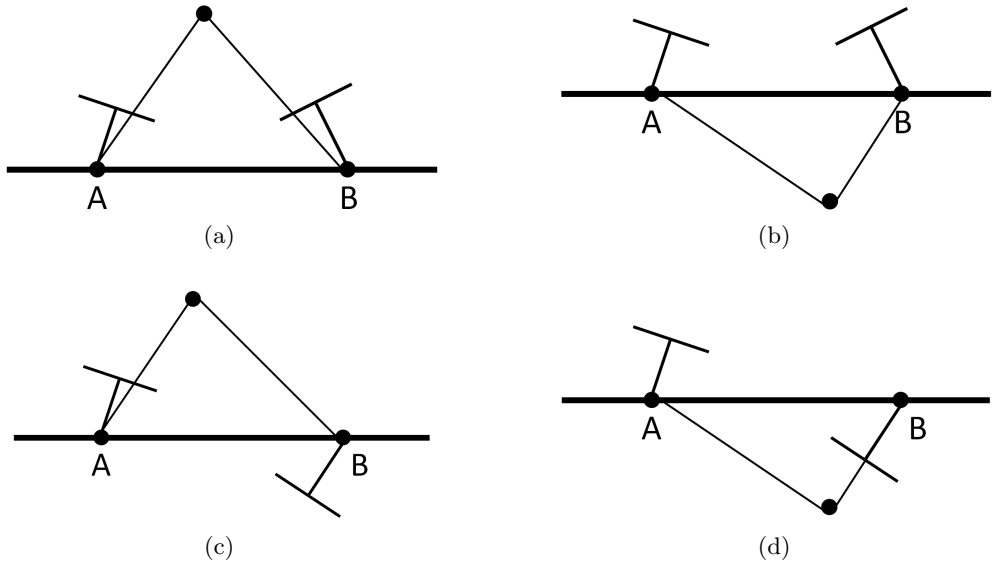


Figure 2.5: The four solutions to the R, t extraction. (a) is the only possible solution, where the landmark stands in front of both cameras

in the two camera frames (See Figure 2.5).

Chapter 3

Problem Formulation and Notations

We consider a scenario where a monocular camera mounted on a mobile robot is tracking a dynamic target while operating in a GPS-deprived unknown environment.

3.1 The Bundle Adjustment Problem

The process of determining the camera poses and the stationary 3D structure given measurements is called *bundle adjustment* (BA), or *simultaneous localization and mapping* (SLAM). Let x_k represent the camera pose (i.e. 6DOF position and orientation) at time-step t_k, and denote all such states up to that time by $X_k \doteq \begin{Bmatrix} x_0 & \dots & x_k \end{Bmatrix}$. We also use $L_k \doteq \begin{Bmatrix} l_1 & \dots & l_n \end{Bmatrix}$ and $Z_k \doteq \begin{Bmatrix} z_0 & \dots & z_k \end{Bmatrix}$ to represent, respectively, all the landmarks observed by time t_k, and the corresponding sensor observations. Here, for each time index $i \in [0, k]$, z_i corresponds to all image observations obtained at time t_i. In particular, we use the notation z_i^j to denote an observation of the jth landmark at time t_i.

Using probabilistic representation, the BA problem can be expressed by the joint pdf

$$P\left(X_k, L_k | Z_k\right). \tag{3.1}$$

Using Bayes' rule, the general recursive Bayesian formula for bundle adjustment can be derived as (See Section 2.3 for more details)

$$P\left(X_k, L_k | Z_k\right) \propto priors \cdot \prod_{i=1}^{k} \prod_{j \in \mathcal{M}_i} p\left(z_i^j | x_i, l_j\right), \tag{3.2}$$

where $\mathcal{M}_i$ is the set of landmarks observed at time index i and $priors$ represent prior

information on the estimated variables.

Considering a standard pinhole camera, the corresponding observation model can be defined as (See Section 2.1)

$$z_i^j = proj\left(x_i, l_j\right) + v_{ij}, \tag{3.3}$$

where $proj\left(\cdot\right)$ is the projection operator [11] and $v_{ij} \sim \mathcal{N}\left(0, \Sigma_v\right)$ is a zero-mean white noise with measurement covariance Σ_v. Under Gaussian distribution assumption, the measurement likelihood of the perception measurement can be expressed as

$$p\left(z|x,l\right) \doteq \frac{1}{\sqrt{|2\pi\Sigma_v|}} \exp\left(-\frac{1}{2}\left\|z - proj\left(x,l\right)\right\|_{\Sigma_v}^2\right). \tag{3.4}$$

We assume camera calibration is known; otherwise, the uncertain calibration parameters could be incorporated into the optimization framework as well.

Solving the bundle adjustment problem would therefore consist in calculating the maximum a posteriori estimate over the joint pdf, defined as

$$X_k^*, L_k^* = \underset{X_k, L_k}{\arg\max}\, P\left(X_k, L_k|Z_k\right). \tag{3.5}$$

Due to the monotonic characteristics of the logarithmic function, calculating the MAP estimate X_k^*, L_k^* becomes equivalent to minimizing the negative log-likelihood of the BA pdf 3.1

$$X_k^*, L_k^* = \underset{X_k, L_k}{\arg\min} - \log P\left(X_k, L_k|Z_k\right). \tag{3.6}$$

This leads to a nonlinear least-squares optimization, where the cost function

$$J_{BA}\left(X_k, L_k\right) = \sum_i \sum_{j \in \mathcal{M}_i} \left\| z_i^j - proj\left(x_i, l_j\right)\right\|_{\Sigma}^2 \tag{3.7}$$

is to be minimized. Note that, to avoid clutter, the prior terms are not explicitly shown in Equation 3.7.

3.2 Bundle Adjustment and Target Tracking

We investigate scenarios in which a dynamic target is tracked by the camera. Based on the camera observations of the target, we seek to estimate its trajectory and velocity over time. We assume the target moves randomly, however, its motion is assumed to follow a known stochastic kinematic model (e.g. constant velocity or constant acceleration).

Let y_k represent the target state at time step t_k, defined generally as

$$y_k \doteq \left[y_{T_k}\ d_{T_k}\right]^T = \left[x_{T_k}, y_{T_k}, z_{T_k}, \dot{x}_{T_k}, \dot{y}_{T_k}, \dot{z}_{T_k}, ...\right]^T, \tag{3.8}$$

where y_{T_k} denotes the target's tri-dimensional position and d_{T_k} its higher order time derivatives required to accommodate the assumed motion model. In the frame of this work, we focus on the target's position and velocity. y_k is therefore a six element vector defined as

$$y_k = \begin{bmatrix} y_{T_k} \\ \dot{y}_{T_k} \end{bmatrix} \in \mathbb{R}^{6\times 1}. \tag{3.9}$$

We denote $Y_k \doteq \begin{Bmatrix} y_0 & \dots & y_k \end{Bmatrix}$ the set of all target's states up to time-step t_k.

Assuming a known Markovian motion model for the target, which likelihood is represented by $p\left(y_i|y_{i-1}\right)$, we define a joint pdf for the random variables involved in the considered problem, given all information thus far, as

$$P\left(X_k, Y_k, L_k|Z_k\right) \propto priors \cdot \prod_{i=1}^{k} \left(p\left(y_i|y_{i-1}\right) p\left(z_i^{y_i}|x_i, y_i\right) \prod_{j\in\mathcal{M}_i} p\left(z_i^j|x_i, l_j\right) \right), \tag{3.10}$$

where $z_i^{y_i}$ denotes the observation of the target by the ith camera and $p\left(z_i^{y_i}|x_i, y_i\right)$ refers to a similar observation model than the one discussed in Section 3.1. $\mathcal{M}_i$ is the set of landmarks observed at time index i and we consider $priors = p\left(x_0\right) p\left(y_0\right)$ as given information.

In this work, as in many robotics applications, we consider a constant velocity model [2], characterized by the equation

$$\ddot{y}\left(t\right) = \widetilde{w}\left(t\right), \tag{3.11}$$

where $\tilde{w}\left(t\right)$ is a continuous time zero-mean white noise representing the slight velocity changes from its actual value.

The target state linear continuous propagation is generally noted as $\dot{y}\left(t\right) = Ay\left(t\right) + Dw\left(t\right)$, where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, or under its discrete form:

$$y_{k+1} = \Phi_k y_k + G_k w_k, \tag{3.12}$$

where G_k is the process noise Jacobian defined as $G_k = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \mathbb{R}^{6\times 3}$ and Φ_k is the state transition matrix and is defined as $\Phi_k = \begin{bmatrix} 1 & \triangle t \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{6\times 6}$ with $\triangle t \doteq t_{k+1} - t_k$. The discrete-time process noise $w_k \sim \mathcal{N}\left(0, \Sigma_w\right)$ relates to the continuous-time one as $w_k = \int_0^{\Delta t} e^{A(\Delta t-\tau)} D\tilde{w}\left(k\Delta t + \tau\right) d\tau$. Under Gaussian distribution assumption, the

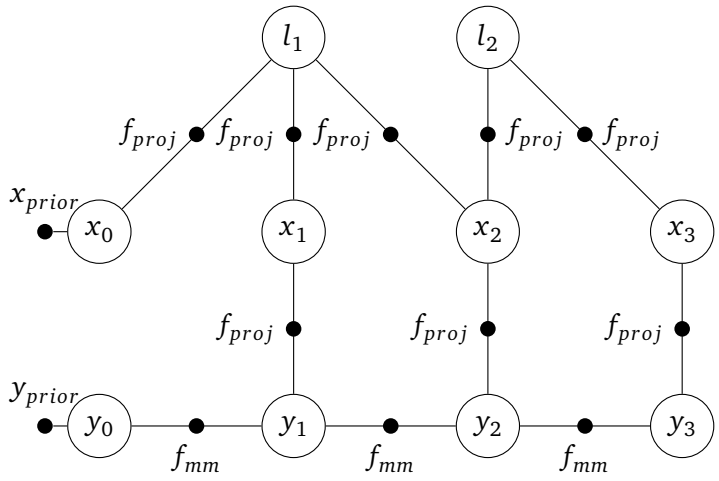


Figure 3.1: Factor graph representing a factorization of the joint pdf for bundle adjustment with single target tracking

motion model likelihood is therefore expressed

$$p\left(y_{k+1}|y_k\right) \doteq \frac{1}{\sqrt{|2\pi\Sigma_{mm}|}} \exp\left(-\frac{1}{2}\left\|y_{k+1} - \Phi_k y_k\right\|^2_{\Sigma_{mm}}\right), \tag{3.13}$$

where $\Sigma_{mm} \doteq G\Sigma_w G^T$.

