The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

# Motion Estimation Based on Predator/Prey Vision

D. van der Lijn, G.A.D. Lopes and R. Babuška

***Abstract*— We present an unscented Kalman filter based state estimator for a fast moving rigid body (such as a mobile robot) endowed with two video cameras. We focus on forward velocity estimation towards the computation of standard energy cost functions for legged locomotion. Points are chosen as image features and the model of each camera is based on the traditional pinhole projection. The resulting filter's state is composed of the rigid body pose and velocities, together with a measure of depth for each tracked point. By taking inspiration from nature's large predatory and grazing mammals eye configuration, we suggest, via simulation results, a solution for the question of finding the best orientation of two cameras, between side and frontal facing, for velocity estimation in a forward moving robot.**

## I. Introduction

Knowledge of full-state information is fundamental for most contemporary control techniques. In mobile robotics, in particular, rigid body full-state information enables navigation, manipulation, and motion optimization [1]. Full-state observers are typically implemented by means of sensor fusion data [2]–[6], each compromising in terms of complexity, accuracy, and cost.

A very useful non-dimensional parameter, denoted *specific resistance* $\epsilon$, accepted as the standard measure for energy efficiency in both biology and in robotics, was originally proposed by Gabrielli and von Karman in 1950 [7]:

$$\epsilon = \frac{P}{mgv},$$

where $P$ is the average power expenditure, $m$ is the total mass, $g$ is the gravitational acceleration, and $v$ is the forward velocity. The specific resistance has been calculated in the literature for many types of animals, including mammals, arthropods, reptiles [8], and machines, including cars, airplanes, bicycles, etc. [7]. For legged robots, this measure is typically used as a cost function that is minimized towards more efficient walking or running gaits [1]. The fundamental parameters required for the computation of the specific resistance are the average power consumption, readily measurable in hardware for robot systems that use electrical motors, and the forward velocity. Unfortunately, without implementing a sensor fusion observer, there exists no standard hardware-based linear velocity sensor that can be purchased for a legged robot. Moreover, traditional off-the-shelf sensors bear various types of limitations that complicate the development of linear velocity estimators. Dead reckoning, used in wheeled robots, suffers inevitably from slipping and requires elaborate implementations for legged robots [9]. Modern solid-state Inertial Measurement Units (IMU) [10] supply angular velocities and linear accelerations. Linear velocity information must be obtained via filtering and can result in large drifting errors after a few minutes [6]. GPS sensors suffer from low update rate and lack of indoor usability that hinder their deployment on fast walking robots.

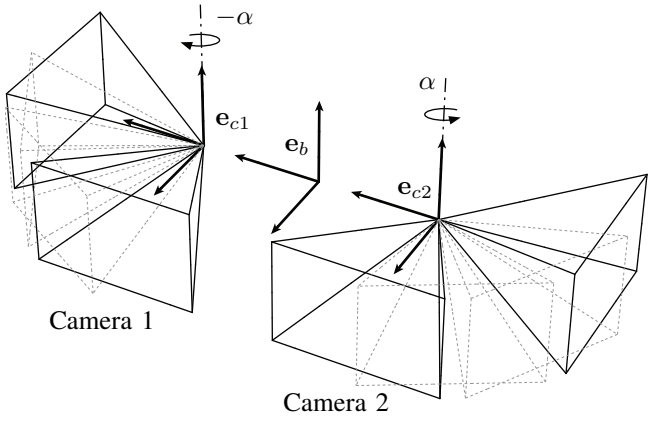



Fig. 1. Configuration of two cameras that can swivel $\alpha$ degrees from facing forward to facing sideways. The pyramid shapes represent the field of view aperture of the cameras. The frames $\mathbf{e}_b$, $\mathbf{e}_{c1}$, and $\mathbf{e}_{c2}$ are attached to the body, camera 1 and camera 2 respectively.

Video cameras offer an alternative to traditional proprioceptive sensors, but due to their high bandwidth throughput and generality, complex algorithms are typically required to reduce the dimensionally of the video data and extract body pose and velocity estimations from it (for a survey on vision-based methods for robot navigation please see [11]). We take this approach in this paper by expanding the work of Chiuso et al. [12] to multiple cameras that can, but do not require to, have overlapping fields of view, with the goal of developing a low-cost, low-complexity velocity estimator for legged robots.

One less explicitly researched topic is how to optimally place various video cameras to improve the estimation task at hand. Chen et al. [13] addressed the problem of automatic camera motion for locating features of interest in an object and Burschka et al. [14] addressed the dual problem of optimal landmark configuration. In this paper we take inspiration from nature to seek for the configuration of two cameras that minimize the estimation error of the forward velocity of a legged robot. Typical predatory mammals have eyes facing forward. This evolutionary outcome sustain the hypothesis that stereoscopic vision is beneficial for object tracking and

Van der Lijn, Lopes and Babuška are with Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands. dvanderlijn@gmail.com, {g.a.delgadolopes, r.babuska} @tudelft.nl

3D manipulation. Grazing mammals (prey) however, have evolved to possess side facing eyes with wide field of views. This is expected since it is of their advantage to be able to detect predators as fast as possible, which is facilitated by wide field of view eyes. This evolutionary difference in nature poses the question (although not directly related from the behavioral point of view) of what is the best camera configuration (from a specific set of orientations, as illustrated in Figure 1) for forward velocity estimation. We denote this as the *predator/prey camera configuration* hypothesis.

