Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 11441–11446

Shutter delay estimation for a low-cost stand-alone visual tracking system ⋆

Arnold Pretorius * **Edward Boje** **

* *Electrical Engineering Department, University of Cape Town, Cape Town, South Africa (e-mail: prtarn001@myuct.ac.za).*

** *Electrical Engineering Department, University of Cape Town, Cape Town, South Africa (e-mail: edward.boje@uct.ac.za).*

Abstract: A cost-efficient alternative to expensive off-the-shelf camera systems is developed with the aim of tracking fast, dynamic manoeuvres of rigid-body objects within a fixed space. Multiple low-cost camera modules, which possess on-board image processing, are statically placed in a pre-defined space, and are used to simultaneously capture visual information of a moving object. This information is used in a stand-alone extended Kalman filter which estimates translation, pose, and the respective derivatives. Additionally, the differential shutter exposure time between cameras is estimated to avoid the need for camera synchronisation. The filter is validated experimentally and shown to give reliable state information with sub-centimetre accuracy.

Keywords: Guidance navigation and control, information and sensor fusion, estimation and filtering, visual tracking, extended Kalman filter.

1. INTRODUCTION

In the context of tracking rigid-body objects, many solutions exist with varying accuracies and applicabilities (Gupte et al., 2012). A common method is to determine the translation and/or attitude using various combinations of on-board sensors, such as GPS modules (Nemra and Aouf, 2009), stereo cameras (Fu et al., 2015), laser range-finding (Gao and Shen, 2016), inertial measurement units (Leishman et al., 2014), and RGB-D camera modules (Huang et al., 2017). These solutions have the benefit of being portable and possessing negligible sensor latencies, but generally have low accuracy, as well as conditional state observability (Omari and Ducard, 2013).

On the other hand, off-board visual tracking solutions can give unconditional state information in a predefined operational volume (Richards, 1999). Off-the-shelf camera systems are most commonly used to give high precision state estimation and several boast sub-millimetre accuracy at high frame rates (Windolf et al., 2008). For example, high end Vicon camera systems can perform feature estimation at over 200 Hz with HD resolution (Vicon Motion Capture Systems Ltd., 2016). However, such a set-up can easily cost over $100000, which is not feasible for many applications.

Low-cost alternatives to expensive motion capture systems have also been investigated by several authors. (Achtelik et al., 2009) used two cameras, as well as IMU information, to localise a UAV with distinct hue-based features. Similarly, work in (Reddi and Boje, 2014) was able to achieve sub-centimetre accuracy object tracking using PSEye cameras, which included IMU and ultrasonic data fusion. State estimation using static RGB-D cameras has also yielded promising results. In addition to visual 2D information, the feature depth is also estimated using a laser projector and CMOS sensor. These camera modules however generally have a limited depth range and frame-rate (Schmitz et al., 2014).

The work documented in this paper seeks to design a static stand-alone visual tracking system which has satisfactory performance relative to the object to be tracked, whilst still being affordable. Several commercially available Pixy camera modules (Charmed Labs, 2016) are used. Each camera costs $69 and can give multiple feature locations at 50 frames per second. The information from these cameras is used in an extended Kalman filter (EKF), which estimates the pose and translation of a rigid-body object marked with multiple infra-red (IR) features. Additionally, the uncertain camera shuttering times between cameras is estimated to avoid the need to synchronise the camera exposure times.

2. CAMERA MODEL

2.1 Generalised camera model

The generalised continuous-time camera model (Hartley and Zisserman, 2003) for a camera is

$$w^c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}^c = \mathbf{C}^c \begin{bmatrix} \mathbf{b}^i \\ 1 \end{bmatrix}, \tag{1}$$

where $[u^c \ v^c]^T$ is the normalised and undistorted x^c-y^c location measurement of an inertial-frame feature $\mathbf{b}^i$ after projection into camera-frame c. w^c is the feature's metric depth measured in the camera-frame, and $\mathbf{C}^c$ is the

⋆ This research was supported by the South African National Research Foundation under Grant 81148 and by the University of Cape Town.