Finally, solving the combined bundle adjustment and target state estimation process consists in calculating the MAP estimate over the joint pdf from Equation 3.10

$$X_k^*, Y_k^*, L_k^* = \underset{X_k, Y_k, L_k}{\arg\max}\, P\left(X_k, Y_k, L_k|Z_k\right) \tag{3.14}$$

3.3 Factor Graph Representation

As mentioned in Section 2.4, the factorization of the joint pdf described in Equation 3.10 can be represented using a factor graph [24], which will be used later to efficiently solve the optimization problem using incremental inference (see Chapter 5) . Using the same observation (Equation 3.3) and motion (Equation 3.12) models, this pdf is expressed in factor graph notation as

$$P\left(X_k, Y_k, L_k|Z_k\right) \propto priors \cdot \prod_{i=1}^{k}\left(f_{mm}\left(y_i, y_{i-1}\right) f_{proj}\left(x_i, y_i\right) \prod_{j\in\mathcal{M}_i} f_{proj}\left(x_i, l_j\right)\right). \tag{3.15}$$

An illustration expressing the above factorization for a small example is shown in Figure 3.1. The corresponding factors in Equation 3.15 are straightforwardly defined as follows: The factor $f_{mm}\left(y_i, y_{i-1}\right)$ corresponds to the target motion model and, referring to

Equations 3.12 and 3.13, is defined as

$$f_{mm}\left(y_i, y_{i-1}\right) \doteq \exp\left(-\frac{1}{2}\left\|y_i - \Phi_{i-1} y_{i-1}\right\|^2_{\Sigma_{mm}}\right). \tag{3.16}$$

The projection factors $f_{proj}\left(x_i, l_j\right)$ and $f_{proj}\left(x_i, y_i\right)$ correspond to the landmarks and target observation models; these factor are defined respectively as

$$f_{proj}\left(x_i, l_j\right) \doteq \exp\left(-\frac{1}{2}\left\|z_i^j - proj\left(x_i, l_j\right)\right\|^2_{\Sigma_v}\right) \tag{3.17}$$

and

$$f_{proj}\left(x_i, y_i\right) \doteq \exp\left(-\frac{1}{2}\left\|z_i^{y_i} - proj\left(x_i, y_i\right)\right\|^2_{\Sigma_v}\right). \tag{3.18}$$

Similarly to the previous section, the MAP estimate is defined as

$$X_k^*, Y_k^*, L_k^* = \underset{X_k, Y_k, L_k}{\arg\max}\, P\left(X_k, Y_k, L_k | Z_k\right), \tag{3.19}$$

and can be efficiently calculated by exploiting the inherent sparse structure of the problem while re-using calculations, as explained in Chapter 5.

This corresponds to the state of the art where inference is performed over camera poses, landmarks and target states. Yet, when the primary focus is navigation rather than mapping, explicit estimation of the observed landmarks in an on-line process is not actually required. Conceptually, estimating only the camera poses and the dynamic target (but not the landmarks) involves less variables to optimize and could be attractive from a computational point of view. In this work, we develop an approach based on this idea.

Chapter 4

LBA and Dynamic Target Tracking

Bundle adjustment is a nonlinear iterative optimization framework typically applied for estimating camera poses and observed landmarks. In this chapter, we integrate target tracking to a structure-less bundle adjustment technique called Light Bundle Adjustment (LBA) [18]. In the first section, we formulate the LBA equations while considering a static scene. These equations are then extended in Section 4.2 to incorporate the dynamic target tracking problem.

Using the factor graph notations from Chapter 3, the joint pdf $P(X_k, L_k|Z_k)$ which corresponds to the static problem can be factorized, similarly to Equation 3.15, as

$$P(X_k, L_k|Z_k) \propto priors \cdot \prod_{i=1}^{k} \left(\prod_{j \in \mathcal{M}_i} f_{proj}(x_i, l_j) \right), \tag{4.1}$$

where $priors = p(x_0)\, p(y_0)$ represents the prior information on the camera and target states.

As mentioned, this works considers robotics applications for which the on-line reconstruction of the 3D structure is of no interest. One way to avoid explicit estimation of the landmarks in the solution is by marginalizing out the latter from the joint pdf as in

$$P(X_k|Z_k) = \int P(X_k, L_k|Z_k)\, dL_k. \tag{4.2}$$

However, this involves a series of calculations which, in the case of on-line operation, could be penalizing: First, performing the exact marginalization would initially require the optimization of the full bundle adjustment problem, including landmarks, before applying a Gaussian approximation to compute the marginal. Secondly, marginalization in the information form involves expensive calculation of the Schur complement over the variables we wish to keep [8]. Moreover, marginalization introduces fill-in, destroying

the sparsity of the information matrix.

In contrast, structure-less BA methods approximate the BA cost function, allowing for estimation of the camera poses without involving the reconstruction of the 3D structure [33, 35]. In this work, we use the recently developed light bundle adjustment (LBA) approach [19, 16], which algebraically eliminates the landmarks from the optimization, using multi-view constraints and in particular, three-view constraints.

4.1 Light Bundle Adjustment (LBA)

LBA allows for reduction of the number of variables involved in the optimization compared to standard bundle adjustment. By algebraically eliminating the landmarks from the problem, the optimization can be performed over the camera poses only. The key idea is to use geometrical constraints relating three views from which the same landmark is observed.

As reviewed in Section 2.2, considering a set of three different poses from which a common landmark is observed (See Figure 2.2), it is possible to derive constraints that relate the three poses while eliminating the landmark [17]. These constraints can be formulated as two two-view constraints g_{2v} between the two pairs of poses and one three-view constraint g_{3v} between the three involved poses [15, 17]. Conceptually, the two-view constraint is equivalent to the epipolar constraint (See Section 2.5), while the three-view constraint relates between the scales of the two translations $t_{k\to l}$ and $t_{l\to m}$ (See Figure 2.2). Writing down the appropriate projection equations, we get

$$g_{2v}\left(x_k, x_l, z_k, z_l\right) = q_k \cdot \left(t_{k\to l} \times q_l\right) \tag{4.3}$$

$$g_{2v}\left(x_l, x_m, z_l, z_m\right) = q_l \cdot \left(t_{l\to m} \times q_m\right) \tag{4.4}$$

$$g_{3v}\left(x_k, x_l, x_m, z_k, z_l, z_m\right) = \tag{4.5}$$
$$\left(q_l \times q_k\right) \cdot \left(q_m \times t_{l\to m}\right) - \left(q_k \times t_{k\to l}\right) \cdot \left(q_m \times q_l\right),$$

where k, l and m are the three overlapping poses. $q_i \doteq R_i^T K_i^{-1} z$ for the i^{th} view and image observation z, where K_i is the calibration matrix, R_i represents the rotation matrix from some reference frame to the i^{th} view, and $t_{i\to j}$ denotes the translation vector from view i to view j, expressed in the global frame.

The resulting probability distribution $P_{LBA}\left(X|Z\right)$ can thus be factorized as

$$P_{LBA}\left(X|Z\right) \propto \prod_{i=1}^{N_h} f_{2v/3v}\left(X_i\right), \tag{4.6}$$

where $f_{2v/3v}$ represents the involved two- and three-view factors and X_i is the relevant subset of camera poses. Referring to Equations 4.3-4.5, under Gaussian distribution

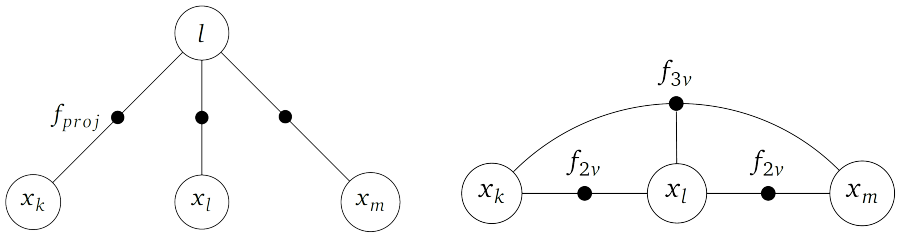


Figure 4.1: Factor graph representation for a small example including three views x_k, x_l, x_m. (a) represents the BA problem, where the three views are related to the landmark l with projection factors. (b) represents the LBA problem, where the landmark l has been eliminated, and the three views are related by two- and three-view constraints

assumption, f_{2v} and f_{3v} are defined as:

$$f_{2v}\left(x_k, x_l\right) \doteq \exp\left(-\frac{1}{2}\left\|g_{2v}\left(x_k, x_l, z_k, z_l\right)\right\|^2_{\Sigma_{2v}}\right) \tag{4.7}$$

and

$$f_{3v}\left(x_k, x_l, x_m\right) \doteq \exp\left(-\frac{1}{2}\left\|g_{3v}\left(x_k, x_l, x_m, z_k, z_l, z_m\right)\right\|^2_{\Sigma_{3v}}\right), \tag{4.8}$$

which correspond to the likelihoods of the two- and three-views constraints involving x_k and x_l in Equation 4.7 and involving x_k, x_l and x_m in Equation 4.8. The covariances Σ_{2v} and Σ_{3v} are defined as:

$$\Sigma_{2v} \doteq \left(\nabla_{z_k, z_l} g_{2v}\right) \Sigma \left(\nabla_{z_k, z_l} g_{2v}\right)^T, \;\; \Sigma_{3v} \doteq \left(\nabla_{z_k, z_l, z_m} g_{3v}\right) \Sigma \left(\nabla_{z_k, z_l, z_m} g_{3v}\right)^T. \tag{4.9}$$

Figure 4.1 shows a comparison between the factor graph representation of LBA and standard BA for a small example.