The contributions of this paper are two fold: i) we augment a current filtering technique developed for a single camera to multiple cameras, with additional constraint equations resulting from overlapping fields of view. ii) We elaborate on the Predator/Prey camera configuration hypothesis concluding that side facing cameras improve forward velocity estimation. Section II revises the single camera filter presented in [12], Section III presents the extension to multiple cameras, Section IV formulates the estimation problem from a filtering perspective, and Section V presents the simulations results.

## II. Single camera model

For the remainder of this paper the following assumptions hold:

- A1 A point tracker is available that returns a vector of projections of points in time.
- A2 Points remain in the camera's field of view throughout the entire length of the experiment.
- A3 Points are assumed to be static in the environment.

These assumptions limit the direct applicability of the presented algorithmic extension, but allow for the answering of the predator/prey hypothesis in a straightforward manner. Assumption A1 avoids the computational cost incurred by feature trackers. By choosing the simplest possible feature, a point, we implicitly consider that problem. For our forthcoming experimental implementation we are currently utilizing the Kanade-Lucas-Tomasi point tracker [15], capable of running in real-time in modern hardware. Assumption A2 can be relaxed by running two parallel filters as mentioned in [12]: the first filter's state contains the robot's pose, velocities and depths of tracked points (has we describe in Section III). The purpose of the second filter, whose state consists solely in the depths of points and uses the robot's pose and velocities as parameters, is to allow for new points to have its depth estimated before they are added to the first filter. This avoids transients in the estimation of the robot's state. Points that never go below a pre-defined threshold of the back projection error[1] are not added to the first filter, removing this way undesirable outliers. Points that are lost are simply removed from the filter's states. Finally, assumption A3 can be dealt by moving object detection [16]. Under these assumptions, we consider the agent to be a rigid body endowed with a video camera observing collections of points while in motion. The camera is modeled by the traditional pinhole projection with added radial lens distortion. The model for the measured observations of the points is the composition of three classes of maps: rigid body transformations, projection, and "distortion" maps. We keep emphasis on the first two, by assuming that the lens distortion is known, i.e. the camera is calibrated a priori. Let $p$ be a three-dimensional point described in generic coordinates. Let $\phi : \mathsf{SE}(3) \times \mathbb{R}^3 \to \mathbb{R}^3$ be a rigid body transformation

$$\phi(R, q, p) = Rp + q,$$

with $R$ and $q$ the traditional rotation matrix and translation vector, respectively. The camera pinhole projection model is realized by a projection map $\pi : \mathbb{R}^3 \to \mathbb{R}^2$,

$$\pi(p) = \frac{1}{z} \begin{bmatrix} x \\ y \end{bmatrix},$$

where $x, y, z$ are the coordinates of the vector $p$. The composition of the maps is described graphically by the informal commutative diagram

$$\underset{\text{world}}{p_w} \overset{\phi}{\longleftrightarrow} \underset{\text{camera}}{p_c} \overset{\pi}{\longrightarrow} \underset{\text{plane}}{p_p} \overset{\psi_l}{\longleftrightarrow} \underset{\text{lens}}{p_l} \overset{\psi_s}{\longleftrightarrow} \underset{\text{sensor}}{p_s},$$

were $\psi_s \circ \psi_l = \psi$ describe lens distortions, always present in real camera setups. The full sensor model is described by:

$$p_s = \psi \circ \pi \circ \phi(R, q, p_w) \tag{1}$$

For the ease of notation, for the remainder of the paper we use the simpler "undistorted" camera model[2], where $p_p$ are points in the image projection plane:

$$p_p(t) = \pi \circ \phi(R(t), q(t), p_w). \tag{2}$$

Here we consider that the camera's motion in time is described by the group parameters $(R(t), q(t)) \in \mathsf{SE}(3)$. Without loss of generality, assume that at the initial time instance $t = 0$ the camera frame aligns with the world frame, i.e. $R(0) = I$, $q = [0\ 0\ 0]^T$ implying that $\phi$ is the identity map. Let $\gamma^x, \gamma^y$ be the homogeneous form for 2D points, and $\xi$ be a measure of depth. Points in the camera frame are represented at the initial state $t = 0$ by:

$$p_c(0) = \phi(p_w(0)) = z_w(0) \begin{bmatrix} \frac{x_w(0)}{z_w(0)} \\ \frac{y_w(0)}{z_w(0)} \\ 1 \end{bmatrix} = \xi \begin{bmatrix} \gamma^x \\ \gamma^y \\ 1 \end{bmatrix} \tag{3}$$

In the image plane these reduce to:

$$p_p(0) = \pi(p_c(0)) = \begin{bmatrix} \gamma^x \\ \gamma^y \end{bmatrix} \tag{4}$$

## III. Multi-camera model

We expand on the previous section by considering two cameras with partially overlapping fields of view. We start

[1]The difference from the observed output to the estimated output.