2405-8963 © 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.1813

constant extrinsic matrix, specific to the camera, which projects $\mathbf{b}^i$ into the camera-frame. Equation (1) can be rewritten, without loss of generality, as

$$\left(\mathbf{c_3}^c \begin{bmatrix}\mathbf{b}^i \\ 1\end{bmatrix}\right)\begin{bmatrix}u \\ v\end{bmatrix}^c = \mathbf{c_{1:2}}^c \begin{bmatrix}\mathbf{b}^i \\ 1\end{bmatrix}, \tag{2}$$

where $\mathbf{c_r}^c$ describes a matrix made up of row(s) $\mathbf{r}$ of $\mathbf{C}^c$.

2.2 Unsynchronised camera model

In the case that multiple cameras are used for 3D triangulation, the various frames need to be synchronised such that the camera shutters expose the scene at the same time. Without synchronisation, each camera's feature location will vary as a function of the differential shutter time and the feature's dynamics. With reference to Fig. 1, camera 1 observe's a feature's inertial-frame location $\mathbf{b}^i(t)$ at time t and is interpreted as the ray $\mathbf{r}_1(t)$. Given a relative shutter delay of τ_2 seconds, camera 2 observes the same feature at $\mathbf{b}^i(t+\tau_2)$ with the ray $\mathbf{r}_2(t+\tau_2)$ instead of $\mathbf{r}_2(t)$. The estimated feature location would therefore incorrectly be at $\hat{\mathbf{b}}^i$. Explicitly, a camera k, that is exposed

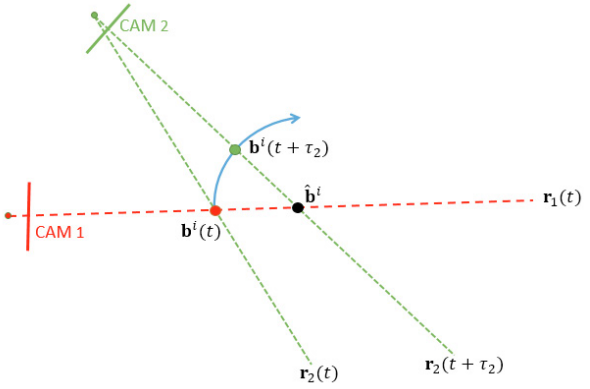

Fig. 1. Effect of differential shutter times for two cameras when triangulating a feature's inertial-frame position. The blue arrow shows the feature's trajectory

τ_k seconds before/after an arbitrarily chosen reference camera will capture information which corresponds to an inertial-frame position, from Taylor series expansion, of

$$\tilde{\mathbf{b}}^i_k = \mathbf{b}^i + \dot{\mathbf{b}}^i\tau_k + \frac{\ddot{\mathbf{b}}^i}{2}\tau_k^2 + ..., \tag{3}$$

where $\mathbf{b}^i$ is the reference camera's measurement. Note that τ_k is bounded in principle by the frame rate of the camera and as such cannot be outside ± 1 frame. With a sufficiently high camera frame rate, relative to the tracked object's motion bandwidth, (3) can be suitably approximated up to its first order. Following from this, the adjusted equation for camera k can be written as

$$\left(\mathbf{c_3}^c_k \begin{bmatrix}\mathbf{b}^i + \dot{\mathbf{b}}^i\tau_k \\ 1\end{bmatrix}\right)\begin{bmatrix}u \\ v\end{bmatrix}^c_k = \mathbf{c_{1:2}}^c_k \begin{bmatrix}\mathbf{b}^i + \dot{\mathbf{b}}^i\tau_k \\ 1\end{bmatrix}. \tag{4}$$

Note that any camera can be chosen as the reference camera and will result in positive and/or negative shutter delay depending on the choice.

3. RESOLVING POSITION AND ORIENTATION

In order to estimate the inertial-frame position and orientation of a rigid-body object, at least three features, placed at known positions in the object's body-frame, need to be localisable. Feature m's inertial-frame position $\mathbf{b}^i_m$ can be related to its constant location in the body-frame $\mathbf{b}^b_m$ as

$$\mathbf{b}^i_m = \mathbf{p}^i + \mathbf{R}^i\mathbf{b}^b_m, \tag{5}$$

where $\mathbf{p}^i$ is the location of the object's body-frame origin in the inertial-frame, $\mathbf{R}^i = \mathbf{R}(\mathbf{q}^i)$ is the rotation matrix of the object, and $\mathbf{q}^i = \begin{bmatrix}q^i_o & q^i_x & q^i_y & q^i_z\end{bmatrix}^T$ is the inertial-frame quaternion. For more information on quaternions, see Diebel (2006). $\mathbf{R}^i$ can be written explicitly as

$$\mathbf{R}^i = \begin{bmatrix}1-2(q_y^2+q_z^2) & 2(q_xq_y - q_oq_z) & 2(q_xq_z + q_oq_y) \\ 2(q_xq_y + q_oq_z) & 1-2(q_x^2+q_z^2) & 2(q_yq_z - q_oq_x) \\ 2(q_xq_z - q_oq_y) & 2(q_yq_z + q_oq_x) & 1-2(q_x^2+q_y^2)\end{bmatrix}^i. \tag{6}$$

