

Tracking a Varying Number of People with a Visually-Controlled Robotic Head

Yutong Ban¹, Xavier Alameda-Pineda¹, Fabien Badeig¹, Sileye Ba^{1,2} and Radu Horaud¹

Abstract—Multi-person tracking with a robotic platform is one of the cornerstones of human-robot interaction. Challenges arise from occlusions, appearance changes and a time-varying number of people. Furthermore, the final system is constrained by the hardware platform: low computational capacity and limited field-of-view. In this paper, we propose a novel method to simultaneously track a time-varying number of persons in three-dimensions and perform visual servoing. The complementary nature of the tracking and visual servoing enables the system to: (i) track several persons while compensating for large ego-movements and (ii) visually control the robot to keep a selected person of interest within the field of view. We propose a variational Bayesian formulation allowing us to effectively solve the inference problem through the use of closed-form solutions. Importantly, this leads to a computationally efficient procedure that runs at 10 FPS. The experiments on the NAO-MPVS dataset confirm the importance of using visual servoing for tracking multiple persons.

I. INTRODUCTION

Robots are currently on the verge of sharing many common spaces with humans. Exemplar scenarios are the front desk of a hotel, museum guides, elder assistance or entertainment for children. In all these situations, and many others, the robotic platform is required to interact with people and, as part of its low-level behavioral skills, to perform person tracking and visual servoing. In practice this means that the robot is supposed to keep track of the locations of the people in the scene and, once a person of interest has been chosen, to keep that person within its visual field of view.

Visual servoing, i.e. robot control based on visual information, has been a well studied problem [1], [2]. Several methods were developed targeting different applications, such as grasping [3], mobile robot navigation [4] or autonomous aerial vehicle guidance [5]. In this paper we are interested in visual servoing using a robot head, commonly referred to as head-eye coordination, which was studied in a wide range of applicative scenarios involving *a single* object/person of interest [6], [7], [8], [9], [10], [11], [12], [13], [14] and based on methodologies such as detect and pursuit, image feature tracking, or Kalmann filtering.

However, the vast majority of everyday situations consist of several persons. Clearly, not all these persons are of interest for the HRI task at hand. Nevertheless, the robot should be able to jointly perform multiple person tracking

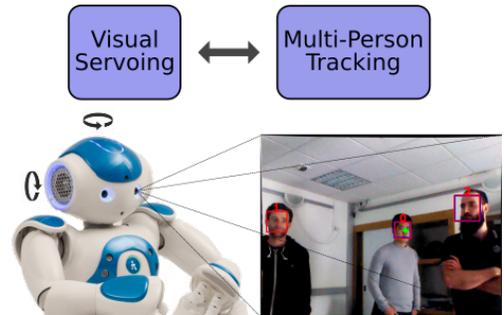


Fig. 1. Schematic overview of the system. The visual servoing module estimates the optimal robot commands and the expected impact of the tracked positions. The multi-person tracking module refines the positions of the persons with the new observations and the information provided by the visual servoing.

and visual servoing of one or a few persons. Compared to single-person methods, the presence of many people remains challenging. First, computationally cheap face/person detection algorithms often deliver noisy detections or even fail to provide a consistent sequence of bounding boxes. Second, even under the hypothesis of high-quality face/person detections, we are still left with the task of correctly associating these detections over time. For instance, [15] proposed to use an EKF for each tracked person, often leading to bad detection-to-person assignments and thus estimating roaming tracks. More computationally demanding algorithms exist, such as particle filtering, *e.g.* [16], but their use in real time applications is rather limited. Third, in real life scenarios, people will continuously appear and disappear from the field-of-view of the robot, and it is highly desirable to robustly track persons that disappear and reappear later on.

There is a plethora of methodologies that address the multiple-object (or person) tracking problem. For example, [17] tackled the problem by combining a sparse representation-based appearance model with a sliding window, and [18] proposed and aggregated local flow descriptor and a dynamic graphical model that is optimized off-line. To the best of our knowledge, most of the existing methods are not designed to deal with controlled camera motions, even if some of them are partially robust to ego-motion, since they need to extract feature points from the background in order to estimate camera motions [19]. It is not straightforward to take ego-motion information into account in case it is available. As we experimentally show, this can cause a huge drop on tracking performance, in particular when addressing the complex scenarios just mentioned.

¹INRIA Grenoble Rhône-Alpes, Montbonnot Saint-Martin, France

² VideoStitch, Paris, France

Work supported by the ERC Advanced Grant “*Vision and Hearing in Action*” (VHIA) #340113.

In order to overcome this issue, we propose to embed visual servoing into the multi-person tracker, as schematically shown in Figure 1. Visual servoing requires the estimation of a Jacobian matrix that maps observed image features onto motor velocities, which in turn requires 3-D information. We do this by combining a person detector with a calibrated camera pair mounted onto the robot head. The estimated motor velocities are then explicitly taken into account by the person tracker itself. The latter is formulated as a Bayesian filtering method and we propose to use a variational approximation [20], [21]. Indeed, this solution is particularly efficient from a computational point of view and hence it is preferred over more standard sampling methods for the following advantages: (i) it is able to handle a number of persons that varies over time, and (ii) it is robust to disappearing/reappearing persons.

To summarize, we propose a joint multi-person tracking and visual servoing method that is able to simultaneously estimate the three-dimensional position of a time-varying number of people and to encompass the effect of the robot’s motion on this estimation. This complements both visual servoing, by leveraging current methods from single-object tracking to multiple-object tracking, and by explicitly taking ego motion into account. We propose a Bayesian filter and its variational approximation allowing to effectively solve the inference problem of the filtering distribution while keeping a reasonably low computational load (the overall system works at 10 FPS). Third, we report a large experimental study on the NAO multi-person visual servoing (NAO-MPVs) dataset showing, not only that the addressed scenarios are challenging, but also that including the impact of the robot’s motion into the probabilistic tracking framework is of utmost importance for the performance of the system.¹

The remaining of the paper is organized as follows. Section II presents the probabilistic tracker which is interleaved with the visual servoing module detailed in Section III. The system architecture is described in Section IV. The experimental protocol and the results are reported in Section V. Section VI concludes the paper.