[2]Note that for the real implementation it is useful to include all the maps in the observation's model since noise distributions get "deformed" by each map. If the noise model is known for the camera's CCD sensor and very wide lens are used, then very different filtered results can be obtained using models (1) or (2).

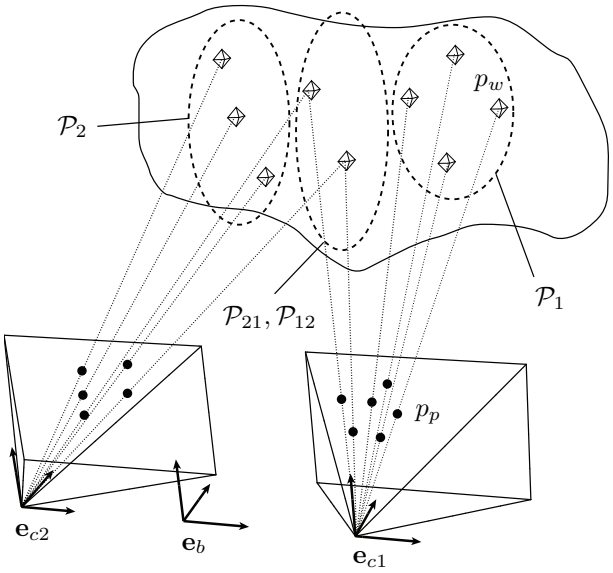



Fig. 2. Dual camera configuration observing collections of points in the environment. The frames $\mathbf{e}_b$, $\mathbf{e}_{c1}$, and $\mathbf{e}_{c2}$ are attached to the body, camera 1 and camera 2 respectively.

by introducing sets of points observed by each camera separately or by both cameras simultaneously. Let:

- $\mathcal{P}_1$ be the set of points visible only by camera 1,
- $\mathcal{P}_2$ be the set of points visible only by camera 2,
- $\mathcal{P}_{12}$ be the set of points visible by both cameras in camera 1 coordinates, and
- $\mathcal{P}_{21}$ be the set of points visible by both cameras in camera 2 coordinates.

Two new assumptions are added for the multi-camera case:

- A4 The correspondence problem is assumed solved for the points in $\mathcal{P}_{12}$ (and $\mathcal{P}_{21}$).
- A5 The relative position of the cameras to the robot's body is known and remains fixed throughout the entire motion.

Assumption A4 sidesteps a potentially computational costly procedure that for a real experimental implementation favors the prey camera configuration: since the field of views are non-overlapping no point correspondence is required. Assumption A5 can be relaxed by including the rigid body transformations for each camera as time-varying parameters in the output equations.

The multi-camera model is similar to the one in Section II, differing by an additional body-to-camera transformation. Let

$$\phi_b(p) = \phi(R(t), q(t), p), \tag{5a}$$
$$\phi_{c1}(p) = \phi(R_{c1}, q_{c1}, p), \tag{5b}$$
$$\phi_{c2}(p) = \phi(R_{c2}, q_{c2}, p). \tag{5c}$$

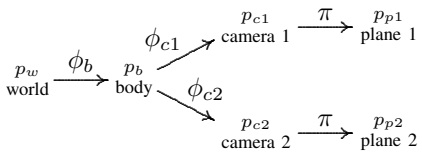
Function (5a) maps points from world to body reference and functions (5b),(5c) map body to cameras 1 and 2 references respectively. For both cameras, the maps $\phi_{c1}, \phi_{c2}$ are assumed to be fixed throughout time. Using the previously defined depth and homogeneous representation, equation (5a) is written as:

$$\begin{aligned}\phi_b(p) &= \phi(R(t), q(t), \xi[\gamma^x \ \gamma^y \ 1]^T) \\ &= \phi_h(R(t), q(t), \xi, \gamma^x, \gamma^y)\end{aligned}$$

The sensor model for two cameras is described by:

$$\begin{aligned}p_{p1} &= \pi \circ \phi_{c1} \circ \phi_b(p_w) \\ p_{p2} &= \pi \circ \phi_{c2} \circ \phi_b(p_w)\end{aligned}$$

The following informal commutative diagram illustrates the composition of the maps:




We utilize again the initial time assumption described in the previous section by considering that the group variables $R$ and $q$ are at the origin for $t = 0$. However, due to the extra body-to-camera transformation, equations (3) and (4) take a different form. Denote by

$$y_1 = p_{p1} = \pi \circ \phi_{c1} \circ \phi_b(p_{w1}), \ p_{w1} \in \mathcal{P}_1$$

the projection of a point observed solely by camera 1. Likewise, denote by $y_2$ the projection of a point observed solely by camera 2, $y_{12}$ observed by both cameras in camera 1 coordinates, and $y_{21}$ observed by both cameras in camera 2 coordinates. The equation of $y_{12}$ at initial time is described by:

$$\begin{aligned}y_{12}(0) = \pi \circ \phi_{c1} \circ \phi_h \big(&R_{c1}^{-1} R(0), q(0) - R_{c1}^{-1} q_{c1}, \\ &\xi_{12}, \gamma_{12}^x, \gamma_{12}^y\big)\end{aligned} \tag{6}$$

The "shifting" terms $R_{c1}^{-1}$ and $-R_{c1}^{-1} q_{c1}$ are added such that

$$\begin{aligned}y_{12}(0) &= \pi \circ \phi_{c1} \left( \xi_{12} R_{c1}^{-1} \begin{bmatrix} \gamma_{12}^x \\ \gamma_{12}^y \\ 1 \end{bmatrix} - R_{c1}^{-1} q_{c1} \right) \\ &= \pi \left( \xi_{12} \begin{bmatrix} \gamma_{12}^x \\ \gamma_{12}^y \\ 1 \end{bmatrix} \right) = \begin{bmatrix} \gamma_{12}^x \\ \gamma_{12}^y \end{bmatrix}\end{aligned}$$

For the equation of $y_{21}$ one uses the same point representation as in (6), resulting in a nonlinear expression in terms of the point parameters $\xi_{12}, \gamma_{12}^x, \gamma_{12}^y$:

$$\begin{aligned}y_{21}(0) = \pi \circ \phi_{c2} \circ \phi_{c1}^{-1} \circ \phi_{c1} \circ \phi_h \big(&R_{c1}^{-1} R(0), \\ &q(0) - R_{c1}^{-1} q_{c1}, \xi_{12}, \gamma_{12}^x, \gamma_{12}^y\big) \\ = \zeta(&\xi_{12}, \gamma_{12}^x, \gamma_{12}^y)\end{aligned}$$

The extra inverse map $\phi_{c1}^{-1}$ in the equation above arrises from the fact that points are represented first in camera 1 coordinates, and then are translated back to camera 2 coordinates by the map $\phi_{c2} \circ \phi_{c1}^{-1}$. Following the same reasoning,

projection points uniquely observed by either camera 1 or 2 at $t = 0$ are represented by:

$$y_1(0) = \pi \circ \phi_{c1} \circ \phi_h \left(R_{c1}^{-1} R(0), \right. \\ \left. q(0) - R_{c1}^{-1} q_{c1}, \xi_1, \gamma_1^x, \gamma_1^y\right) = \begin{bmatrix} \gamma_1^x \\ \gamma_1^y \end{bmatrix}$$

$$y_2(0) = \pi \circ \phi_{c2} \circ \phi_h \left(R_{c2}^{-1} R(0), \right. \\ \left. q(0) - R_{c2}^{-1} q_{c2}, \xi_2, \gamma_2^x, \gamma_2^y\right) = \begin{bmatrix} \gamma_2^x \\ \gamma_2^y \end{bmatrix}$$

Let $\phi_1, \phi_2$ be defined by:

$$\phi_1(R, q, \xi, \gamma^x, \gamma^y) = \phi_h \left(R_{c1}^{-1} R, q - R_{c1}^{-1} q_{c1}, \xi, \gamma^x, \gamma^y\right) \\ \phi_2(R, q, \xi, \gamma^x, \gamma^y) = \phi_h \left(R_{c2}^{-1} R, q - R_{c2}^{-1} q_{c2}, \xi, \gamma^x, \gamma^y\right)$$

The equations of projected points for an arbitrary time $t$ are:

$$\begin{aligned} y_1(t) &= \pi \circ \phi_{c1} \circ \phi_1(R(t), q(t), \xi_1, \gamma_1^x, \gamma_1^y) \\ y_2(t) &= \pi \circ \phi_{c2} \circ \phi_2(R(t), q(t), \xi_2, \gamma_2^x, \gamma_2^y) \\ y_{12}(t) &= \pi \circ \phi_{c1} \circ \phi_1(R(t), q(t), \xi_{12}, \gamma_{12}^x, \gamma_{12}^y) \\ y_{21}(t) &= \pi \circ \phi_{c2} \circ \phi_1(R(t), q(t), \xi_{12}, \gamma_{12}^x, \gamma_{12}^y) \end{aligned}$$

which simplify to:

$$y_1(t) = \pi \circ \phi_h(R(t), q(t), \xi_1, \gamma_1^x, \gamma_1^y) \tag{7a}$$
$$y_2(t) = \pi \circ \phi_h(R(t), q(t), \xi_2, \gamma_2^x, \gamma_2^y) \tag{7b}$$
$$y_{12}(t) = \pi \circ \phi_h(R(t), q(t), \xi_{12}, \gamma_{12}^x, \gamma_{12}^y) \tag{7c}$$
$$y_{21}(t) = \pi \circ \phi_{c2} \circ \phi_1(R(t), q(t), \xi_{12}, \gamma_{12}^x, \gamma_{12}^y) \tag{7d}$$

Although not formally demonstrated here, the state is observable when motion is present (up to some pathological cases [17]) due to the known relative position between both cameras, represented by $R_{c1}, q_{c1}, R_{c2}, q_{c2}$. These known quantities, together with the assumption of the world frame matching the body frame for the initial condition, eliminate the requirement of fixing some of the state variables as in Chiuso's [12] implementation.