Therefore, rather than optimizing the cost function 3.7, that involves the camera and landmark states, the optimization is performed on the cost function [19]

$$J_{LBA}(X) \doteq \sum_{i=1}^{N_h} \left\|h_i(X_i, Z_i)\right\|^2_{\Sigma_i}, \tag{4.10}$$

where $h_i \in \{g_{2v}, g_{3v}\}$ represents a single two- or three-view constraint involving the set of poses X_i and the set of image observations Z_i, N_h being the number of resulting constraints.

Practically, when a landmark is observed by a new view x_k and some earlier views x_l and x_m, a single two-view (between x_k and one of the two other views) and a single three-view constraint are added (between the three views). The reason for not adding the second two-view constraint (between views x_l and x_m) is that this constraint was already added when processing these past views. In case a landmark is observed by

only two views, we add a single two-view constraint.

4.2 LBA and Dynamic Target Tracking

In this section we integrate dynamic target tracking into the LBA framework. As will be shown in Chapter 6, the resulting approach provides comparable accuracy for both target tracking and camera pose estimation while significantly reducing running time, compared to an equivalent BA approach.

The idea behind the proposed method is to incorporate the target tracking problem into the LBA framework in order to yield a proxy for the joint pdf $P(X_k, Y_k|Z_k)$ which involves significantly less variables than the joint pdf $P(X_k, Y_k, L_k|Z_k)$, while somewhat avoiding the expensive calculations involved in the marginalization process [19]. Indeed, if $X_k \in \mathbb{R}^{M_k \times 1}$, $Y_k \in \mathbb{R}^{N_k \times 1}$ and $L_k \in \mathbb{R}^{O_k \times 1}$, then the amount of variables involved in the optimization is decreased form $M_k + N_k + O_k$ to $M_k + N_k$ only, which would reduce computational complexity (We note that $O_k \gg M_k$ and $O_k \gg N_k$).

We integrate the factors $f_{2v/3v}$ corresponding to the camera poses described in Equations 4.7 and 4.8 with the target tracking related factors f_{mm} and f_{proj} defined in Equations 3.16 and 3.18 to yield the joint pdf $P(X_k, Y_k|Z_k)$ over the relevant states only. The target becomes therefore the only 3D point to be estimated in the process:

$$P(X_k, Y_k|Z_k) \propto priors \cdot \prod_{i=1}^{k-1} \left(f_{mm}(y_i, y_{i-1}) f_{proj}(x_i, y_i) \prod_{j=1}^{N} f_{2v/3v}(X_j) \right), \quad (4.11)$$

where, similarly to Equation 3.15, $priors = p(x_0)\, p(y_0)$ represents the prior information and X_j is the relevant subset of views for the i_{th} frame. An illustration expressing the above factorization for the same example as in Figure 3.1 is shown in Figure 4.2.

Solving the localization and target tracking problem then corresponds to estimating the MAP

$$X_k^*, Y_k^* = \arg\max_{X_k, Y_k} P(X_k, Y_k|Z_k), \quad (4.12)$$

which is equivalent to minimizing the cost function

$$\begin{aligned} J(X_k, Y_k) &= \|x_0 - \hat{x}_0\|^2_{\Sigma_x} + \|y_0 - \hat{y}_0\|^2_{\Sigma_y} + \\ &+ \sum_{i=1}^{k} \left(\|y_i - \Phi_i y_{i-1}\|^2_{\Sigma_{mm}} + \|z_i^{y_i} - proj(x_i, y_i)\|^2_{\Sigma_v} + \sum_{j}^{N_h} \|h_j(X_j, Z_j)\|^2_{\Sigma_j} \right). \end{aligned} \quad (4.13)$$

Solving the above mentioned non-linear least square problem is achievable using several optimization methods. In the next chapter, we present the methods used in this work to perform this task efficiently.

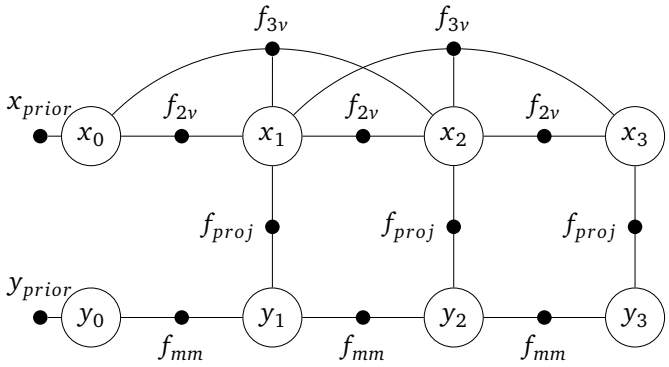


Figure 4.2: Factor graph representing a factorization of the joint pdf for LBA and target tracking

Chapter 5

Incremental Inference

Estimating the MAP $X_k^*, Y_k^* = \arg\max_{X_k,Y_k} P\left(X_k, Y_k | Z_k\right)$ involves solving a non-linear least square problem in which the correspondent cost function $J\left(X_k, Y_k\right)$ is minimized. This can be achieved using different types of optimization methods. While standard BA technique often require trust-region optimization methods such as Levenberg-Marquardt or Dogleg, LBA has been shown to converge using Gauss-Newton [19], a non trust-region method. This allows for additional improvement of the processing time. Gauss-Newton, which we review in the next section, will therefore be the method of choice in the frame of this work. Computational complexity can be further reduced using incremental techniques, which allow to re-use calculations from previous steps. Such a technique is described in Section 5.2.

5.1 Gauss-Newton Optimization

The Gauss-Newton algorithm is an iterative process allowing to find the variables for which a specific cost function is minimized. Considering the above mentioned cost function $J\left(X_k, Y_k\right)$, it is performed as follows:

First, the cost function is linearized using the first-order Taylor expansion:

$$J\left(\bar{X} + \Delta X, \bar{Y} + \Delta Y\right) = \|\bar{x}_0 - \hat{x}_0 + \Delta x_0\|^2_{\Sigma_x} + \|\bar{y}_0 - \hat{y}_0 + \Delta y_0\|^2_{\Sigma_y} + \sum_{i=1}^{k} \Bigg(\|\bar{y}_i - \Phi_i \bar{y}_{i-1} + \Delta\hat{y}_i - \Phi_i \Delta\hat{y}_{i-1}\|^2_{\Sigma_{mm}} + \|z_i^{y_i} - proj\left(\bar{x}_i, \bar{y}_i\right) - \nabla_{x_i} proj \cdot \Delta\hat{x}_i - - \nabla_{y_i} proj \cdot \Delta\hat{y}_i\|^2_{\Sigma_v} + \sum_{j}^{N_h} \left\| h_j\left(\bar{X}_j, Z_j\right) + \nabla_{X_j} h_j \cdot \Delta X_j \right\|^2_{\Sigma_j} \Bigg), \quad (5.1)$$

where $\nabla_q f = \frac{\partial f}{\partial q}|_{\bar{q}}$. Using the identity $\|f\|^2_{\Sigma} \doteq \left\|\Sigma^{-\frac{1}{2}} f\right\|^2$ for a function f with covariance

Σ, equation 5.1 is re-organized to yield the following representation:

$$J\left(\overline{\Theta} + \triangle\Theta\right) \approx \|A\triangle\Theta - b\|^2, \tag{5.2}$$

where $\Theta \doteq \{X, Y\}$. A is called the *Jacobian Matrix* and comprises the jacobians of all the involved functions with respect to the involved variables, while b is called the *residuals*.

Next, The optimal increment $\Delta\Theta$ is calculated by solving the linear equation

$$A\triangle\Theta = b, \tag{5.3}$$

and the linearization point is updated for all variables

$$\overline{\Theta} + \triangle\Theta \rightarrow \overline{\Theta}. \tag{5.4}$$

This process is then repeated until convergence.

Still, several operations in the Gauss-Newton process are potentially demanding in terms of computational efforts. First, the solution of the linear Equation 5.3 is not a trivial task. A naive approach would process by calculating

$$\triangle\Theta = \left(A^T A\right)^{-1} A^T b. \tag{5.5}$$

As said, A includes the jacobians of all the involved functions with respect to all the involved variables, and thus, A can grow to be very large. As a consequence, the calculation of the inversed *information matrix* $(A^T A)^{-1}$ can become an expensive procedure. Futhermore, the Gauss-Newton process is usually done using batch optimization, which re-performs the optimization process from scratch at every time-step (or batch of steps) and can be penalizing in the case of on-line operation.

A better alternative is thus to use Incremental Smoothing And Mapping (iSAM), a recently developed algorithm ([20]) that exploits the sparsity of the involved matrices while re-using calculations from previous time steps in order to reduce computational complexity. This method is reviewed in the next section.

5.2 Incremental Smoothing

Incremental smoothing and mapping (iSAM) [5, 21, 20] allows to efficiently handle the optimization process described in the previous section. The iSAM method is applied in the framework of our simulations and experiments in order to speed up calculations, regardless of the bundle adjustment technique in use. The method is therefore reviewed here for the sake of self-completeness.

First, the SAM method exploits the sparsity of the Jacobian Matrix by resorting to matrix factorization in order to simplify the linear Equation 5.3 (Matrix factorization is

a cost-effective procedure only when the involved matrix is sparse). Matrix factorization is usually performed using QR or Cholesky method, both yield an upper-triangular matrix R called the *square-root information matrix* which replaces the Jacobian matrix A in Equation 5.3. Using QR method:

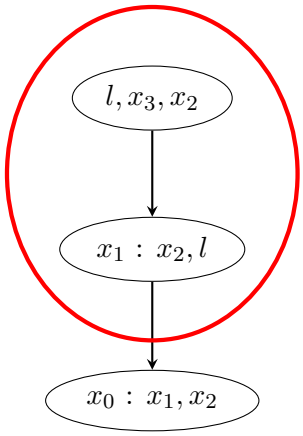
$$||A\Delta\Theta - b||^2 = \left|\left|Q\begin{bmatrix}R\\0\end{bmatrix}\Delta\Theta - b\right|\right|^2 = \left|\left|Q^TQ\begin{bmatrix}R\\0\end{bmatrix}\Delta\Theta - Q^Tb\right|\right|^2 =$$
$$= \left|\left|\begin{bmatrix}R\\0\end{bmatrix}\Delta\Theta - \begin{bmatrix}d\\e\end{bmatrix}\right|\right|^2 = ||R\Delta\Theta - d||^2 + ||e||^2 \quad (5.6)$$

where Q is an orthogonal matrix . $||A\Delta - b||^2$ is thus minimal if $R\Delta\Theta = d$, leaving $||e||^2$ as the residual. $\Delta\Theta$ is then obtained by back-substitution.

