THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue le *TBD* **par :**
MÉDÉRIC FOURMY

Tightly coupled legged robot state estimation

JURY

Rapporteur
Rapporteur
Examinateur

École doctorale et spécialité :
EDSYS : Robotique 4200046

Unité de Recherche :
LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes (UPR 8001)

Directeur(s) de Thèse :
Nicolas MANSARD et *Joan Solà*

Rapporteurs :
et

Contents

Intro

Chapter 1

Introduction

For robots, perception of oneself and of its environment is a major challenge on the road toward many real world applications. Tasks that are instinctive to us, like for instance manipulating an object, actually involve a complex interactions between our many senses, our nervous and our muscular system. Though our sensory motor skills have been developed by millions of year of evolution, and are refined throughout our childhood, the task of representing them in abstracted algorithm to be implemented on a cybernetic system is a serious challenge. Let's investigate the example a human lifting package. Our vision might inform us about the general form of the object, its location in space with respect to us, some of its physical properties through our prior knowledge of the world. Our proprioception instinctively guide our arms toward the right path. Our sense of touch might infer the surface texture of the object, its softness, making us adapt our grip. During this whole process, our vestibular system provides us with a sense of balance to counter gravity, while our hears make us aware of events external to our current enterprise.

All these complex phenomena happen mostly at the subconscious level while our conscious mind focuses on high level decisions. Imitating these skills in robot system requires then to build models of available sensor modalities and to integrate them through sensor fusion. This can be achieved at several levels depending on the task to solve. In the legged robot community, one of the core task is locomotion. For this application, robust algorithms exist in the literature using a limited set of sensors, most often inertial and contact detection. On the other end of the blind robot approach, a broad field of research has been concentrated on building representations of the environment using exteroceptive sensors such as cameras and LIDARs. This in turn enables planning algorithms to navigate the robot in its environment. Many approaches decouple the two tasks, using layered perception systems. However, theoretically, a system able to tightly fuse all the available modalities would benefit a better consideration of the correlations between the different quantities to estimate. Even though recent approaches have taken step in this direction, such a system is still not widely used in legged robotics. This thesis is a contribution to this goal.

Chapter 2

Why tightly coupling the estimation?

2.1 Legged robot state estimation

[1]

2.2 Graph optimization state estimation

[2]

Part I

Theory

Chapter 3

Tutorial on MAP estimation

Contents

Mix of Barfoot, Sola, Kaess etc.

3.1 Geometry

Notations Lie theory primer

3.2 Probabilities

Primer of Probabilities PDF Bayes rule Gaussian special properties Probabilities on Lie algebra

3.3 Estimation as factor graph optimization

MAP problem as Factor graph NLLS problems NLLS on Lie groups

Chapter 4

Object level vision

Contents

4.1 General factor

Let's assume that an algorithm provides us ${}^{C}\widetilde{\mathbf{T}}_{O} \in SE(3)$, a measurement the pose of an element of the scene with an attached frame O with respect to the camera frame C located at the Camera optical frame. The kinematic chain of the problem described in 4.1 unrolls as ${}^{W}\mathbf{T}_{O} = {}^{W}\mathbf{T}_{B}\,{}^{B}\mathbf{T}_{C}\,{}^{C}\mathbf{T}_{O}$ where W and B correspond to the world and body frames. Given measurement ${}^{C}\widetilde{\mathbf{T}}_{O}$, this relation can therefore be turned into a residual relating the robot pose, the camera extrinsics and the object pose:

$$\mathbf{r}({}^{W}\mathbf{T}_{B}, {}^{B}\mathbf{T}_{C}, {}^{W}\mathbf{T}_{O}) = \mathrm{Log}({}^{W}\mathbf{T}_{O}\,{}^{W}\mathbf{T}_{B}^{-1}\,{}^{B}\mathbf{T}_{C}^{-1}\,{}^{C}\mathbf{T}_{O}^{-1}) \ \in \mathbb{R}^{6} \tag{4.1}$$

We also assume that we have access to the covariance of this measurement $\mathbf{\Sigma_{V}} \in \mathbb{R}^{\mathbf{6\times 6}}$. We will now describe two applications of this factor, one using Apriltag fiducial markers and one using a Deeplearning object pose estimation algorithm.