## IV. Filter design

The estimation of the robot velocity can be transformed into a filtering problem by utilizing the previously described multi-camera model. For this implementation we choose the unscented Kalman filter [18] since it avoids the computation of the Jacobians of the process and observation models as in the extended Kalman filter case. This is useful for the upcoming experimental implementation where the dimension of the state in the filter changes dynamically as new points are tracked or others leave the camera's field of views. We start by vectorizing the rotation matrix $R$ into $\Omega$ by utilizing the standard inverse Rodrigue's formula:

$$\Omega = \mathrm{Rod}^{-1}(R)$$

Since there is no a priori knowledge of the absolute location of the observed points in the world, these are added to the state being estimated. Let $n = |\mathcal{P}_1|,\ m = |\mathcal{P}_2|$ and $l = |\mathcal{P}_{12}|$. For each point $\xi_{i,j}, \gamma_{i,j}^x, \gamma_{i,j}^y$ indexed by $j$, with $i \in \{1, 2, 12\}$ belonging to the sets $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_{12}$ let:

$$\Xi_1 = [\xi_{1,1} \ \dots \ \xi_{1,n}] \tag{8a}$$
$$\Gamma_1^x = [\gamma_{1,1}^x \ \dots \ \gamma_{1,n}^x] \tag{8b}$$
$$\vdots$$
$$\Gamma_{12}^y = [\gamma_{12,1}^y \ \dots \ \gamma_{12,l}^y] \tag{8c}$$

The full state x of dimension $12 + 3(n + m + l)$ is described by:

$$\mathrm{x} = \left[q^T \ \Omega^T \ v^T \ \omega^T \ \Xi_1 \ \Gamma_1^x \ \Gamma_1^y \ \Xi_2 \ \Gamma_2^x \ \Gamma_2^y \ \Xi_{12} \ \Gamma_{12}^x \ \Gamma_{12}^y\right]^T,$$

where $v$ represents linear velocity and $\omega$ angular velocity, with its associated dynamical model

$$\mathrm{x}(k+1) = f(\mathrm{x}(k)) + \sigma_x(k), \tag{9}$$

and process noise assumed zero mean Gaussian with covariant matrix $Q$:

$$\sigma_x(k) \sim \mathcal{N}(0, Q).$$

Equation (9) is described in detail by:

$$q(k+1) = e^{T_s \tilde{\omega}(k)} q(k) + T_s v(k) + \sigma_q(k) \tag{10a}$$
$$\Omega(k+1) = \mathrm{Rod}^{-1}(e^{T_s \tilde{\omega}(k)} e^{\tilde{\Omega}(k)}) + \sigma_\Omega(k) \tag{10b}$$
$$v(k+1) = v(k) + \sigma_v(k) \tag{10c}$$
$$\omega(k+1) = \omega(k) + \sigma_\omega(k) \tag{10d}$$
$$\Xi_1(k+1) = \Xi_1(k) + \sigma_{\Xi,1}(k) \tag{10e}$$
$$\Gamma_1^x(k+1) = \Gamma_1^x(k) + \sigma_{\Gamma,1}(k) \tag{10f}$$
$$\vdots$$
$$\Gamma_{12}^y(k+1) = \Gamma_{12}^y(k) + \sigma_{\Gamma,\cdot}(k) \tag{10g}$$

where "$\sim$" is the standard skew operator ($\tilde{a}b = a \times b$). The parameter $T_s$ is the sampling time and equations (10a)-(10d) are obtained by integrating constant accelerations over a time step $T_s$.

This model can be reduced by assuming a perfect measure of the 2D point coordinates $\Gamma_1^x, \dots, \Gamma_{12}^y$ in the image plane at the initial instance. These do not change throughout the motion since they are defined in the fixed world frame that matches the camera frame for the initial instance. The reduced state of dimension $12 + n + m + l$ is then:

$$\mathrm{x} = \left[q^T \ \Omega^T \ v^T \ \omega^T \ \Xi_1 \ \Xi_2 \ \Xi_{12}\right]^T, \tag{11}$$