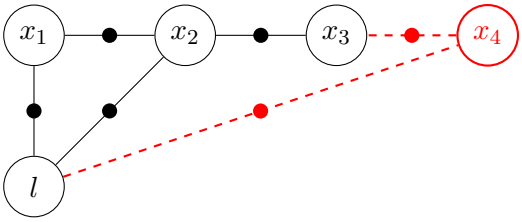
Still, matrix factorization remains the main task performed at every optimization step. Batch optimization performs this from scratch each time new variables are added to the problem. In contrast, incremental smoothing updates the problem as new measurements and variables arrive, by directly updating the square root information matrix R and by recalculating only the matrix entries that actually change. A key insight is the fact that factorizing the measurement Jacobian matrix A into a sparse square root information matrix R is equivalent to eliminating the corresponding factor graph into a *Bayes net* with a certain variables elimination order. Having that in mind, updates on the matrix R can be performed by directly updating the corresponding Bayes net. For this exact purpose, the *Bayes tree* is introduced, a directed tree in which the nodes represent *cliques* of the underlying Bayes net. As new variables and factors are added, the SAM problem is updated directly by changing the affected part of the Bayes tree only, thus avoiding the need to re-factorize the Jacobian Matrix.

Bayes trees are graphical models that represent probability densities. Each of the k cliques C_k represents a set of fully connected variables in the Bayes net and is assigned a conditional probability. A Bayes tree is constructed from a Bayes net by discovering its cliques using the maximum cardinality search algorithm [36]. For a new factor $f(x_i, x_j)$, only paths between the cliques containing either x_i or x_j and the root are affected. During the inference step, the affected part of the Bayes tree is converted back into a factor graph, which can be re-eliminated into a Bayes net after the addition of $f(x_i, x_j)$. The Bayes net is then re-converted into a Bayes tree, which can be re-attached to the unaffected sub-trees of the original Bayes tree. An example of this process is described in Figure 5.1.

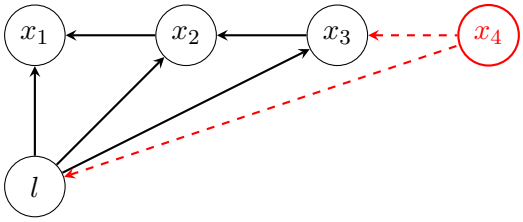
Additionally, using a Bayes tree eliminates the need for batch reordering. Instead, affected variables can be re-ordered continuously at every incremental update (see 5.1c), keeping sparsity at a relatively constant level. Although this is not the optimal solution in terms of global variables ordering, as only the variables affected by the update are being re-ordered, it has provided considerably better results than the periodic batch



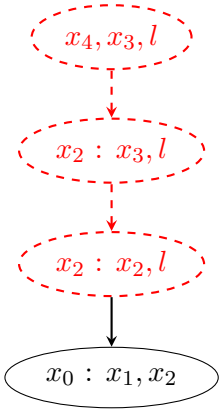
(a) The affected cliques are detected in the original Bayes tree.



(b) Affected cliques are extracted from the Bayes tree to recreate their corresponding factor graph, to which the new factors $f(x_4, x_3), f(x_4, l)$ and the new variable x_4 are added.



(c) The updated factor graph is then re-eliminated with a new elimination order x_1, x_2, x_3, l, x_4 to create a Bayes net.



(d) The Bayes net is then converted into a Bayes tree which is re-attached to the unaffected part of the previous Bayes tree.

Figure 5.1: Update of a Bayes tree with new variable x_4 and factors $f(x_3, x_4)$ and $f(x_4, l)$.

re-ordering. Furthermore, rather than fully re-linearizing the whole set of variables at heuristically determined points in time, *fluid re-linearization* triggers re-linearization of a variable only when the deviation between its current estimate and the linearization point is larger than a defined threshold β, set heuristically or as part of a "tuning process". The same idea is used for the state update at the back-substitution step. Because new variables and factors generally impact their direct surroundings, only a limited part of the system's variables needs to be updated. Therefore, iSAM keeps track of the previous values of the Δ vector, stopping the update process whenever a clique is encountered that refers to a variable for which Δ changes by less than a defined threshold α.

Chapter 6

Results

We demonstrate the benefits of the proposed method with simulations performed on synthetic datasets and with real-imagery experiments. For each scenario, target tracking and ego-pose estimation using LBA and full BA are compared in terms of accuracy and processing time. All experiments were run on an Intel i7-4720HQ quad core processor with 2.6 GHz clock rate and 8GB of RAM. The methods used for comparison were implemented using the GTSAM library[1].

6.1 Experimental Evaluation with Synthetic Datasets

A series of simulations were performed on synthetic datasets in order to compare our method with full BA technique and to demonstrate its capability in terms of computational performance and estimation accuracy for both camera and target states. We present two types of studies: A statistical performance study on an approximately 3km long aerial scenario (Figure 6.1), and a case study in a larger aerial scenario (Figure 6.3). In both cases, the downward-facing camera operates in GPS-denied environments and occasionally re-visits previously explored locations, providing occasional loop-closure measurements. The simulated target takes a similar course on the ground and for the sake of simplicity, stays in the camera's field of view throughout the process. The priors $p(x_0)$ and $p(y_0)$ are Gaussians with means equal to their initial values, and with $\sigma = 2$ [m] standard deviation. The measurement model assumes an image noise $\sigma = 0.5$ [pix]. The continuous-time system is discretized with time-step $\triangle t = 3$ [sec]. Regarding target motion, we use the constant velocity model described in Section 3.2 and assume a zero-mean, white Gaussian noise $\sigma = [30, 30, 0.001]^T$ [$^{\mathrm{m}}/_{\mathrm{sec}}$]. Here, we constrained the noise on the z axis to prevent divergence, both with LBA and BA, which use data only from a single monocular camera. Addressing this issue would probably require additional information or constraints on the target motion (multi-robot setup, additional sensors, geometric constraints, etc.).

[1] https://research.cc.gatech.edu/borg/download

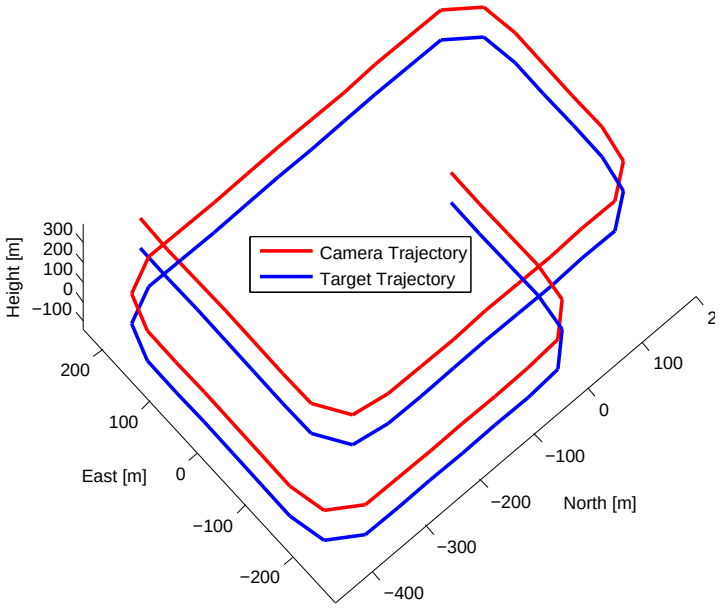


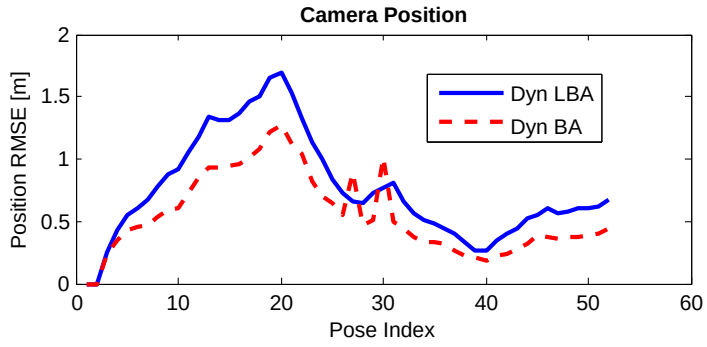
Figure 6.1: Scenario used for statistical study. Camera and target trajectories are shown in red and blue respectively.

6.1.1 Statistical Simulation Results

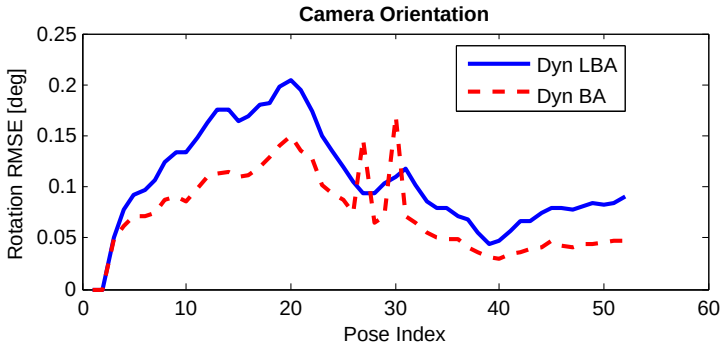
A performance comparison between the proposed method and BA with target tracking is presented, in a 45-run Monte-Carlo study. The scenario used in this simulation, showed in Figure 6.1, contains 52 frames, gathered over ˜160 seconds. Loop-closures can be noticed around view 20 and 38. The comparisons presented in Figure 6.2a-6.2c are given in terms of root-mean-square error (RMSE), calculated over the norms of the error vectors. All results refer to incremental estimations, i.e. at each time t_k performance is evaluated given Z_k, which is in particular important for online navigation.

Figures 6.2a and 6.2b describe the camera incremental position and orientation errors and Figure 6.2c shows the target position error. We observe similar levels of accuracy with the two techniques. The camera pose and target trajectory errors are bounded, with clear negative trend in both the camera and target position errors around view 20, upon loop-closure. We note that, in this case, the navigation is performed relatively to the camera's and target's initial positions. Those were initialized from their ground truth values, causing initial errors to be zero for all the estimated states.