Equation (5) can be differentiated with respect to time to obtain the velocity

$$\dot{\mathbf{b}}^i_m = \dot{\mathbf{p}}^i + \dot{\mathbf{R}}^i\mathbf{b}^b_m = \dot{\mathbf{p}}^i + \mathbf{R}^i\boldsymbol{\omega}^b \times \mathbf{b}^b_m, \tag{7}$$

where $\boldsymbol{\omega}^b = \begin{bmatrix}\omega^b_x & \omega^b_y & \omega^b_z\end{bmatrix}^T$ is the body-frame rotational rate. Equation (5) and (7) can then be used to populate (4), which makes the camera equation a function of the state variables of interest.

4. HARDWARE SET-UP

Our motion capture system uses five Pixy camera modules, spaced and located appropriately to give a redundant capture volume of 4×5×3.5 m. The camera modules have on-board processors and are able to output multiple features' raw pixel locations at 50 Hz, with a resolution of 320×200 (Charmed Labs, 2016). The cameras communicate with a base station PC using UART at a baudrate of 460800. A visible light filter is placed between the camera lense and charge-coupled device (CCD) to pass IR light, whilst attenuating most of the visible light. IR feature tracking is chosen over hue-based feature tracking as it is more robust under nominal lighting conditions. Three table tennis balls, fitted with IR LEDs inside, are used as the features, and are placed on the object to be tracked. The cameras are calibrated using open-source Matlab software (Bouguet, 2015). With this, the camera intrinsics, extrinsics, and distortion coefficients can be found.

5. EXTENDED KALMAN FILTER STRUCTURE

5.1 State equations

The six-degree-of-freedom motion of the object to be tracked can be described by the following state vector

$$\mathbf{x} = \begin{bmatrix}\mathbf{p}^{iT} & \mathbf{v}^{iT} & \mathbf{q}^{iT} & \boldsymbol{\omega}^{bT} & \boldsymbol{\tau}^T\end{bmatrix}^T, \tag{8}$$

where $\mathbf{p}^i$ and $\mathbf{v}^i$ are the object's inertial-frame position and velocity respectively, $\mathbf{q}^i$ is the quaternion describing the object's orientation, $\boldsymbol{\omega}^b$ is the body-frame rotational rate, and $\boldsymbol{\tau} = \begin{bmatrix}\tau_2 & \tau_3 & \tau_4 & \tau_5\end{bmatrix}^T$ is the shutter delay vector containing each camera's shutter delay relative to camera 1. For sake of clarity, superscripts indicating reference frames will be omitted from this point onwards, unless explicitly needed. $\mathbf{v}$ and $\boldsymbol{\omega}$ are driven by zero-mean noise signals $\mathbf{n}_v$ and $\mathbf{n}_\omega$ respectively, chosen so as to match the dynamic capabilities of the object to be tracked. Similarly,

the shutter delay vector is modelled as a random walk state, driven by zero-mean noise term $\mathbf{n}_\tau$. The discrete input vector at sample i, containing the aforementioned noise vectors, is

$$\mathbf{u}_i = \begin{bmatrix}\mathbf{n}_v{}^T & \mathbf{n}_\omega{}^T & \mathbf{n}_\tau{}^T\end{bmatrix}_i^T, \tag{9}$$

with corresponding standard deviations of $\boldsymbol{\sigma}_v, \boldsymbol{\sigma}_\omega$, and $\boldsymbol{\sigma}_\tau$ respectively. The discrete time difference equations for a time step of Δt follow as

$$\mathbf{p}_{i+1} = \mathbf{p}_i + \Delta t \mathbf{v}_i, \tag{10}$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \Delta t \mathbf{n}_{vi}, \tag{11}$$