II. MULTIPLE-PERSON TRACKING

We adopt the probabilistic multiple person tracking formulation recently proposed in [20]. Let N_t denote the number of persons at time t . Let $\mathbf{X}_{tn} \in \mathbb{R}^3$ and $\mathbf{B}_{tn} \in \mathbb{R}^2$ denote the position of person n at t and its bounding box (width and height), respectively. Making use of the three-dimensional locations in the joint tracking-servoing method has two prominent advantages. First, it leads to a more stable tracker that is also more robust to object occlusions. Second, it allows us to compute the Jacobian associated to the visual servoing in closed form, and therefore the expected effect of the robot motion into the observed scene can be computed without any prior knowledge about the persons to be tracked

¹Supplemental material for this paper can be found at <https://team.inria.fr/perception/mot-servoing/>

(see Section III). This is a crucial advantage over existing methods, since it allows to encompass the effect of the robot control in the tracking framework and therefore *to infer the persons’ locations by taking the robot motion into account*.

Aiming to privilege smooth trajectories, we track the velocity and the bounding box of each person in addition to his/her position. More formally, the tracking state variable is a concatenation of three variables: $\mathbf{M}_{tn} = [\mathbf{X}_{tn}^T, \mathbf{B}_{tn}^T, \dot{\mathbf{X}}_{tn}^T]^T$. These variables are expressed in the coordinate frame of the camera pair. Below we describe the probabilistic model (Section II-A) from which we derive the filtering distribution (Section II-B) based on a variational Bayes approximation [22]. The birth process allowing to take into account new disappearing/reappearing persons is detailed in Section II-C.

A. Probabilistic model

The probabilistic model consists of two main components. On one hand, the tracking state dynamics delineates the probabilistic behavior of the state variable over time. On the other hand, the observation model associates the state variable at current t , \mathbf{M}_{tn} , to the observations. Such an association is modeled by assignment variables $\{Z_{tk}\}_{k=1}^{K_t}$, namely $Z_{tk} = n, n \in \{1, \dots, N_t\}$, observation k at time t is assigned to person n .

1) *The state dynamics*: The state dynamics models the temporal evolution of the state variable. We make two hypotheses. Firstly, we assume that at each time instance, the assignment variable Z_t doesn’t depend on Z_{t-1} . Therefore, we can factorize the dynamic distribution into the observation-to-person prior distribution and the predictive distribution. Secondly, the state dynamics follow a first-order Markov chain, meaning that \mathbf{M}_{tn} only depends on \mathbf{M}_{t-1n} :

$$p(Z_t, \mathbf{M}_t | Z_{t-1}, \mathbf{M}_{t-1}) = \prod_{n=1}^N p(\mathbf{M}_{tn} | \mathbf{M}_{t-1n}) \prod_{k=1}^{K_t} p(Z_{tk}). \quad (1)$$

The two modeling choices that need to be done are: the prior probability of the assignment variable Z_{tk} and the dynamic model. A priori, there is no reason to believe that one person is more prone to generate observations than another one, hence we set $p(Z_{tk}) = a_{tk} = \frac{1}{N_t+1}$, for all k . Regarding the dynamics of \mathbf{M}_t , we propose a transition model that takes the robot’s motion explicitly into account. Indeed, let \mathbf{C}_{tn} denote the expected \mathbf{X}_{t-1n} due to the motion of the robot (see Section III-B). Importantly, \mathbf{C}_{tn} can be computed in closed-form thanks to the proposed formulation. We concatenate \mathbf{C}_{tn} with a 5-dimensional vector of zeros (that would correspond to the expected shift of the bounding box and the velocity), and construct a 8-dimensional vector that for the sake of simplicity we will also denote with \mathbf{C}_{tn} . The explicit computation of \mathbf{C}_{tn} , described in detail in Section III, allows us to better predict when a person appears, disappears, or reappears in the field of view. Notice that, due to the potentially large appearance variation, the use of the

geometric proprioceptive information may become crucial for the tracking performance. More formally, we model the transition probability with a Gaussian distribution defined as:

$$p(\mathbf{M}_{tn}|\mathbf{M}_{t-1n}) = \mathcal{N}(\mathbf{M}_{tn}; \mathbf{D}\mathbf{M}_{t-1n} + \mathbf{C}_{tn}, \Lambda_n), \quad (2)$$

where \mathbf{C}_{tn} is a translation associated to the effect of the controlled robot motion (see Section III for details), Λ_n models the uncertainty over the dynamics of the n -th source, and \mathbf{D} is the following matrix:

$$\mathbf{D} = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{I}_3 \\ \mathbf{0} & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{pmatrix}.$$

This definition of \mathbf{D} is equivalent to have a first order model on the dynamics of the person. In other words, the bounding box and the velocity do not change, while the position changes according to the previous velocity.

2) *The observation model:* The observation model provides a principled definition of the probabilistic relationship between the observations and the tracking state. Let us assume that at every time t there are K_t observations, denoted by \mathbf{o}_{tk} , $k \in \{1, \dots, K_t\}$. The way we obtain observations is described in Section IV-4. Here we assume $\mathbf{o}_{tk} = [\mathbf{g}_{tk}^\top, \mathbf{a}_{tk}^\top]^\top$, where \mathbf{g}_{tk} (resp. \mathbf{a}_{tk}) provides some geometric (resp. appearance) information of the person.

One of the modelling difficulties briefly discussed above is that, given that the detectors are not perfect, misdetections, false positive and noisy detections frequently happen. In order to account for these problems, the categorical random variable $Z_{tk} \in \{1, \dots, N_t\}$, assigning observation k at time t to source $\{1, \dots, N_t\}$ is provided by:

$$p(\mathbf{o}_t|Z_t, \mathbf{M}_t) = \prod_{k=1}^{K_t} p(\mathbf{g}_{tk}|Z_{tk}, \mathbf{M}_t) p(\mathbf{a}_{tk}|Z_{tk}), \quad (3)$$

where Z_t and \mathbf{M}_t are defined analogously to \mathbf{o}_t . Notice that, while the geometric information depends on the state variable \mathbf{M}_t and on the assignment variable Z_{tk} , the appearance only depends on the latter. Implicitly we are assuming that the appearance does not depend on the position in a way that can be intuitively modeled.