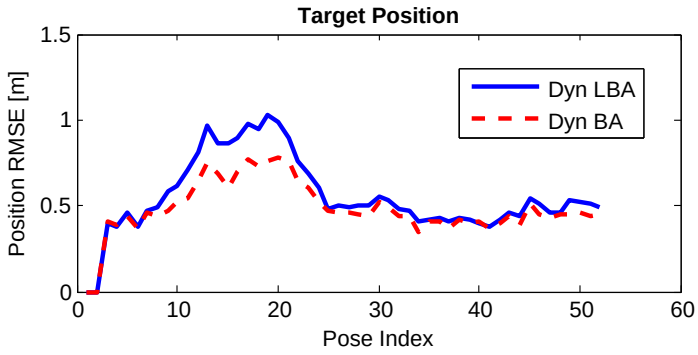
Figure 6.2d shows statistics over running time between the proposed method and full BA with target tracking. For BA, a distinct increase in computation time can be observed at view 38, where a loop-closure occurs. While one can already observe a significant difference in running time between the two methods in favor of LBA, we confirm this observation further in a larger scenario and with real imagery experiments in the next sections.



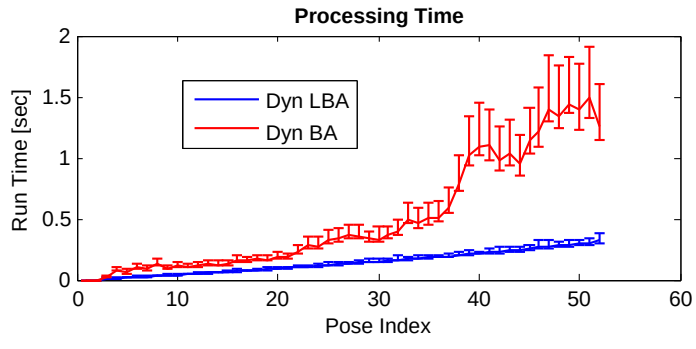
(a)



(b)



(c)



(d)

Figure 6.2: Monte-Carlo study results comparing between the proposed method and full BA with target tracking (a) Camera position RMSE; (b) Camera orientation RMSE (including close-up); (c) Target position RMSE; (d) Running time average with lower and upper boundaries.

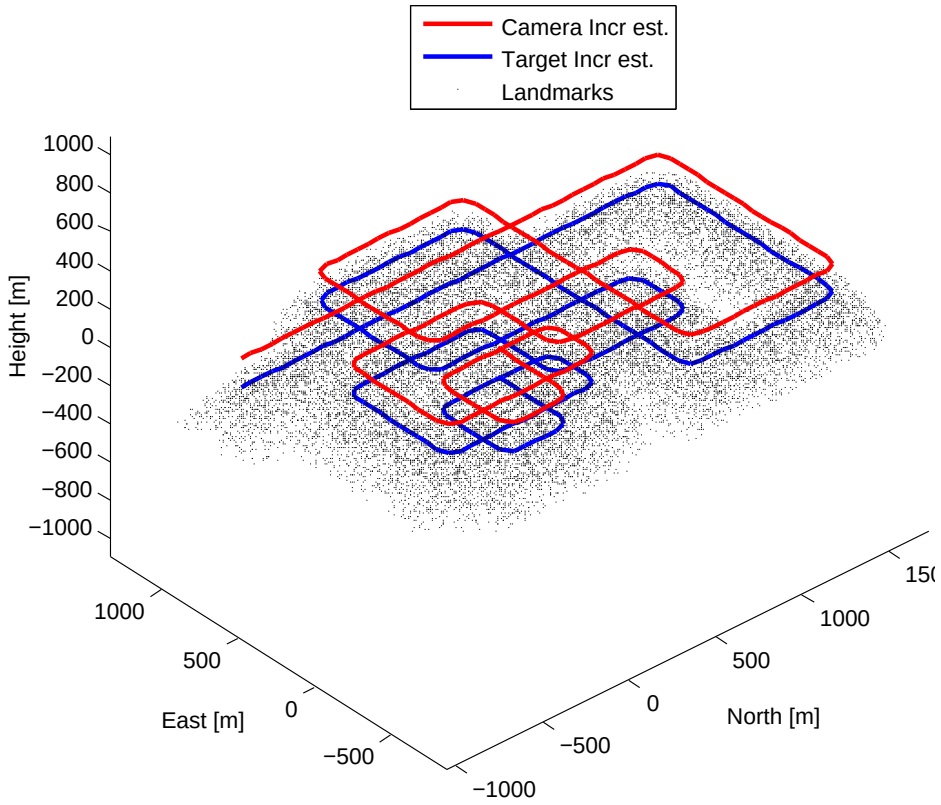
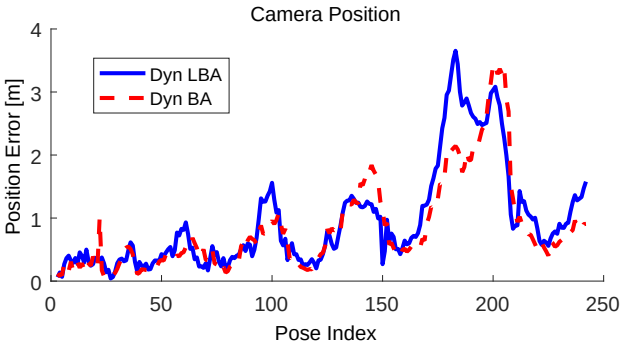


Figure 6.3: Large synthetic scenario with about 25300 observed landmarks (shown in black). Camera and target trajectories are shown in red and blue respectively.

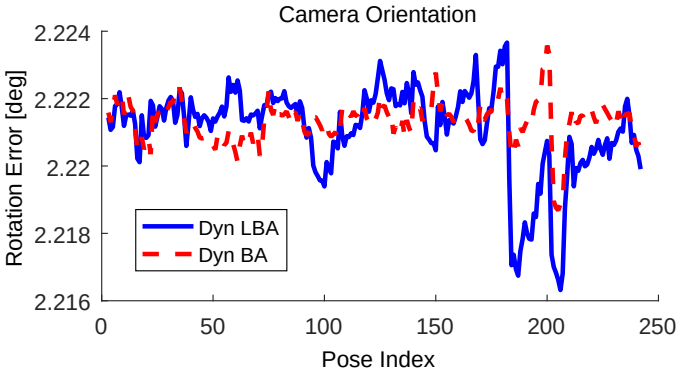
6.1.2 Large Scenario

The large scenario, showed in Figure 6.3, simulates an approximately 14.5km long aerial path and involves a series of loop closures, resulting in variables recalculation during optimization. As in the previous case, the target takes a similar course on the ground. A comparison in terms of accuracy and processing time are presented in Figure 6.4.

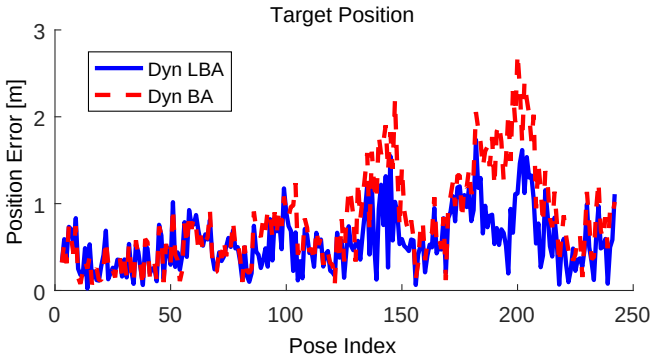
The obtained camera average position incremental errors for LBA and BA are 0.94 and 0.83 meters, respectively, with a maximum error of 3.65 and 3.45 meters. While the accuracy levels are similar, one can easily notice the difference in running time. Loop closures have a high impact on BA running time due to landmark re-elimination and re-linearization they trigger; this process is avoided with LBA. It results in an average processing time of 2.5 seconds for LBA with target tracking, versus 20.7 seconds for BA method. The obtained overall processing time for the same scenario is 602 seconds for the proposed method, versus 5020 seconds with BA.



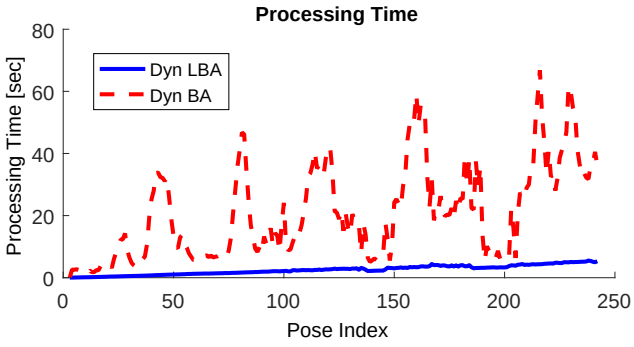
(a)



(b)



(c)



(d)

Figure 6.4: Comparison between the proposed method and full BA with target tracking for a large scale synthetic scenario