$$\mathbf{q}_{i+1} = \left(\mathbf{I}^{4\times4} + \frac{\Delta t}{2}\lfloor\boldsymbol{\omega}_i\rfloor_\times^{4\times4}\right)\mathbf{q}_i, \tag{12}$$

$$\boldsymbol{\omega}_{i+1} = \boldsymbol{\omega}_i + \Delta t \mathbf{n}_{\omega i}, \tag{13}$$

$$\boldsymbol{\tau}_{i+1} = \boldsymbol{\tau}_i + \Delta t \mathbf{n}_{\tau i}, \tag{14}$$

where $\lfloor\boldsymbol{\omega}_i\rfloor_\times^{4\times4}$ is the 4×4 skew symmetric matrix

$$\lfloor\boldsymbol{\omega}_i\rfloor_\times^{4\times4} = \begin{bmatrix}0 & -\omega_x & -\omega_y & -\omega_z\\ \omega_x & 0 & \omega_z & -\omega_y\\ \omega_y & -\omega_z & 0 & \omega_x\\ \omega_z & \omega_y & -\omega_x & 0\end{bmatrix}_i. \tag{15}$$

With reference to (8) and (9), the linearised state difference equation is

$$\Delta\mathbf{x}_{i+1} = \mathbf{A}_i\Delta\mathbf{x}_i + \mathbf{L}_i\Delta\mathbf{u}_i. \tag{16}$$

The linearised state transition matrix is

$$\mathbf{A}_i^{17\times17} = \begin{bmatrix}\mathbf{I}^{3\times3} & \Delta t\mathbf{I}^{3\times3} & \mathbf{O}^{3\times4} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4}\\ \mathbf{O}^{3\times3} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4}\\ \mathbf{O}^{4\times3} & \mathbf{O}^{4\times3} & \mathbf{A}_q^q & \mathbf{A}_w^q & \mathbf{O}^{4\times4}\\ \mathbf{O}^{3\times3} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4} & \mathbf{I}^{3\times3} & \mathbf{O}^{3\times4}\\ \mathbf{O}^{4\times3} & \mathbf{O}^{4\times3} & \mathbf{O}^{3\times4} & \mathbf{O}^{4\times3} & \mathbf{I}^{4\times4}\end{bmatrix}_i, \tag{17}$$

where $\mathbf{A}_q^q = \left.\frac{\partial\mathbf{q}_{i+1}}{\partial\mathbf{q}_i}\right|_{\hat{\mathbf{x}}_{i|i-1}}$ and $\mathbf{A}_w^q = \left.\frac{\partial\mathbf{q}_{i+1}}{\partial\boldsymbol{\omega}_i}\right|_{\hat{\mathbf{x}}_{i|i-1}}$. The input noise covariance is $\mathbf{Q}_i^{10\times10} = diag\{\boldsymbol{\sigma}_v^2, \boldsymbol{\sigma}_\omega^2, \boldsymbol{\sigma}_\tau^2\}$, and the state noise Jacobian, which maps the input noise vector into the corresponding state variables, is

$$\mathbf{L}_i^{17\times10} = \begin{bmatrix}\mathbf{O}^{3\times3} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4}\\ \Delta t\mathbf{I}^{3\times3} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4}\\ \mathbf{O}^{4\times3} & \mathbf{O}^{4\times3} & \mathbf{O}^{4\times4}\\ \mathbf{O}^{3\times3} & \Delta t\mathbf{I}^{3\times3} & \mathbf{O}^{3\times4}\\ \mathbf{O}^{4\times3} & \mathbf{O}^{4\times3} & \Delta t\mathbf{I}^{4\times4}\end{bmatrix}. \tag{18}$$

5.2 Output equations

The output equation for a particular camera comprises of all the detected x^c-y^c feature locations. Each feature has two output equations, which means that given n features, each camera will have at most $2n$ valid equations. With reference to (2) and (5), the output measurement equation and output estimate respectively, for camera 1 and feature m, is

$$(\mathbf{y}_{1m})_i = \left(\mathbf{c}_{\mathbf{3}1}\begin{bmatrix}\hat{\mathbf{p}}_{i|i-1} + \mathbf{R}(\hat{\mathbf{q}}_{i|i-1})\mathbf{b}_m^b\\ 1\end{bmatrix}\right)\begin{bmatrix}u_m + n_x\\ v_m + n_y\end{bmatrix}_i \tag{19}$$