For both the geometric and the appearance observations we assume that they can belong either to a clutter class, which writes $Z_{tk} = 0$, or to a person. In other words we extend the set in which Z_{tk} takes values with a *garbage* class to attract all noisy observations. Observations belonging to this extra class follow an uniform distribution on \mathbf{g}_{tk} and on \mathbf{a}_{tk} . Otherwise, we model the geometric information with a Gaussian distribution:

$$p(\mathbf{g}_{tk}|Z_{tk} = n, \mathbf{M}_t) = \mathcal{N}(\mathbf{g}_{tk}; \mathbf{P}\mathbf{M}_{tn}, \Sigma), \quad (4)$$

where $\mathbf{P} = (\mathbf{I}_5 \mathbf{0}) \in \mathbb{R}^{5 \times 8}$ extracts the three-dimensional position of the source and the size of the bounding box from the state vector \mathbf{M}_{tn} and Σ models the uncertainty

of the detector. The model for the appearance follows a Bhattacharya distribution:

$$p(\mathbf{a}_{tk}|Z_{tk} = n) = \mathcal{B}(\mathbf{a}_{tk}; \mathbf{a}_{tn}^*) = \frac{1}{W_\lambda} \exp(-\lambda d_B(\mathbf{a}_{tk}, \mathbf{a}_{tn}^*)), \quad (5)$$

where λ is a positive skewness parameter, $d_B(\cdot)$ is the Bhattacharya distance between feature vectors, W_λ is the normalization constant and \mathbf{a}_{tn}^* is the appearance model closest to the observations \mathbf{a}_{tk} among those that have been previously associated to person n . This is done so as to integrate the high-performance philosophy of tracking-learning-detection [23] into the proposed probabilistic model, and therefore keep on enriching the appearance model with the newly available observations.

B. Variational inference

In order to merge all the previous observations together with the current information gathered at time t , we write the *filtering distribution of the hidden random variables*:

$$p(Z_t, \mathbf{M}_t | \mathbf{o}_{1:t}) \propto p(\mathbf{o}_t | Z_t, \mathbf{M}_t) p(Z_t, \mathbf{M}_t | \mathbf{o}_{1:t-1}), \quad (6)$$

where the second term is the so-called predictive distribution, which is related to the filtering distribution at time $t-1$ by:

$$p(Z_t, \mathbf{M}_t | \mathbf{o}_{1:t-1}) = p(Z_t) \int_{\mathbf{M}_t} p(\mathbf{M}_t | \mathbf{M}_{t-1}) \sum_{Z_{t-1}} p(Z_{t-1}, \mathbf{M}_{t-1} | \mathbf{o}_{1:t-1}).$$

Since (6) does not accept a computationally tractable closed-form expression, we choose to use a variational approximation [22]. If properly designed, such approximations have the prominent advantage of deriving into closed-form updates for the a posteriori (filtering) probabilities. Concisely, variational approximations consist on imposing a partition over the hidden variables, in our case:

$$p(Z_t, \mathbf{M}_t | \mathbf{o}_{1:t}) \approx \prod_{n=1}^{N_t} q(\mathbf{M}_{tn}) \prod_{k=1}^{K_t} q(Z_{tk}), \quad (7)$$

and then finding the optimal distributions $q(\mathbf{M}_{tn})$ and $q(Z_{tk})$ in the Kullback-Leibler distance sense.

The optimal posterior distribution of the assignment variable $q(Z_{tk})$ writes:

$$q(Z_{tk} = n) = \alpha_{tkn} = \frac{\varepsilon_{tkn} a_{tkn}}{\sum_{m=0}^N \varepsilon_{tkm} a_{tkm}}, \quad (8)$$

with a_{tkn} is the prior assignment probability for identity n and ε_{tkn} is defined as:

$$\begin{cases} \mathcal{U}(\mathbf{g}_{tk}) \mathcal{U}(\mathbf{a}_{tk}) & n = 0, \\ \mathcal{N}(\mathbf{g}_{tk}, \mathbf{P}\boldsymbol{\mu}_{tn}, \Sigma) e^{-\frac{1}{2} \text{trace}(\mathbf{P}^\top \Sigma^{-1} \mathbf{P} \Gamma_{tn})} \mathcal{B}(\mathbf{a}_{tk}; \mathbf{a}_{tn}^*) & n \neq 0, \end{cases}$$

where $\text{trace}(\cdot)$ is the trace operator and $\boldsymbol{\mu}_{tn}$ and Γ_{tn} are defined by (9) and (10) below. Intuitively, the assignment of an observation to a person is based on spatial proximity between the geometric observation and the current estimated position as well as the similarity between the observation's appearance and the person's previous appearances.

The a posteriori distribution for \mathbf{M}_{tn} turns out to be a Gaussian distribution with mean μ_{tn} and covariance Γ_{tn} given by:

$$\Gamma_{tn} = \left(\Lambda_n^{-1} + \sum_{k=0}^{K_t} \alpha_{tkn} \mathbf{P}^\top \Sigma^{-1} \mathbf{P} \right)^{-1}, \quad (9)$$

$$\mu_{tn} = \Gamma_{tn} \left(\Lambda_n^{-1} (\mathbf{D}\mu_{t-1n} + \mathbf{C}_{tn}) + \sum_{k=0}^{K_t} \alpha_{tkn} \mathbf{P}^\top \Sigma^{-1} \mathbf{g}_{tk} \right) \quad (10)$$

where μ_{t-1n} is the expected position of person n in the previous time step. These two steps are commonly iterated a few times at every time step. Remarkably, this strategy can also be used to learn the parameters of the model, for which we would then be required to derive the so-called M-step. The reader is referred to [20] for an exhaustive discussion.