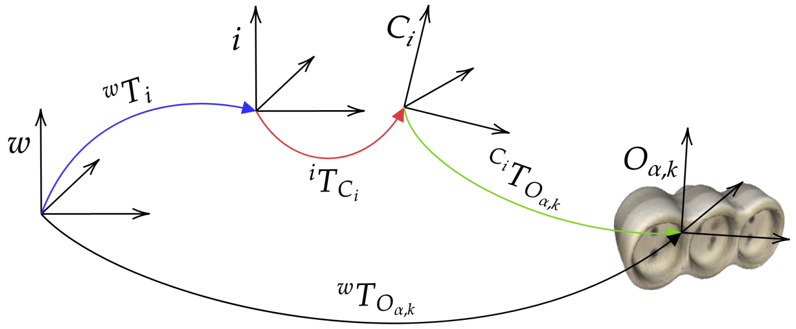

Figure 4.1: Camera/object kinematic chain from ICRA paper, to replace

4.2 Fiducial marker

4.2.1 Markers Pose estimation algorithms

Sota on the question Apriltag: [3] IPPE: [4]

4.2.2 Covariance model

Appart from designing more precise and more efficient algorithm for fiducial marker detection and specialized PnP algorithms, obtain covariances from these prediction has been the focus of a series of papers. SOTA

Instead of directly obtaining Jacobians from the PnP algorithms, we found that a natural way to proceed is to take the opposite direction. It should somehow be possible, knowing the marker size and the relative pose measurement, and assuming pixel noise, to recover $\mathbf{\Sigma_{OV}}$. A simple model of this noise is to assume isotropic gaussian noise on the pixels. If we stack four pixel (the four tag corners) $\mathbf{x}_i = [u_i, v_i] \in \mathbb{R}^2$ and we stack then in the vector $\mathbf{x} = [\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4]$, we have therefore that $\mathbf{x}$ is corrupted by a gaussian noise $\mathbf{\Sigma_x} = \sigma_\mathbf{x}^\mathbf{2}\mathbf{I}$, where σ_x usually takes values of 1 or 2 pixels. PnP algorithm provides us with a function *pnp* defined as:

$$\begin{aligned} f : \mathbb{R}^8 &\rightarrow SE(3) \\ \mathbf{x} &\rightarrow {}^{C}\mathbf{T}_O = pnp_w(\mathbf{x}) \end{aligned} \tag{4.2}$$

where w denotes the dependency on the width of the marker. This pnp_w function implementation depends on the specificities of the PnP algorithm used and is in general hard to differentiate. Instead if we consider the inverse function

$$\begin{aligned} g : SE(3) &\rightarrow \mathbb{R}^8 \\ {}^{C}\mathbf{T}_O &\rightarrow \mathbf{x} = proj_w({}^{C}\mathbf{T}_O) \end{aligned} \tag{4.3}$$

that maps the relative pose to the projection of the tag in the image, a rather simple jacobian expression can be derived using the chain rule, as follows.

Let's defined the marker corner coordinates in the marker frame, like shown in Fig.4.2:

Figure 1: AprilTag 36h11 ID = 12

Figure 4.2: Dummy tag image, annotate with coordinates

$$c = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} \quad c_1 = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix} \quad c_2 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \quad c_3 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad c_4 = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \tag{4.4}$$

which coordinates have been rescaled by a factor $\frac{w}{2}$, that will be reintroduced later.

Then, assuming that images are corrected (no distortion), the pinpoint camera model gives us that

$$\mathbf{x}_i = \mathit{eucl}(h_i) = \mathit{eucl}(K\,{}^{C}\mathbf{T}_O c_i) \tag{4.5}$$

for each corner c_i, where h_i are the homogeneous coordinates representing the projected corners and *eucl* is the euclideanization function defined as

$$\begin{aligned} &\mathit{eucl} : \mathbb{R}^3 \to \mathbb{R}^2 \\ &h = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \to \mathbf{x} = \begin{pmatrix} x/z \\ y/z \end{pmatrix} \end{aligned} \tag{4.6}$$