The observations vector of dimension $2(n + m) + 4l$ is represented by:

$$\mathrm{y} = \left[\mathrm{y}_1^T \ \mathrm{y}_2^T \ \mathrm{y}_{12}^T \ \mathrm{y}_{21}^T\right]^T,$$

with the associated observation equations

$$\mathrm{y}(k) = h(\mathrm{x}(k)) + \sigma_y(k), \tag{12}$$

where the observation noise $\sigma_y(k)$ is assumed zero mean Gaussian with covariant matrix $R$:

$$\sigma_y(k) \sim \mathcal{N}(0, R).$$

The observation model (12) is described by:

$$\begin{aligned}
\mathrm{y}_1(k) &= \bar{\pi} \circ \bar{\phi}_h \left(\mathrm{Rod}(\Omega(k)), q(k), \Xi_1(k), \Gamma_1^x(k), \Gamma_1^y(k)\right) \\
\mathrm{y}_2(k) &= \bar{\pi} \circ \bar{\phi}_h \left(\mathrm{Rod}(\Omega(k)), q(k), \Xi_2(k), \Gamma_2^x(k), \Gamma_2^y(k)\right) \\
\mathrm{y}_{12}(k) &= \bar{\pi} \circ \bar{\phi}_h \left(\mathrm{Rod}(\Omega(k)), q(k), \Xi_{12}(k), \Gamma_{12}^x(k), \Gamma_{12}^y(k)\right) \\
\mathrm{y}_{21}(k) &= \bar{\pi} \circ \bar{\phi}_{c2} \circ \bar{\phi}_1 \left(\mathrm{Rod}(\Omega(k)), q(k), \right. \\
&\qquad \left. \Xi_{12}(k), \Gamma_{12}^x(k), \Gamma_{12}^y(k)\right), \qquad (13)
\end{aligned}$$

where the maps $\bar{\pi}, \bar{\phi}_h, \bar{\phi}_{c2}$, and $\bar{\phi}_1$ are the multidimensional versions of their counterparts in equations (7a) to (7d). For the simulations presented in this paper the process noise covariance matrix is initialized as a diagonal matrix

$$Q = \mathrm{diag}\left(\sigma_q, \sigma_\Omega, \sigma_v, \sigma_\omega, \sigma_\Xi\right), \qquad (14)$$

where

$$\begin{aligned}
\sigma_q = \sigma_\Omega = \sigma_v = \sigma_\omega &= 10^{-3} I^{3\times 3} \\
\sigma_\Xi &= 10^{-2} I^{(n+m+l)\times(n+m+l)}
\end{aligned}$$

The observations noise covariance matrix is assumed to be

$$R = 10^{-4} I^{(2(n+m)+4l)\times(2(n+m)+4l)} \qquad (15)$$

The dynamical model (9) with the reduced state (11), together with the observation model (12),(13), (using the simplified notations $\mathrm{x}_k = \mathrm{x}(k)$ and $\mathrm{y}_k = \mathrm{y}(k)$), and the corresponding covariant matrices $Q$ and $R$, are utilized in the standard UKF algorithm revisited next:

1. Initialization

$$\hat{\mathrm{x}}_0 = E[\mathrm{x}_0]; \qquad P_0 = E[(\mathrm{x}_0 - \hat{\mathrm{x}}_0)(\mathrm{x}_0 - \hat{\mathrm{x}}_0)^T]$$

2. Calculate $2N+1$ sigma points and weights (with $N = 12 + n + m + l$)

$$\begin{aligned}
\mathcal{X}_{k-1} &= \left[\hat{\mathrm{x}}_{k-1} \quad \hat{\mathrm{x}}_{k-1} + \eta\sqrt{P_{k-1}} \quad \hat{\mathrm{x}}_{k-1} - \eta\sqrt{P_{k-1}}\right] \\
w &= [\lambda/\eta \quad \underbrace{1/(2\eta) \quad \cdots \quad 1/(2\eta)}_{\times 2N}]^T
\end{aligned}$$

3. Prediction step

$$\begin{aligned}
\mathcal{X}_{k|k-1} &= f(\mathcal{X}_{k-1}) \\
\hat{\mathrm{x}}_k^- &= \mathcal{X}_{k|k-1} w \\
P_k^- &= Q + \sum_{i=0}^{2N} w_i \left(\left(\mathcal{X}_{k|k-1}\right)_i - \hat{\mathrm{x}}_k^-\right)\left(\left(\mathcal{X}_{k|k-1}\right)_i - \hat{\mathrm{x}}_k^-\right)^T \\
\mathcal{Y}_{k|k-1} &= h(\mathcal{X}_{k|k-1}) \\
\hat{\mathrm{y}}_k &= \mathcal{Y}_{k|k-1} w
\end{aligned}$$