C. Birth process

Until now we assumed that N_t was known, but this is an unrealistic assumption in real-world applications. We model the variability of N_t with a track birth process, that allows creating new ‘‘identities’’. At time t , after the algorithm has estimated the a posteriori distribution for Z_t and \mathbf{M}_t we perform a test to decide whether a new track is created or not. Intuitively, we test if the observations assigned to the clutter class in the past T_{new} frames are consistent enough to belong to a person that has not been seen before. Let $\mathcal{H} = \{\mathbf{g}_{t'k}, \mathbf{a}_{t'k}\}_{t'=t-T_{new}, Z_{tk}=0}$ such set of observations. If $p(\mathcal{H}) > \tau$, where τ is the probability that these observations are generated by clutter, a new track is created, μ_{tN_t+1} is set to the most recent geometric observation in \mathcal{H} , and the appearance observations are assigned to the new person.

III. VISUALLY-CONTROLLED HEAD MOVEMENTS

In this section we detail the visual servoing model allowing the robot to focus its attention on targets of interest. In order to simplify the discussion we remove the temporal and person indices t and n . The objective of the visual servoing module is to compute the required motor velocity to bring the person of interest to the center of the image. Therefore we need the Jacobian linking the image space to the motor space. Since such relationship is difficult to model, classically one models the motor-to-image Jacobian and then computes the inverse. In our case, we pass by the three-dimensional world and compute the Jacobian as the composite of a world-to-image Jacobian and a motor-to-world Jacobian.

A. World-to-image Jacobian

We consider the coordinate system associated to the left camera (at the initial head’s position) to be the world’s coordinate system. This is an arbitrary choice that can be replaced with any other static coordinate system with a simple rigid transformation. The non-linear mapping between world-coordinate and image-coordinate is:

$$\mathbf{V} = \mathbf{K} \frac{1}{X_3} \mathbf{X}, \quad (11)$$

where $\mathbf{X} = (X_1, X_2, X_3)^\top$, $\mathbf{V} = (V_1, V_2)^\top$ and $\mathbf{K} \in \mathbb{R}^{2 \times 3}$ is the matrix of intrinsic parameters of the pinhole camera model. The Jacobian of this transformation writes:

$$\dot{\mathbf{V}} = \underbrace{\mathbf{K} \begin{pmatrix} 1/X_3 & 0 & -X_1/X_3^2 \\ 0 & 1/X_3 & -X_2/X_3^2 \end{pmatrix}}_{\mathbf{J}_{wi}(\mathbf{X})} \dot{\mathbf{X}}, \quad (12)$$

where $\dot{\mathbf{X}}$ is the velocity vector at \mathbf{X} , and \mathbf{V} models the velocity as seen in the left camera image.

B. Motor-to-world Jacobian

In order to compute the Jacobian relating the velocity at \mathbf{X} and the motor velocity, we first recall that in a general, the velocity of a three-dimensional point when the coordinate system is subject to a rigid motion, namely a rotation $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$ and a translation $\mathbf{u} = [u_1, u_2, u_3]^\top$, can be expressed as:

$$\dot{\mathbf{X}} = \boldsymbol{\omega} \times \mathbf{X} + \mathbf{u} = \begin{pmatrix} \mathbf{S}(\boldsymbol{\omega}) & \mathbf{u} \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}, \quad (13)$$

where $\mathbf{S}(\boldsymbol{\omega})$ is the skew symmetric twist matrix representing the exterior product by the three-dimensional vector $\boldsymbol{\omega}$.

In our case, $\boldsymbol{\omega}$ and \mathbf{u} depend on the motor yaw and pitch rotation velocities $\dot{\alpha}$ and $\dot{\beta}$ respectively. As shown in the literature [24], for a rotation velocity $\dot{\alpha}$, the velocity at \mathbf{X} can be expressed as:

$$\dot{\mathbf{X}} = \begin{pmatrix} -\mathbf{S}(\mathbf{X}) & \mathbf{I}_3 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \mathbf{u}_1 \end{pmatrix} \dot{\alpha}, \quad (14)$$

where the values of ω_1 and \mathbf{u}_1 are acquired through a calibration phase (see Section IV-.3).

The effect of the pitch is quite similar, with the only difference that, since we first apply the yaw rotation and then the pitch rotation, one has to take into account the effect of $\dot{\beta}$ after the rotation induced by $\dot{\alpha}$. Formally we write:

$$\dot{\mathbf{X}} = \begin{pmatrix} -\mathbf{S}(\mathbf{X}) & \mathbf{I}_3 \end{pmatrix} \begin{pmatrix} \mathbf{R}\omega_2 \\ -\mathbf{S}(\mathbf{R}\omega_2)\mathbf{t} + \mathbf{R}\mathbf{u}_2 \end{pmatrix} \dot{\beta}, \quad (15)$$

where ω_2 and \mathbf{u}_2 are obtained through the calibration and \mathbf{R} and \mathbf{t} are the rotation and translation vectors associated to the yaw state α . In all, the motor-to-world Jacobian writes:

$$\dot{\mathbf{X}} = \underbrace{\begin{pmatrix} -\mathbf{S}(\mathbf{X}) & \mathbf{I}_3 \end{pmatrix} \mathbf{L}(\alpha)}_{\mathbf{J}_{mw}(\mathbf{X})} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix}, \quad (16)$$

where $\mathbf{L}(\alpha) \in \mathbb{R}^{6 \times 2}$ is a matrix that implicitly depends on the calibration parameters ω_1 , ω_2 , \mathbf{u}_1 and \mathbf{u}_2 .