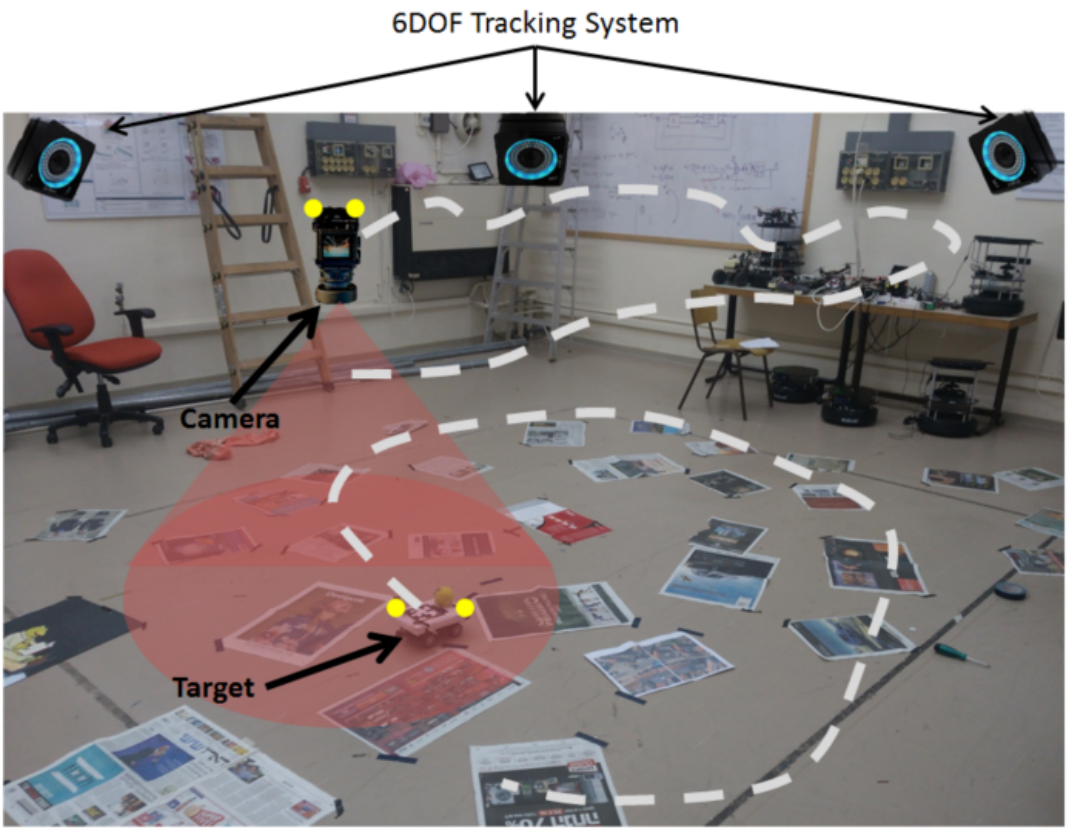


Figure 6.5: Scheme of the lab setup for the real-imagery experiments. The yellow dots represent the trackers installed on the platforms, allowing for detection by the ground truth system. Images were scattered on the floor to densify the observed environment. Best seen in colour

6.2 Experimental Evaluation with Real-Imagery Datasets

Further evaluations were performed through real-world experiments conducted at the Autonomous Navigation and Perception Lab (ANPL). Similarly to the synthetic dataset evaluation, these experiments involve a downward-facing camera which performed an aerial pattern while tracking a dynamic target moving on the ground. Ground truth data was gathered for the camera and the dynamic target using an independent real-time 6DOF optical tracking system. A scheme of the lab setup is presented in Figure 6.5 and two samples of typical captured images are presented in Figure 6.6. The recorded datasets are available online and can be accessed at `http://vindelman.net.technion.ac.il`.

Two different datasets were studied. In the first dataset, *ANPL1,* the camera and the target perform circular patterns, while in the second, *ANPL2*, they move in a more complex and unsynchronized manner, with occasional loss of target sight. Both cover an area of approximately $10\,[m] \times 6\,[m]$. In *ANPL1* the camera and target travel 26.9 and 34.6 meters respectively, while in *ANPL2*, the distance traveled is 19 and 21.1 meters respectively. Image sensing was performed using a high definition, wide angle

	Camera Resolution [pix]	Frames	Duration [sec]	Landmarks	Observations
ANPL1	1280×960	80	40	2439	31333
ANPL2	1920×1080	40	117	3366	25631

Table 6.1: Datasets details

Figure 6.6: Typical images from the *ANPL1* real-imagery dataset

camera and image distortion was corrected in the process using calibration data. Table 6.1 provides further details regarding the number of views and observations, camera settings and dataset durations.

Data association is performed using an implementation of the RANSAC algorithm [9]. The target is detected by identification of the most highly recurrent feature, although more advanced techniques exist but are outside the scope of this work. Since the experiments were conducted in a relatively constrained area with a wide field-of-view camera, numerous loop closures occur, as locations are often re-visited.

We compare the pose estimation errors of the camera and the position errors of the dynamic target with respect to ground truth for both LBA with target tracking and full BA cases. Incremental smoothing was applied for both methods in *ANPL1* dataset and standard batch optimization in *ANPL2*. QR factorization was used in all cases. We assume priors $p(x_0)$ and $p(y_0)$ on the initial camera and target states with means equal to their respective ground truth values and a $\sigma = 0.3$ [m] standard deviation. For the rest of the estimation process, new camera states are initialized by composition of last estimated pose with the relative motion from ground truth, corrupted with a white Gaussian noise $\sigma = 0.1$ [m] for position (i.e. the typical distance traveled between two frames), and $\sigma \sim= 5$ [deg] (0.09 [rad]) on each axis for orientation. A different option, tested with the LBA method, is composing the previous estimate and relative motion extracted from the essential matrix calculated during the data association process, as reviewed in Section 2.5. Results using the latter initialization method indicate similar performance with respect to the former initialization method. Here again, we use the constant velocity model for the dynamic target. This motion

	Target Position Error [m]		**Camera Position Error [m]**	
	Mean	Max	Mean	Max
ANPL1	0.07	0.19	0.06	0.18
ANPL2	0.14	0.42	0.08	0.34

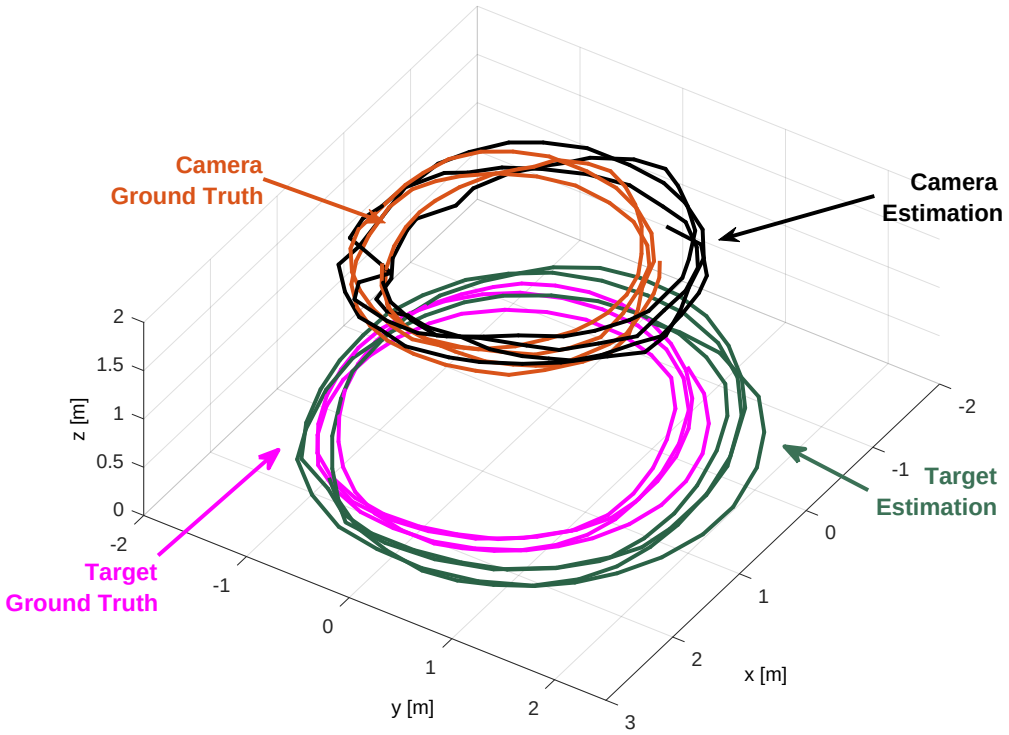
Table 6.2: Relative estimation errors summary of LBA method with respect to BA method for the camera and target positions in *ANPL1* and *ANPL2* datasets. The table entries are absolute values

		Processing Tine [sec]	
		Mean	Total
ANPL1	BA	5.6	447.8
	LBA	2.2	177.1
ANPL2	BA	3.1	222.9
	LBA	1.9	139.4

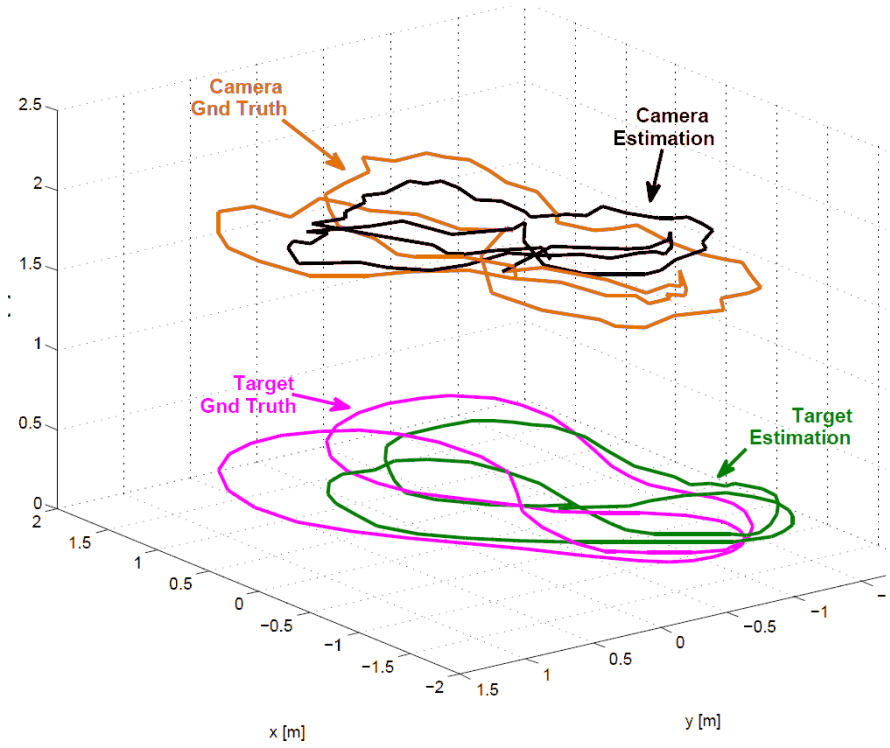
Table 6.3: Summary of the processing times with LBA and BA methods for the *ANPL1* dataset

model becomes the only available information for trajectory estimation when the target moves out of the camera's field of view, as it is the case for ˜15% of the frames in *ANPL2*. Similarly to the synthetic simulations, we assume the target moves on the ground, and thus constrain the first vertical velocity to zero. The measurement model assumes an image noise $\sigma = 0.5$ [pix].