$$(\hat{\mathbf{y}}_{1m})_i = \mathbf{c}_{\mathbf{1:2}1}\begin{bmatrix}\hat{\mathbf{p}}_{i|i-1} + \mathbf{R}(\hat{\mathbf{q}}_{i|i-1})\mathbf{b}_m^b\\ 1\end{bmatrix}, \tag{20}$$

where n_x and n_y are the uncorrelated camera-frame noise values common across all cameras and features, with corresponding standard deviations of σ_x, and σ_y. Note that all shutter times are relative to camera 1, making camera 1 appear to not have shutter delay. With reference to (4), (5) and (7), the measurement equation and output estimate for cameras $k > 1$ and feature m can be written as

$$(\mathbf{y}_{km})_i = \left(\mathbf{c}_{\mathbf{3}k}\begin{bmatrix}\tilde{\mathbf{x}}_i\\ 1\end{bmatrix}\right)\begin{bmatrix}u_m + n_x\\ v_m + n_y\end{bmatrix}_i, \tag{21}$$

$$(\hat{\mathbf{y}}_{km})_i = \mathbf{c}_{\mathbf{1:2}k}\begin{bmatrix}\tilde{\mathbf{x}}_i\\ 1\end{bmatrix}, \tag{22}$$

where

$$\begin{aligned}\tilde{\mathbf{x}}_i = \ & \hat{\mathbf{p}}_{i|i-1} + \hat{\mathbf{v}}_{i|i-1}\hat{\tau}_{ki|i-1} + \\ & \hat{\mathbf{R}}_{i|i-1}\left(\mathbf{b}_m^b - \lfloor\mathbf{b}_m^b\rfloor_\times^{3\times3}\hat{\boldsymbol{\omega}}_{i|i-1}\hat{\tau}_{ki|i-1}\right),\end{aligned} \tag{23}$$

and

$$\lfloor\mathbf{b}_m^b\rfloor_\times^{3\times3} = \begin{bmatrix}0 & -b_z & b_y\\ b_z & 0 & -b_x\\ -b_y & b_x & 0\end{bmatrix}_m^b. \tag{24}$$

The normalised pixel noise values n_x and n_y are modelled based on the quantisation effect from the cameras having finite resolution, as well as the pixel re-projection error given from camera calibration. The output measurement covariance follows as $\mathbf{V}^{30\times30} = diag\{\sigma_x^2, \sigma_y^2, ..., \sigma_x^2, \sigma_y^2\}$. The output measurement equations of (19) and (21) are a function of the state variables, and as such requires finding its sensitivity to the output noise signals. It is described by

$$\begin{aligned}\mathbf{M}^{30\times30} = diag\{&\frac{\partial y_{111}}{\partial n_x}, \frac{\partial y_{112}}{\partial n_y}, \frac{\partial y_{121}}{\partial n_x}, \frac{\partial y_{122}}{\partial n_y}, \dots,\\ &\frac{\partial y_{531}}{\partial n_x}, \frac{\partial y_{532}}{\partial n_y}\},\end{aligned} \tag{25}$$

where y_{kmr} is row r of the output measurement equation for camera k and feature m. Lastly, the observation matrix $\mathbf{H}^{30\times17}$ is required to populate the Kalman gain and error covariance. The reader is spared the derivation as it is lengthy, but easily solved using symbolic tool software. For sake of completeness, the projection and update equations are (Brown and Hwang, 1997)

$$\hat{\mathbf{x}}_{i|i-1} = \mathbf{f}(\hat{\mathbf{x}}_{i-1|i-1}), \tag{26}$$

$$\hat{\mathbf{y}}_i = \mathbf{h}(\hat{\mathbf{x}}_{i|i-1}), \tag{27}$$

$$\mathbf{P}_{i|i-1} = \mathbf{A}_i\mathbf{P}_{i-1|i-1}\mathbf{A}_i^T + \mathbf{L}_i\mathbf{Q}_i\mathbf{L}_i^T, \tag{28}$$

$$\mathbf{K}_i = \mathbf{P}_{i|i-1}\mathbf{H}_i^T\left[\mathbf{H}_i\mathbf{P}_{i|i-1}\mathbf{H}_i^T + \mathbf{M}_i\mathbf{V}_i\mathbf{M}_i^T\right]^{-1}, \tag{29}$$

$$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + \mathbf{K}_i\left[\mathbf{y}_i - \hat{\mathbf{y}}_i\right], \tag{30}$$

$$\mathbf{P}_{i|i} = \left[\mathbf{I}^{17\times17} - \mathbf{K}_i\mathbf{H}_i\right]\mathbf{P}_{i|i-1}. \tag{31}$$

The quaternion estimate is normalised in every iteration after evaluating (30).