Importantly, since this equation is true for any point in the scene \mathbf{X} , it can be applied to estimate predicted people’s current position from the previous time step, *i.e.* $\mathbf{D}\mu_{t-1n}$. By doing this, we compute the velocity of the person due to the robot’s motion. In other words, at time t , the n -th person will not be around position $\mathbf{D}\mu_{t-1n}$, but close to $\mathbf{D}\mu_{t-1n} + \mathbf{J}_{mw}(\mathbf{D}\mu_{t-1n}) \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix}$. This is the value given to the translation due to the robot control:

$$\mathbf{C}_{tn} = \mathbf{J}_{mw}(\mathbf{D}\mu_{t-1n}) \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix}. \quad (17)$$

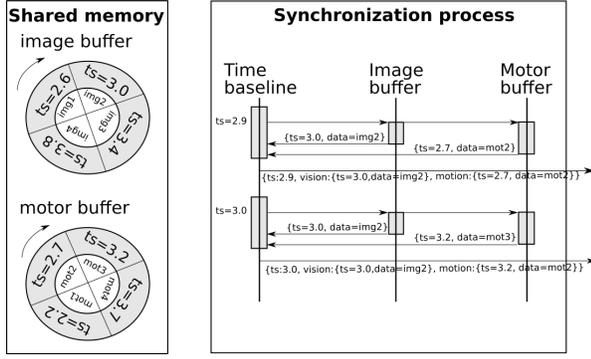


Fig. 2. Robot data synchronization with NAOLab. The shared buffers contain time-stamped data. During the synchronization process, the nearest pairs of data are associated together regarding the time chosen time baseline.

C. Joint Jacobian

The joint motor-to-image Jacobian is the product of the Jacobians above:

$$\dot{\mathbf{V}} = \mathbf{J}_{wi}(\mathbf{X})\mathbf{J}_{mw}(\mathbf{X}) \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix} = \mathbf{J}(\mathbf{X}) \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix}. \quad (18)$$

To summarize, we are interested in two Jacobian operators. First, the inverse of the motor-to-image Jacobian maps the desired image shift $\Delta\mathbf{V}$ into motor velocities:

$$\begin{pmatrix} \dot{\alpha}_c \\ \dot{\beta}_c \end{pmatrix} = \gamma \mathbf{J}^{-1}(\mathbf{X}_s) \Delta\mathbf{V}, \quad (19)$$

where \mathbf{X}_s is the servo position in three-dimension, $0 < \gamma < 1$ is a scale factor and $\dot{\alpha}_c$ and $\dot{\beta}_c$ are the yaw and pitch velocities to control the robot. Second, we can estimate the impact of these motor velocities onto the people's position by computing \mathbf{C}_{in} using (17) with $\dot{\alpha}_c$ and $\dot{\beta}_c$.

IV. SYSTEM AND ARCHITECTURE

The proposed joint multi-person tracking and visual servoing system is implemented on top of the *NAOLab* middleware, which utilises the synchronisation strategy to find temporal matches between proprioceptive and perceptive information. A motor-camera calibration procedure is used to estimate the spatial relationship between the motor and the camera coordinate systems. At the end of the section, implementation details of the whole system is introduced.

1) *NAOLab*: There are several reasons to use a middleware architecture. First, algorithm implementations can be platform-independent and thus easily portable. Second, the use of external computational resources is transparent. Third, prototyping is much faster. For all these reasons, we developed a remote and modular layer-based middleware architecture named *NAOLab*.

NAOLab consists of 4 layers: drivers, shared memory, synchronization engine and application programming interface (API). Each layer is divided into 3 modules devoted to vision, audio and proprioception respectively. The first layer is platform-dependent and interfaces the sensors and actuators

through the network using serialized data structures. The second layer implements a common shared memory that provides a concurrent interface to deserialize data from the robot sensors and implements an event-based control for robot command. The third layer is dedicated to synchronize the audio, video and proprioception data, so that the joint tracking-servoing system handles temporally coherent information. The last layer of *NAOLab* provides a general programming interface in C++ or Matlab to handle the robot's sensor data and manage its actuators.

2) *Synchronization engine of NAOLab*: The synchronization is implemented in the third *NAOLab* layer thanks to a circular data buffer (initialized to a fixed maximum size). The synchronization engine exploits these circular buffers together with the robot clock, and builds packages containing audio, visual and proprioception data whose corresponding time-stamps are close to each other. Figure 2 depicts the synchronization process for the multi-person tracking and visual servoing system (without audio involved), with a time baseline of 0.1 s and a buffer size of four packages.

As illustrated in Figure 3, the robot produces vision and proprioception data at different sampling rates. Each type of data is grabbed by a dedicated parallel process (drivers) who *publishes* the serialized data into the shared memory. After synchronization, the joint tracking and visual servoing module is able to request data from the shared memory or send motor-control commands to the motion drivers.

3) *Motor-camera calibration*: As previously discussed, the motor-to-world Jacobian required for the visual servoing depends on four parameters obtained through calibration: ω_1 , ω_2 , \mathbf{u}_1 and \mathbf{u}_2 . In order to do that, we first notice that when the robot's head rotates from α_0 to α_i , there is an extrinsic rotation matrix $\mathbf{Q}_{0 \rightarrow i}$ that can be expressed as a function of ω_1 and \mathbf{u}_1 :

$$\mathbf{Q}_{0 \rightarrow i} = \mathbf{I}_4 + \sin(\alpha_i - \alpha_0) \begin{pmatrix} \mathbf{S}(\omega_1) & \mathbf{u}_1 \\ \mathbf{0} & 0 \end{pmatrix} + (1 - \cos(\alpha_i - \alpha_0)) \begin{pmatrix} \mathbf{S}(\omega_1) & \mathbf{u}_1 \\ \mathbf{0} & 0 \end{pmatrix}^2. \quad (20)$$

At the same time, thanks to the cameras, the external matrix can be estimated with visual information. Indeed, the images of a static chessboard are recorded before and after the rotation, and by manually detecting the chessboard in the image, one can estimate the extrinsic matrix $\tilde{\mathbf{Q}}_{0 \rightarrow i}$. Based on the previous equation and on the properties of the trigonometric functions one can write:

$$J(\omega_1, \mathbf{u}_1) = \sum_i \left\| 2 \sin(\alpha_i - \alpha_0) \begin{pmatrix} \mathbf{S}(\omega_1) & \mathbf{u}_1 \\ \mathbf{0} & 0 \end{pmatrix} - \tilde{\mathbf{Q}}_{0 \rightarrow i} + \tilde{\mathbf{Q}}_{i \rightarrow 0} \right\|_F^2$$

where $\|\cdot\|_F$ is the Frobenius norm. This cost function is then minimized to find the optimum values for the calibration parameters ω_1 and \mathbf{u}_1 . The analogous procedure is repeated for the calibration parameters ω_2 and \mathbf{u}_2 .