4. Filtering step

$$\begin{aligned}
P_{yy,k} &= R + \sum_{i=0}^{2N} w_i \left(\left(\mathcal{Y}_{k|k-1}\right)_i - \hat{\mathrm{y}}_k\right)\left(\left(\mathcal{Y}_{k|k-1}\right)_i - \hat{\mathrm{y}}_k\right)^T \\
P_{xy,k} &= \sum_{i=0}^{2N} w_i \left(\left(\mathcal{X}_{k|k-1}\right)_i - \hat{\mathrm{x}}_k^-\right)\left(\left(\mathcal{Y}_{k|k-1}\right)_i - \hat{\mathrm{y}}_k\right)^T \\
K_k &= P_{xy,k} P_{yy,k}^{-1} \\
\hat{\mathrm{x}}_k &= \hat{\mathrm{x}}_k^- + K_k(\mathrm{y}_k - \hat{\mathrm{y}}_k) \\
P_k &= P_k^- - K_k P_{yy,k} K_k^T
\end{aligned}$$

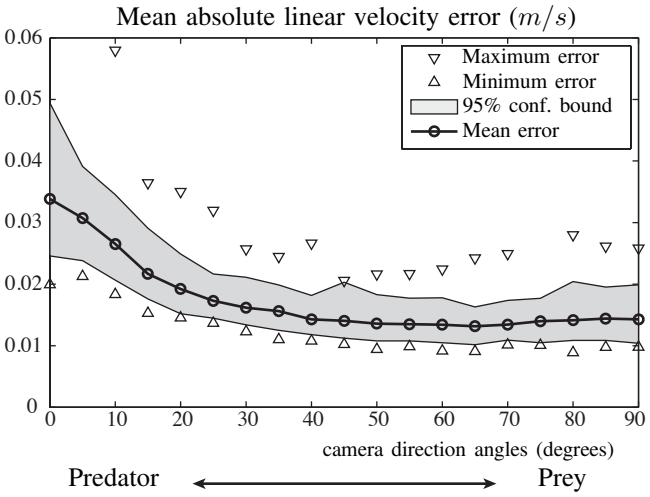



Fig. 3. Mean absolute linear velocity error plots for predator/prey camera configuration. Zero degrees corresponds to forward pointing eyes as in mammal predators, and 90 degrees corresponds to side pointing eyes as in typical grazing mammals.

The parameter $\eta = N + \lambda$, with $\lambda = N(\alpha^2 - 1)$; $10^{-1} \leq \alpha \leq 1$. The notation $(\cdot)_i$ represents the $i$-th column of the enclosed matrix or the $i$-th element of a vector.

## V. Simulations

A simulation environment was designed following assumptions A1-A5. A set of $z$ points is generated according to a uniform distribution in a 3-dimensional box enclosing the agent's desired motion domain. Next, a reference trajectory is produced and a simple controller is designed to follow such trajectory. The solution of the differential equations for the rigid body motion is computed in continuous time and stored. All $z$ points are then projected into the simulated camera plane, and the sets $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_{12}$ are filled with points that verify each appropriate field of view constraints for all time instances. A fixed number of points is then chosen from each set. Figure 3 compiles the results for the mean error in the velocity estimation for various camera orientation angles, averaged over 50 simulations per angle. The mean absolute linear velocity error measure is computed as:

$$\text{error} = \mathrm{mean}_{i,k} |\hat{v}_i(k) - v_i(k)|$$

where $\hat{v}_i(k)$ is the estimated linear velocity for simulation $i$ indexed by time $k$ and $v_i(k)$ is the stored velocity solution. The results suggest that for forward motion, the prey camera configuration fares better. Figures 4 and 5 illustrate sample world motion plots of simulations for predator and prey configurations respectively.

## VI. Conclusions and Future Work

Simulation results suggest that the prey configuration is beneficial for linear velocity estimation when the robot is moving forward. Moreover, in the prey configuration no point correspondence is required as in the predator case, since the

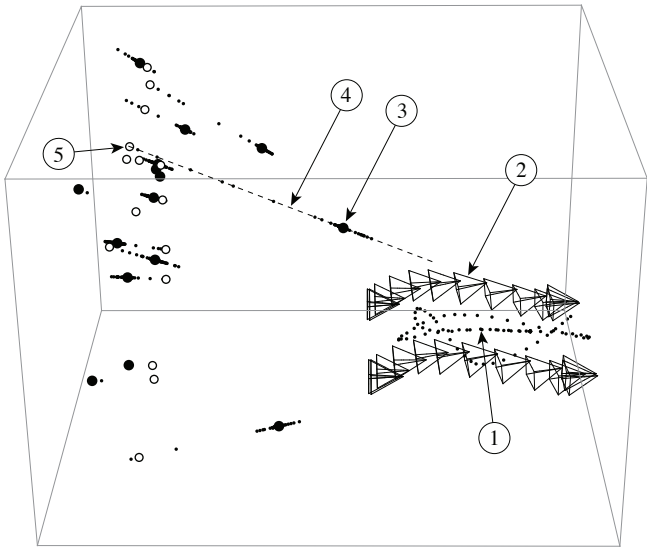



Fig. 4. Predator camera configuration. The trajectory of the rigid body is illustrated by the small black dots (1) that are surrounded by the field of view of each camera, represented by the pyramidal shapes (2). The large black dots (3) represent the real location of the feature points in the environment, the large white dots (5) represent the initial condition estimate of the world points based on unknown depths and the remaining small black points represent the various estimates of depth over time. A point ray is illustrated by the dashed line (4). Only a small number of positions of the camera field of view are plotted for illustration purposes.