We need to compute the jacobian of each corner projection with respect to the estimated relative pose $J^{x_i}_{\mathbf{T}} = J^{x_i}_{h_i} J^{h_i}_{\mathbf{T}}$. Regarding the transformation, we will consider it to be an element of $\mathbb{R}(3) \times SO(3)$ since the translation and rotation part of transformation are treated separately in our solver. The expressions of those functions are therefore expressed as:

$$\begin{aligned} &h_i = K({}^{C}\mathbf{R}_O c_i + {}^{C}\mathbf{p}_O) \\ &J^{h_i}_{{}^{C}\mathbf{p}_O} = K \qquad J^{h_i}_{{}^{C}\mathbf{R}_O} = -K\,{}^{C}\mathbf{R}_O[c_i]_\times \\ &J^{h_i}_{{}^{C}\mathbf{T}_O} = [J^{h_i}_{{}^{C}\mathbf{p}_O} \;\; J^{h_i}_{{}^{C}\mathbf{R}_O}] = K[I_3 \quad -{}^{C}\mathbf{R}_O[c_i]_\times] \end{aligned} \tag{4.7}$$

while the euclideanization jacobian is found to be

$$J^{x_i}_{h_i} = \begin{pmatrix} 1/z_i & 0 & -x_i/z_i^2 \\ 0 & 1/z_i & -y_i/z_i^2 \end{pmatrix} \tag{4.8}$$

Finally, we can stack the 4 jacobians to get the full jacobian to be used for covariance propagation.

$$J \triangleq J^{\mathbf{x}}_{{}^C\mathbf{T}_O} = \frac{w}{2} \begin{pmatrix} J^{\mathbf{x}_1}_{{}^C\mathbf{T}_O} \\ J^{\mathbf{x}_2}_{{}^C\mathbf{T}_O} \\ J^{\mathbf{x}_3}_{{}^C\mathbf{T}_O} \\ J^{\mathbf{x}_4}_{{}^C\mathbf{T}_O} \end{pmatrix} \in \mathbb{R}^{8\times 6} \tag{4.9}$$

reintroduction the $\frac{w}{2}$ factor common to the 4 corners.

We therefore have the covariance propagation equation $Q_{\mathbf{x}} = JQ_{{}^C\mathbf{T}_O}J^T$. This equation must be inverted in order to recover the needed covariance. J being non square, we have to use the pseudo inverse to write: $\mathbf{\Sigma}_{{}^C\mathbf{T_O}} = \mathbf{J}^{\mathbf{T},\dagger}\mathbf{\Sigma}_{\mathbf{x}}\mathbf{J}^{\dagger}$. Knowing that the pixel noise covariance is isotropic as explained above, this equation simplifies to:

$$Q_{{}^C\mathbf{T}_O} = \sigma^2_{\mathbf{x}}(J^T J)^{-1} \tag{4.10}$$

4.3 Object pose reconstruction

Cosypose

Small sota + general cosypose workings

Empirical covariance estimation

blabla

Chapter 5

Preintegrated sensors

Contents

5.1 Generalized preintegration

5.2 IMU preintegration

5.3 IMU preintegration on Lie groups

5.4 External force preintegration

Part II

Applications

Chapter 6

Fiducial marker based visual inertial SLAM

Chapter 7

Centroidal estimation

Chapter 8

Cosy SLAM

Part III

Conlusion

Just the conclusion

Bibliography

[1] Maurice F Fallon, Matthew Antone, Nicholas Roy, and Seth Teller. “Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing”. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE. 2014, pp. 112–119 (cit. on p. 5).

[2] Frank Dellaert, Michael Kaess, et al. “Factor graphs for robot perception”. In: *Foundations and Trends® in Robotics* 6.1-2 (2017), pp. 1–139 (cit. on p. 5).

[3] John Wang and Edwin Olson. “AprilTag 2: Efficient and robust fiducial detection”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2016 (cit. on p. 12).

[4] Toby Collins and Adrien Bartoli. “Infinitesimal plane-based pose estimation”. In: *International journal of computer vision* 109.3 (2014), pp. 252–286 (cit. on p. 12).