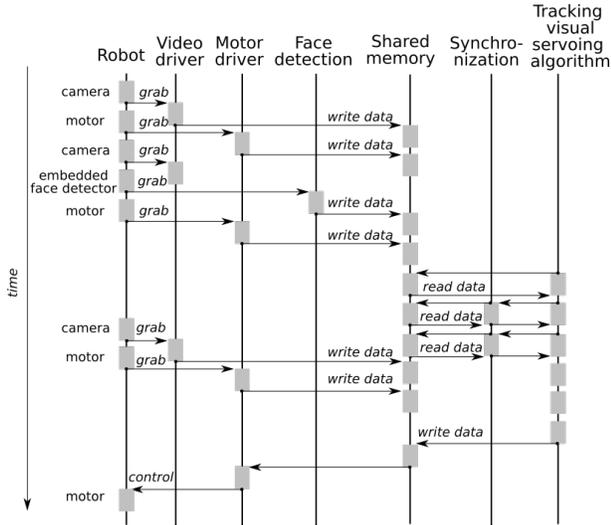


Fig. 3. Data temporal flow chart: the drivers published serialized data into the shared memory. After synchronization, the joint tracking and servoing algorithm requests the data from which computes the appropriate motor control command, sent to the motor drivers through the shared memory.

4) *Implementation details*: The overall system is implemented in C++, within the middleware framework described in Section IV-1. For the sake of reproducibility, we use the face detector and descriptor built-in on NAO, *i.e.* provided by NAO’s API. The geometric observations, \mathbf{g}_{jk} are face bounding boxes (image position, width and height). The position of the bounding box from the left and right camera images is combined by means of epipolar geometry, and triangulation to recover 3D face position. The face appearance descriptor is based on color histograms. Importantly, the detector and descriptor can be replaced or combined with other techniques thanks to the flexibility of the proposed probabilistic model for tracking. The detection and description of faces runs at 10 frames per second (FPS). Since the joint tracking-servoing computational load is less than 70 ms per time step, we are able to provide an on-line implementation of the joint multi-person tracking and visual servoing system.

The proposed variational model is governed by several parameters. Aiming at providing an algorithm that is dataset-independent and that features a good trade-off between flexibility and performance, we set the observation covariance matrix Σ and the state covariance matrix Λ_n automatically from the detections. More precisely, both matrices are imposed to be diagonal; for Σ , the variances of the three-dimensional position, of the width, and of the horizontal (resp. vertical) speed are $1/2$, $1/2$ and $1/4$ of the average detected width (resp. height). The rationale behind this choice is that we consider that the true detection lies approximately within the width and height of the detected bounding box. Regarding Λ_n , the diagonal entries are half of the tracked width and 5 times of motor speed. The window length chosen for the birth process is $T_{new} = 4$.

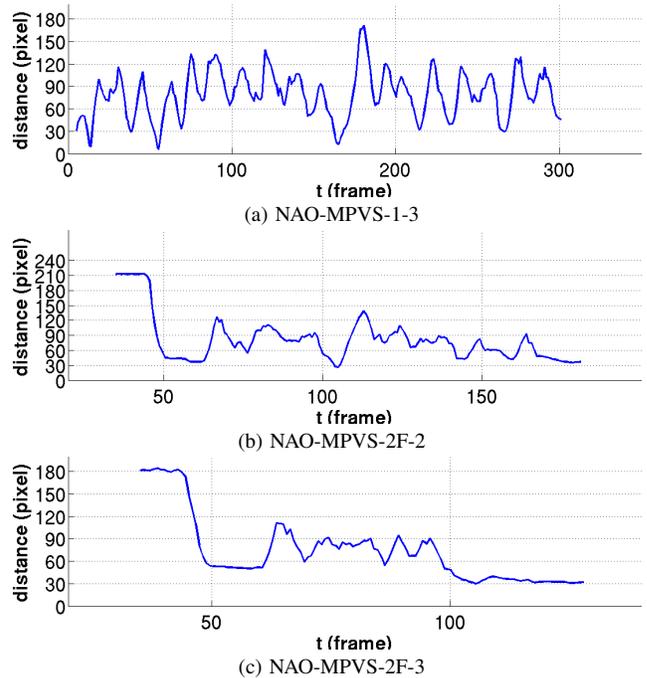


Fig. 4. The distance between tracked person and left camera image center (in pixels) over time (in frames) for three different sequences.

V. EXPERIMENTS

The proposed joint multi-person tracking and visual servoing system is evaluated on a series of scenarios using the NAO robot. Both left and right cameras provide VGA images, which are 640×480 pixels. Ten different sequences have been recorded in a regular living room scenario with its usual lighting source and background, where various people were moving around. The recorded sequences are thus challenging because of illumination variations, occlusions, appearance changes, and people leaving the robot field-of-view. We tried two different high-level control rules: (i) the robot should servo the first tracked person and (ii) the robot should sequentially change the pursued person every three seconds. The sequences with the servoing-tracking results are publicly available ². The sequences are named with the following scheme: NAO-MPVS-NS-P, which stands for NAO Multi-Person Visual Servoing, N is the number of people present in the sequence (although not constantly visible), S defines the strategy when $N > 1$ (“F” for following the first tracked person and “J” for jumping every three seconds) and P for the trial. For instance NAO-MPVS-1-1, is the first trial of a scenario involving one person, while NAO-MPVS-2J-3 is the third trial of a scenario involving two people and the control rule set to “jumping”. In the following, we provide both quantitative and qualitative evaluation of both the visual servoing and of the multi-person tracking.

1) *Visual servoing*: Figure 4 shows the distance in pixels from the tracked person to the left camera image center over time, for three different sequences of the dataset, all under the servoing strategy of following the first tracked person.