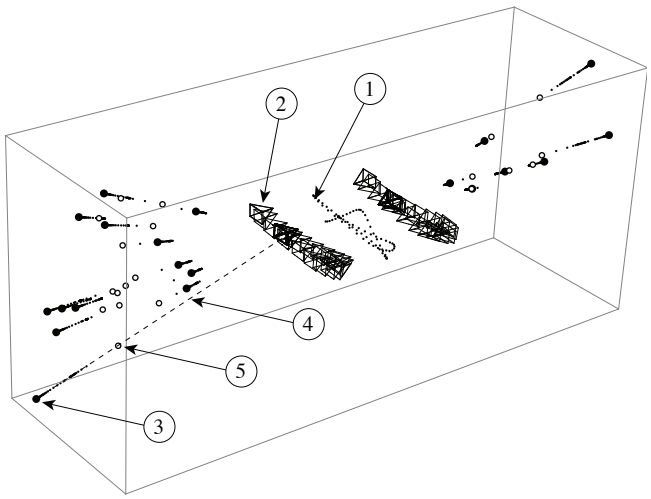



Fig. 5. Prey camera configuration. Properties are the same as in Figure 4.

cameras field of view do not overlap. For the angular velocity estimation no clear optimal camera configuration was found. The simulation results additionally suggest that different types of motion benefit from different camera configurations, so it is useful to dynamically actuate the camera mounts.

The presented unscented Kalman filter for state estimation using two cameras offers a few advantages over a single camera setup. For non-overlapping fields of view, as in the prey configuration, there exists no depth ambiguity as long as the cameras are not mounted in such a way that the focal points coincide. For overlapping fields of view, points observed by two cameras have depth convergence without motion. The filtering technique utilized is invariant to the geometry of the environment, assuming a rich could of features. If the point tracking algorithms can be efficiently implemented, then the additional Kalman filter does not dramatically increase the computation complexity.

Parallels to this formulation can be found in nature, in particular in insect vision, where compound eyes track feature contrasts in a similar way to tracking points. We are currently working on the experimental validation on a legged robotic platform fitted with two synchronized cameras mounted on servo motors for dynamic actuation.

## REFERENCES

[1] J. Weingarten, G. Lopes, M. Buehler, R. Groff, and D. Koditschek, "Automated gait adaptation for legged robots," in *in Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 3, 2004, pp. 2153–2158.

[2] S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "A high-integrity imu/gps navigation loop for autonomous land vehicle application," *IEEE Transactions in Robotics and Automation*, vol. 15, no. 3, pp. 572–578, 1999.

[3] P. J. Escamilla-Ambrosio and N. Mort, "A hybrid kalman filter fuzzy logic architecture for multisensor data fusion," in *in Proc. Int. Symp. Intelligent Control*, 2001, pp. 364–369.

[4] R. C. Ren, S. H. Phang, and K. L. Su, "Multilevel multisensor based decision fusion for intelligent animal robot," in *in Proc. IEEE Int. Conf. Robotics and Automation*, vol. 4, 2001, p. 42264231.

[5] M. Abdelrahman and P. Abdelrahman, "Integration of multiple sensor fusion in controlled design," in *in Proc. American Control Conf.*, 2002, pp. 2609–2614.

[6] P.-C. Lin, H. Komsuoglu, and D. Koditschek, "Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 932 – 943, 2006.

[7] G. Gabrielli and T. Von Karman, "What price speed," *Mechanical Engineering*, vol. 72, no. 10, pp. 775–781, 1950.

[8] R. Full, "Animal motility and gravity," *Physiologist*, vol. 34, no. 1, pp. S15–18, 1991.

[9] G. Roston and E. Krotkov, "Dead reckoning navigation for walking robots," in *in Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, 1992, pp. 607–612.

[10] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions in Robotics and Automation*, vol. 11, no. 3, pp. 328–342, 1995.

[11] G. DeSouza and A. Kak, "Vision for mobile robot navigation: A survey," *IEEE transactions on pattern analysis and machine intelligence*, pp. 237–267, 2002.

[12] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 523–535, 2002.

[13] S. Chen and Y. Li, "Automatic sensor placement for model-based robot vision," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 393–408, 2004.

[14] D. Burschka, J. Geiman, and G. Hager, "Optimal landmark configuration for vision-based control of mobile robots," in *in Proc of IEEE Int. Conf. on Robotics and Automation*, vol. 3, 2003, pp. 3917–3922.

[15] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon University Technical Report CMU-CS-91-132, Tech. Rep., 1991.

[16] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.

[17] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley, "Robust 6DOF motion estimation for non-overlapping, multi-camera systems," in *IEEE Workshop on Applications of Computer Vision*, 2008, pp. 1–8.

[18] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, vol. 3, 1997, p. 26.