Figure 6.7 shows the estimated trajectories and ground truth for the camera and the dynamic target in both datasets, using LBA method. We calculate an average error in position estimation of 22 and 38 centimeters for the camera and the target respectively in *ANPL1* dataset, and of 49 and 47 centimeters in the *ANPL2* dataset. The same level of position accuracy is calculated for the BA method. These errors are due to a specific practical data synchronization issue (ground thruth data vs. image sequence) during the experiment. Since we are interrested to assess the similarity in terms of accuracies between the two techniques, we show in Figures 6.8a to 6.8c the *relative* errors between LBA and BA methods, meaning the difference between the estimation errors using both methods. Then, a comparison of the processing time is shown in Figure 6.8d.

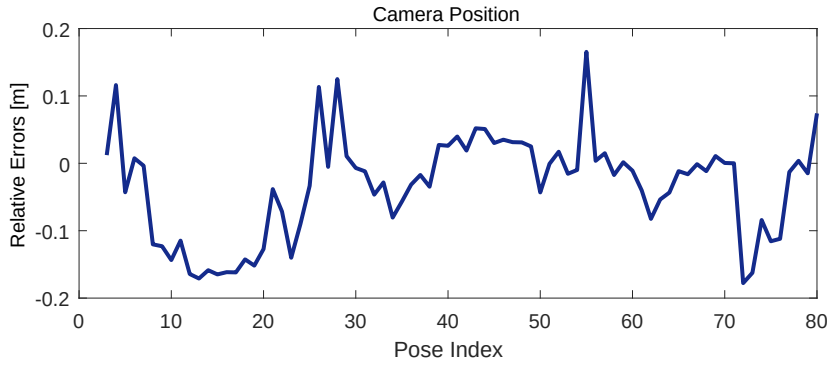


(a)

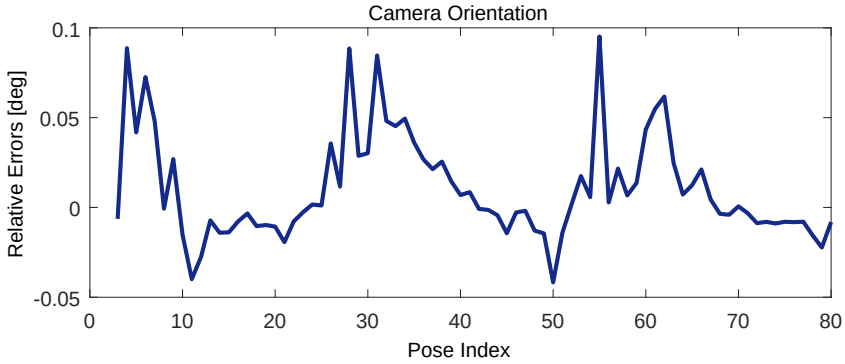


(b)

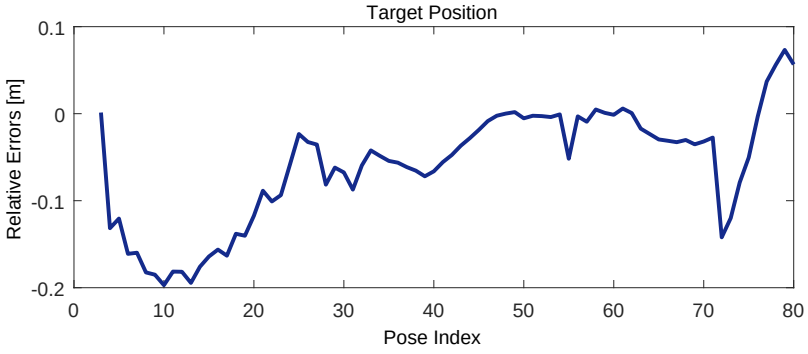
Figure 6.7: Estimated vs. ground truth 3D trajectories with real-imagery datasets for LBA approach in (a) *ANPL1* dataset (b) *ANPL2* dataset. BA approach produces similar results in terms of estimation errors, as shown in Table 6.2.



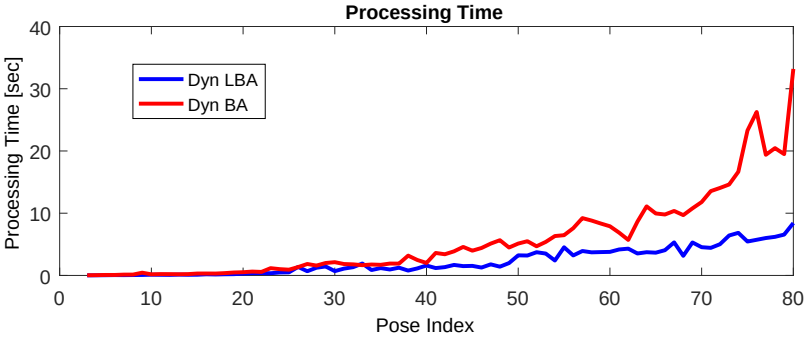
(a)



(b)



(c)



(d)

Figure 6.8: Incremental relative errors of LBA method with respect to BA method for the (a) camera position, (b) camera orientation, (c) target position, in *ANPL1* dataset. (d) presents a comparison of the processing times per frame

Table 6.2 and 6.3 summarize the absolute values of the relative errors and the processing times for the two datasets. In both cases, the two methods show similar levels of accuracy: The average values for target and camera positions are 7 and 6 centimeters respectively, for *ANPL1* dataset, and 14 and 8 centimeters for *ANPL2* dataset. In contrast, LBA with dynamic target tracking shows consequently better computational performances. The mean processing time per step is reduced by 61% for *ANPL1* and by 39% for *ANPL2*.

Chapter 7

Conclusions and Future Work

We presented an efficient method for simultaneous ego-motion estimation and target tracking using the LBA framework. By algebraically eliminating the observed landmarks from the optimization, we allow the target to become the only reconstructed 3D point in the process. This reduces significantly the number of variables compared to full BA methods, and thus, allows for processing time improvements. We presented the mathematical process involved in the integration of the target tracking problem into the LBA framework, leading to a cost function that is formulated in terms of multi-view constraints, target motion model and observations of the target. Computational efforts are further reduced by applying incremental inference over factor graphs representing the optimization problem, thus performing partial calculations at each optimization step.

We investigated performance of the proposed approach and compared it to the corresponding bundle adjustment formulation using synthetic and real-imagery datasets. While the two approaches exhibit similar accuracy levels, a significantly reduced running time was obtained for the proposed approach with both experimental methods. In particular, the presented method was up to eight times faster than full bundle adjustment in the simulations and up to two and a half times faster in the real-imagery experiments. This difference, however, is expected to vary with the number of landmarks observed per frame. The created real-imagery datasets have been made available to the research community through the ANPL website. These datasets include recorded images with synchronized ground truth for both the camera and the target, and is seen as a contribution by itself.

As for future work, the extension of this method to a multi-robot localization and/or multi-target tracking problem seems natural continuations, as those subjects have attracted great attention in recent years. In this case, the method used for target detection would have to be adapted and represents a real challenge in the frame of real-imagery experiments.

Bibliography

[1] Y. Bar-Shalom and T.E. Fortmann. *Tracking and data association.* Academic Press, New York, 1988.

[2] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software.* John Wiley & Sons, 2004.

[3] Denis Chekhlov, Andrew P Gee, Andrew Calway, and Walterio Mayol-Cuevas. Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–4. IEEE Computer Society, 2007.

[4] Ralph M Clendenin and Raymond S Freeman. Optical target tracking and designating system, June 7 1983. US Patent 4,386,848.

[5] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, Dec 2006.

[6] M.W.M.G. Dissanayake, P.M. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Automat.*, 17(3):229–241, 2001.

[7] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS titanic with SLAM information filters. In *Robotics: Science and Systems (RSS)*, Jun 2005.

[8] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans. Robotics*, 22(6):1100–1114, Dec 2006.

[9] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981.

[10] Dirk Hähnel, Dirk Schulz, and Wolfram Burgard. Mobile robot mapping in populated environments. *Advanced Robotics*, 17(7):579–597, 2003.

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[12] F.S. Hover, R.M. Eustice, A. Kim, B.J. Englot, H. Johannsson, M. Kaess, and J.J. Leonard. Advanced perception, navigation and planning for autonomous in-water ship hull inspection. *Intl. J. of Robotics Research*, 31(12):1445–1464, Oct 2012.

[13] Guoquan Huang, Ke Zhou, Nikolas Trawny, and Stergios I Roumeliotis. A bank of maximum a posteriori (map) estimators for target tracking. *IEEE Transactions on Robotics*, 31(1):85–103, 2015.

[14] V. Ila, J. M. Porta, and J. Andrade-Cetto. Information-based compact Pose SLAM. *IEEE Trans. Robotics*, 26(1), 2010. In press.

[15] V. Indelman. *Navigation Performance Enhancement Using Online Mosaicking.* PhD thesis, Technion - Israel Institute of Technology, 2011.

[16] V. Indelman. Bundle adjustment without iterative structure estimation and its application to navigation. In *IEEE/ION Position Location and Navigation System (PLANS) Conference*, April 2012.

[17] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Real-time vision-aided localization and navigation based on three-view geometry. *IEEE Trans. Aerosp. Electron. Syst.*, 48(3):2239–2259, July 2012.

[18] V. Indelman, R. Roberts, C. Beall, and F. Dellaert. Incremental light bundle adjustment. In *British Machine Vision Conf. (BMVC)*, September 2012.

[19] V. Indelman, R. Roberts, and F. Dellaert. Incremental light bundle adjustment for structure from motion and robotics. *Robotics and Autonomous Systems*, 70:63–82, 2015.

[20] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31:217–236, Feb 2012.

[21] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, Dec 2008.

[22] K. Konolige. Sparse sparse bundle adjustment. In *British Machine Vision Conf. (BMVC)*, September 2010.

[23] K. Konolige and M. Agrawal. FrameSLAM: from bundle adjustment to realtime visual mapping. *IEEE Trans. Robotics*, 24(5):1066–1077, 2008.

[24] F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001.