²<https://team.inria.fr/perception/mot-servoing/>

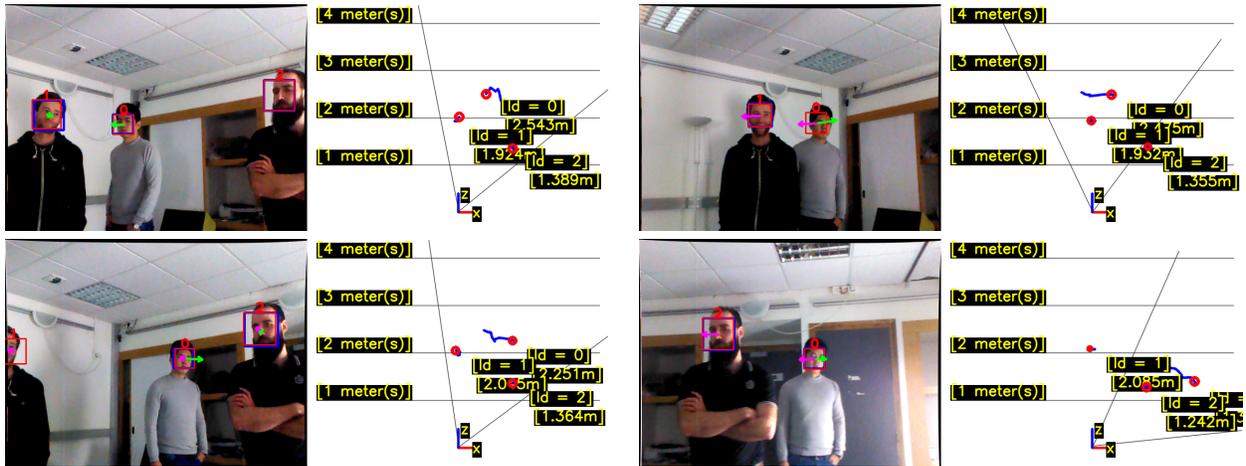


Fig. 5. Left: robot left camera view, red bounding boxes represent the three-dimensional tracking results projected on the image, blue bounding boxes represents three-dimensional face-detection, green arrows represent people’s self-velocity, magenta arrows represent the velocity due to the robot control. Right: scenario bird-view, red circles represent current tracking positions and the blue lines represent tracked people’s trajectories. Example results are from NAO-MPVS dataset sequence NAO-MPVS-3F-1

We can clearly see the oscillation due to the lag between the person’s motion and the control response. Remarkably, shortly after each of the person’s movements, the servoing mechanism position back the person in the image center. Indeed, after a few seconds the distance between the tracked person and the image center has decreased to below 30 pixels. Furthermore, if we compute the average distance for all frames of all sequences (*i.e.* almost 2,000 frames), we obtain an average distance of 80.1 pixels, indicating that the proposed system is able to approximately maintain the person’s face at the image center.

Qualitatively speaking, Figure 5 shows four frames of the most challenging sequence in the dataset, NAO-MPVS-3F-1. This sequence involves three people, among which the tracked one passes behind the other two. Each of the frames shows the marked-up left camera image together with a bird-view representation of the tracked scene. While in the marked-up image we can see the face detection (blue), the tracked bounding box with the tracking ID (red), the target motion due to the robot control (magenta) and the target’s self-motion (green), in the bird-view we can see the tracking ID and the trajectories. We can observe different prominent characteristics of the proposed system. Firstly, the ability to separate the image motion due to the robot control, from the image motion due to the natural movements of the target allows for the estimation of a smooth trajectory in the three-dimensional space. Secondly, the algorithm is able to keep a rough estimate of the positions of the targets that are out of the field of view, and even more important, to correctly re-assign the identity to a re-appearing person thanks to the cooperation of the state dynamics and appearance model. Thirdly, the capacity of the system to create a new track when a new person appears in the field-of-view thanks to the birth process. Finally, robustness to identify switches, even with illumination and appearance changes, occlusions and the robot’s self-motion.

TABLE I
COMPARISON OF TRACKING RESULT W/O AND W CONTROL BY MOT METRICS ON THREE SEQUENCES WITH INCREASING COMPLEXITY.
↑ : THE HIGHER THE BETTER ↓ : THE LOWER THE BETTER

Sequence	Ctrl	MOTA(↑)	MOTP(↑)	FP(↓)	FN(↓)	IDs(↓)
NAO-MPVS-1-1	w/o	92.1	67.2	9	9	0
	w/	91.3	68.7	10	10	0
NAO-MPVS-2J-1	w/o	52.8	67.1	93	207	2
	w/	81.6	68.0	30	88	0
NAO-MPVS-3J-1	w/o	35.8	62.3	159	433	19
	w/	63.1	62.1	83	268	0
Overall	w/o	48.8	65.0	261	649	21
	w/	73.1	65.3	123	366	0

2) *Multi-person tracking*: We have also evaluated the impact of the visual servoing from the multi-person tracking perspective. Aiming to this, we compared the performance of the system when using/discarding the image-motion due to the robot control. In more detail, we manually annotated the position of the persons in three different sequences of increasing complexity (NAO-MPVS-1-1, -2J-1 and -3J-1) and we computed the following standard multi-person tracking evaluation metrics [25]: *multiple-object tracking accuracy* (MOTA), *multiple-object tracking precision* (MOTP), *false positives* (FP), *false negatives* (FN) and *identity switches* (ID). While for MOTA and MOTP are the higher the better, for the rest are the lower the better. Table I reports all these measures with (w/) and without (w/o) using the impact of the robot control (Ctrl) on the targets’ position, *i.e.* C_{in} .

In light of the results, we can see that indeed NAO-MPVS-1-1 is an easy sequence. Indeed, the only person to be tracked does not perform large movements. It is therefore not surprising that (i) the performance measures are very high and (ii) there is no much performance difference when adding the impact of the control variable. When the complexity of the scenario increases (more people to track, larger movements) the proposed tracking framework including motor information leads to higher accuracy and less FP/FN/IDs results. This difference is specially remarkable in the case

of the NAO-MPVS-3J-1 sequence, showing that tracking based only on the appearance and the position of people is not sufficient when multiple people need to be tracked, while at the same time the robot performs some movements. We also notice that the MOTP measure is not strongly affected by the information provided by the robot control, and this is expected. Indeed, MOTP measures the tracking precision in terms of how much do the bounding boxes of the detected positives overlap with their assigned true positives. In other words, if a detected positive is too far from all true positives, it counts as a FP, but is not computed as a precision error. This confirms our hypothesis that the use of C_m is crucial to correct large deviations of the tracking estimates due to the motor control; And at the same time result shows that it is not specially helpful to refine these tracking estimates. In other words, the use of C_m is complementary to developing precise tracking methodologies which are able to provide very accurate bounding box localization once the large corrections due to the motor-control are applied.

Overall the proposed joint multiple-person tracking and visual servoing framework leads to promising results even in sequences which contain large and frequent robot motions under challenging illumination conditions. Remarkably, the method is able to systematically keep the right person identity for all three sequences. This feature is highly desirable for numerous applications and critically depends upon the use of motor information during tracking.

VI. CONCLUSIONS

This paper proposes a novel joint variational multi-person tracking and visual servoing system which is able to continuously estimate the three-dimensional position of a time-varying number of people and to encompass the effect of the robot's ego-motion. In addition, we propose a probabilistic formulation and a variational approximation allowing to effectively solve the inference problem while keeping a reasonably low computational cost (the overall systems works at 10 FPS). Furthermore, thanks to the motor information, the system can separate people's self-motion from the robot's ego-motion, leading to more robust tracking capabilities. The experimental study on the NAO Multi-Person Visual Servoing dataset confirms our hypothesis that including the robot's ego-motion into the tracking probabilistic framework is of utmost importance for the performance of the system. In the future, we will investigate (i) the calibration of other motors (e.g. robot's leg-joint), thus compensating for the full ego-motion and (ii) the combination of audio information to construct a tracking system based on audio-visual information, thus able to track outside the camera field-of-view.

REFERENCES

- [1] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, 1993.
- [2] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, 1999.
- [3] A. A. Moughlby, E. Cervera, and P. Martinet, "Error regulation strategies for model based visual servoing tasks: Application to autonomous object grasping with nao robot," in *International Conference on Control Automation Robotics & Vision*, 2012.
- [4] C. Gaskett, L. Fletcher, A. Zelinsky *et al.*, "Reinforcement learning for visual servoing of a mobile robot," in *Australian Conference on Robotics and Automation*, 2000.
- [5] L. Mejias, S. Saripalli, P. Campoy, and G. S. Sukhatme, "Visual servoing of an autonomous helicopter in urban areas using feature tracking," *Journal of Field Robotics*, vol. 23, no. 3-4, 2006.
- [6] A. Crétual, F. Chaumette, and P. Bouthemy, "Complex object tracking by visual servoing based on 2d image motion," in *International Conference on Pattern Recognition*, vol. 2, 1998.
- [7] A. Crétual and F. Chaumette, "Application of motion-based visual servoing to target tracking," *The International Journal of Robotics Research*, vol. 20, no. 11, 2001.
- [8] R. Stolkin, I. Florescu, M. Baron, C. Harrier, and B. Kocherov, "Efficient visual servoing with the abcsift tracking algorithm," in *IEEE International Conference on Robotics and Automation*, 2008.
- [9] É. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, 2005.
- [10] D. Omrčen and A. Ude, "Redundant control of a humanoid robot head with foveated vision for object tracking," in *IEEE International Conference on Robotics and Automation*, 2010.
- [11] J. Rajruangrabin and D. O. Popa, "Robot head motion control with an emphasis on realism of neck-eye coordination during object tracking," *Journal of Intelligent & Robotic Systems*, vol. 63, no. 2, 2011.
- [12] D. J. Agravante, J. Pages, and F. Chaumette, "Visual servoing for the reem humanoid robot's upper body," in *IEEE International Conference on Robotics and Automation*, 2013.
- [13] M. Antonelli, A. P. Del Pobil, and M. Rucci, "Bayesian multimodal integration in a robot replicating human head and eye movements," in *IEEE International Conference on Robotics and Automation*, 2014.
- [14] L. Vannucci, N. Cauli, E. Falotico, A. Bernardino, and C. Laschi, "Adaptive visual pursuit involving eye-head coordination and prediction of the target motion," in *IEEE-RAS International Conference on Humanoid Robots*, 2014.
- [15] J. Satake and J. Miura, "Robust stereo-based person detection and tracking for a person following robot," in *IEEE ICRA Workshop on People Detection and Tracking*, 2009.
- [16] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *IEEE International Conference on Robotics and Automation*, 2001.
- [17] L. Fagot-Bouquet, R. Audigier, Y. Dhome, and F. Lerasle, "Improving multi-frame data association with sparse representations for robust near-online multi-object tracking," in *European Conference on Computer Vision*, 2016.
- [18] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *IEEE International Conference on Computer Vision*, 2015.
- [19] W. Choi, C. Pantofaru, and S. Savarese, "A general framework for tracking multiple people from a moving camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, 2013.
- [20] S. Ba, X. Alameda-Pineda, A. Xompero, and R. Horaud, "An on-line variational bayesian model for multi-person tracking from cluttered scenes," *Computer Vision and Image Understanding*, vol. 153, 2016.
- [21] Y. Ban, S. Ba, X. Alameda-Pineda, and R. Horaud, "Tracking multiple persons based on a variational bayesian model," in *European Conference on Computer Vision Workshops*, 2016.
- [22] V. Smidl and A. Quinn, *The variational Bayes method in signal processing*, ser. Signals and communication technology. Berlin: Springer, 2006.
- [23] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, 2012.
- [24] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, 1996.
- [25] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan, "The clear 2006 evaluation," in *International Evaluation Workshop on Classification of Events, Activities and Relationships*, 2006.