[25] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1442–1447, 1991.

[26] Alan J Lipton, Hironobu Fujiyoshi, and Raju S Patil. Moving target classification and tracking from real-time video. In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pages 8–14. IEEE, 1998.

[27] M.I. A. Lourakis and A.A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.

[28] Luis Montesano, Javier Minguez, and Luis Montano. Modeling the static and the dynamic parts of the environment to improve sensor-based navigation. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 4556–4562. IEEE, 2005.

[29] P. Moutarlier and R. Chatila. Stochastic multisensor data fusion for mobile robot location and environment modelling. In *5th Int. Symp. Robotics Research*, 1989.

[30] J. Neira, A.J. Davison, and J.J. Leonard. Guest editorial special issue on visual slam. *Robotics, IEEE Transactions on*, 24(5):929–931, Oct 2008.

[31] Joan Solà Ortega. *Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach.* PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2007.

[32] Ardhisha Pancham, Nkgatho Tlale, and Glen Bright. Literature review of slam and datmo. 2011.

[33] A. L. Rodríguez, P. E. López de Teruel, and A. Ruiz. Reduced epipolar cost for accelerated incremental sfm. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3104, June 2011.

[34] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Intl. J. of Robotics Research*, 5(4):56–68, 1987.

[35] R. Steffen, J.-M. Frahm, and W. Förstner. Relative bundle adjustment based on trifocal constraints. In *ECCV Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments*, 2010.

[36] R.E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13(3):566–579, 1984.

[37] Trung-Dung Vu. *Vehicle perception: Localization, mapping with detection, classification and tracking of moving objects.* PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2009.

[38] Trung-Dung Vu, Julien Burlet, and Olivier Aycard. Grid-based localization and local mapping with moving object detection and tracking. *Information Fusion*, 12(1):58–69, 2011.

[39] C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 842–849, 2003.

[40] Chieh-Chih Wang. *Simultaneous Localization, Mapping and Moving Object Tracking.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2004.

[41] Chieh-Chih Wang and Chuck Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2918–2924. IEEE, 2002.

[42] SE Wright. Uavs in community police work. *AIAA Infotechs., Aerospace, Arlington*, 2005.

[43] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202. IEEE Computer Society, 2008.

מורכבות חישובים מושפעת ממספר גורמים. בנוסף למספר המשתנים המעורבים בשיערוך, לאופן בו מבוצעת האופטימיזציה חשיבות גדולה. השיטות החדישות ביותר לSLAM מבוססות על אופטימיזציה "בקבוצה" (batch-optimization), כלומר, התהליך מבוצע מתחילתו בכל צעד או מספר צעדים. בשיטה זו, כל החישובים הנדרשים לתהליך האופטימיציה מבוצעים מחדש, ללא תלות בחישובים שנעשו בשלבים הקודמים. אנו מציעים, במסגרת עבודה זו, להשתמש בשיטת Incremental Smoothing and Mapping (iSAM), המאפשרת לבצע שימוש חוזר בחישובים מצעדים קודמים. שיטה זו מבוססת על שימוש במודל גראפי הנקרא Factor Graph, המתאר את הבעיה הפיזיקלית על ידי צמתים המסמלים את המשתנים האקראיים בבעיה (מצב המצלמה, מיקום המטרה וכדומה) וקשתות המסמלות את האילוצים בין המשתנים השונים (מודל דינאמי, מדידה וכדומה). בעזרת פעולות פשוטות המבוצעות על ה־Factor Graph , ניתן לזהות את המשתנים המושפעים מהוספת מידע חדש ובכך לבצע שימוש חוזר בחישובים קודמים המערבים את שאר המשתנים.

לאורך העבודה הזו, אנו מנסחים מתמטית את הבעיה הטמונה בשיערוך סימולטני של מצב פלטפורמה ותנועה של מטרה ניידת תוך שימוש בשיטת LBA. באמצעות ייצוג הסתברותי של הבעיה, אנו מבדילים בין שימוש בשיטת LBA לבין שיטות סטנדרטיות הכוללות ביצוע מיפוי הסביבה (BA). כמו כן, אנו מציגים את שיטת iSAM בה נעשה שימוש לצורך ייעול תהליך האופטימיזציה. יתרון השיטה המוצעת מודגם על ידי שימוש בסימולציות וניסויים המתבססים על תרחיש בו מצלמה מורכבת על פלטפורמה אוירית ומכוונת כלפי הקרקע, עליה נע רכב המהווה את המטרה. הסימולציות מבוצעות על בסיס נתונים סינטטי ומכילות אנליזה סטטיסטית על מסלול קצר יחסית וניתוח מקרה בוחן של מסלול ארוך המונה כ־ 250 תמונות. ניסויים בוצעו במעבדת Autonomous Navigation and Perception Lab (ANPL) ומכילים תחרישים שונים במורכבותם. ביצענו השוואה מבחינת זמני החישוב בין השיטה המוצעת לבין ביצוע BA סטנדרטי. למרות רמות דיוק דומות בין שתי השיטות, אנו מראים שיפור דרמאטי בזמן עיבוד ממוצע לצעד לטובת השיטה המוצעת, של עד פי 8 עבור הסימולציות ועד פי 2.5 עבור הניסויים המעבדתיים.

תקציר

בעבודה זו אנו מציגים גישה יעילה מבחינה חישובית לשיערוך סימולטני של מצב פלטפורמה דינאמית ותנועת מטרה ניידת, בסביבה לא ידועה וללא מידע חיצוני. שיערוך המצב של פלטפורמות דינאמיות (מיקום ואורינטציה) מהווה מזה זמן רב מקור עניין לטובת יישומים מגוונים כגון בקרת תנועה של מערכות אוויריות, ניווט אוטונומי ומציאות מדומה. עקיבה ושיערוך תנועה של מטרה ניידת מהווה גם היא יכולת קריטית ומשמשת מערכות מעקב אוויריות ויישומים צבאיים מזה מספר עשורים. למרות המחקר הרב בנושא, רוב העבודות מניחות את מיקום החיישן כידוע או מסלולו כצפוי. יישומים חדשים מתחום הרובוטיקה, כגון מעקב אוטונומי אחר מטרה ניידת או ממשק אדם־מכונה, מצריכים יכולות לשערוך התנועה של מטרה מפלטפורמות הנעות בסביבה לא ידועה.

בהעדר מידע חיצוני כגון GPS, קיים צורך לעשות שימוש בחיישנים המותקנים על הפלטפורמה בלבד. יכולת השימוש במצלמה כמקור מידע לצורך שיערוך תנועה התפתחה באופן ניכר בשני העשורים האחרונים, הודות לזמינות הפתרונות בשוק, מחירם הנמוך יחסית והאיכות ההולכת וגוברת של המידע הניתן להפקה. בעבודה זו נתמקד בבעית שיערוך תנועה על סמך מצלמה אחת כמקור מידע יחיד.

כאשר פלטפורמה נעה בסביבה לא ידועה, בעית שיערוך המיקום קשורה ישירות לבעית המיפוי: חישוב המיקום נעשה על סמך העצמים הנמצאים בסביבה, שאת מיקומם ניתן לשערך רק בהנתן מיקום המצלמה, ולכן, שתי הבעיות צריכות להפתר בעת ובעונה אחת. תהליך אופטימיזציה זה ידוע כ־ Bundle Adjustment (BA), או Simultaneous Localization And Mapping (SLAM) בקהילת הרובוטיקה. בתהליך משולב של MALS ועקיבה אחרי מטרה משוערכים מצב המצלמה, תנועת המטרה ומיפוי הסביבה בה התרחיש מתקיים. תהליך זה מבוצע באופן "מצטבר" כאשר בכל צעד, מתווספים משתנים חדשים ומדידות חדשות, דבר המגדיל באופן מתמיד את מורכבות הבעיה מבחינה חישובית. אחד האתגרים המכזיים ביישומים הדורשים תפעול מתמשך הינו לשמור על רמת מאמץ חישובי יציב ונמוך ככל האפשר לאורך זמן.

אנו מתייחסים ליישומים רובוטיים עבורם אין עניין בתהליך מיפוי הסביבה בזמן אמת. במקרים אלו, המאמץ הכרוך בשיערוך מיקום העצמים הנמצאים בסביבה מהווה ביזבוז משאבים, ולכן לשיטות המאפשרות להמנע ממיפוי הסביבה יש פוטנציאל לשיפור הביצועים החישוביים. קיימות מספר שיטות לביצוע SLAM ללא צורך במיפוי הסביבה. שיטות אלו מתבססות על שימוש באילוצים שאינם מערבים את העצמים (landmarks) הנמצאים בסביבה, דבר המקטין משמעותית את מספר המשתנים המעורבים באופטימיזציה. בעבודה זו, אנו עושים שימוש בשיטת Light Bundle Adjustment (LBA) שפותחה באחרונה, המבצעת שיערוך מצב מצלמה תוך שימוש באילוצים גיאומטריים המאפשרים להסיר מתהליך האופטימיזציה בצורה אלגברית את העצמים הסטטיים הנמצאים בסביבה. אנו משלבים עם שיטה זו שיערוך תנועה של מטרה ניידת, ההופכת לנקודה התלת מימדית היחידה המעורבת בחישוב.

המחקר בוצע בהנחייתו של פרופ"מ ואדים אינדלמן מפקולטה להנדסת אווירונאוטיקה וחלל ושל פרופ'
אהוד ריבלין מפקולטת מדעי המחשב בטכניון - מכון טכנולוגי לישראל

M. Chojnacki and V. Indelman. Vision-based target tracking and ego-motion estimation using incremental light bundle adjustment. In *Israel Robotics Conference*, 2016.

שערוך מבוסס ראייה ממוחשבת של מצב מצלמה ומסלול מטרה ניידת תוך שימוש בשיטת Incremental Light Bundle Adjustment

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים ברובוטיקה ומערכות אוטונומיות

מיכאל שושנקי

הוגש לסנט הטכניון --- מכון טכנולוגי לישראל
שבט התשע"ז חיפה פברואר 2017

שערוך מבוסס ראייה ממוחשבת של מצב מצלמה ומסלול מטרה ניידת תוך שימוש בשיטת Incremental Light Bundle Adjustment

מיכאל שושנקי