6. FILTER ANALYSIS

The noise terms in (19) and (21) are scaled by the depth estimate. This means that the elements of the output equation covariance $\mathbf{MVM}^T$ corresponding to a specific feature and camera will naturally increase as the feature moves farther from said camera. In other words, the feature depth estimate of each camera acts as a trust metric, which exploits the fact that the 3D triangulation resolution of a camera is inversely proportional to the feature depth. Observability of a specific feature requires at least two cameras to detect the feature. If three features on an object are observable, a plane is defined. This

makes the position and orientation of the object resolvable. On the other hand, the shutter delay estimate for a particular camera is not observable when the detected features either do not move, or move along their respective camera-frame rays. Explicitly, this requires $||\mathbf{c_{1:2}}_k(\hat{\mathbf{v}}_{i|i-1} - \hat{\mathbf{R}}_{i|i-1}\lfloor \mathbf{b}^b_m \rfloor^{3\times3}_{\times}\hat{\boldsymbol{\omega}}_{i|i-1})|| > 0$ for at least one of camera k's features, for τ_k to be observable. This however is not an issue as the shutter delay is expected to be slowly time-varying, if at all. Note that $\mathbf{n}_\tau$ dictates the shutter delay estimate's slew rate, and can be adjusted accordingly.

7. CORRESPONDENCE MATCHING

The choice of using IR features over features of different hues has the disadvantage that each feature is not immediately distinguishable. This requires feature matching across all cameras from one frame to the next. The method of correspondence matching in this paper is as follows: The object to be tracked starts in a predefined position and pose, which places the three features in known inertial-frame locations. These known inertial-frame locations are projected into each camera's frame, and the x^c-y^c distance from each detected feature to the back-projected feature estimate is calculated. For three detected features there will be a 3×3 distance matrix $\mathbf{D}$, where $\mathbf{D}(i,j)$ corresponds to the distance from feature i to back-projected feature prediction j . Feature i corresponds to feature prediction j if row i's minimum is in column j. This minimum pixel distance needs to be below a chosen error threshold in order to validate it as a correct correspondence. Additionally, two feature predictions cannot correspond to a single measured feature, and vice versa. If only two features are detected, the height and width of the feature is analysed, as two features may be lying close enough in the camera-frame to appear as a single large feature. The same is done in the event of only one feature being detected. All feature measurements are then sent to the EKF to populate the output measurement equation. If a particular feature is invalid, the corresponding measurement covariance is set to a sufficiently large value to discourage use of the incorrect reading. This is done instead of restructuring the output equations and corresponding matrices as it is computationally simpler. After the filter's iteration, the new *a priori* state estimate is used to find the inertial-frame features' new location estimates based on (5), which is then projected into each camera's frame for use in the next iteration of correspondence matching.

8. RESULTS

8.1 Simulation results

The visual tracking system is first simulated using the Matlab/Simulink environment. The calculated camera intrinsics and extrinsics of the physical set-up are used in the simulation with an update rate of 50 Hz. Table 1 shows the noise values used in the simulation. σ_v and σ_ω are measured in m/s^2 and rad/s^2 respectively, whilst σ_τ, σ_x, and σ_y are unitless. The three feature locations are generated by back-projecting the true inertial-frame position through each camera's extrinsics, and then adding the appropriate noise and quantisation. The object to be localised starts at an initial position of $\mathbf{p}^i_o = [0\ 1\ 1]^T$ metres. After 0.1 seconds it begins to move in a 1 metre radius circle in the inertial-frame x-y plane, whilst keeping a constant altitude. The attitude of the object constantly varies during the manoeuvre. Fig. 2 and 3 show the inertial-frame position and attitude estimate tracking over the first 1.8 seconds. Fig. 4 shows the convergent behaviour of the shutter delay estimates.

Table 1. Standard deviations used for the input and output measurements

σ_v	σ_ω	σ_τ	σ_x	σ_y
10	5	1×10^{-3}	1.2×10^{-3}	1.9×10^{-3}

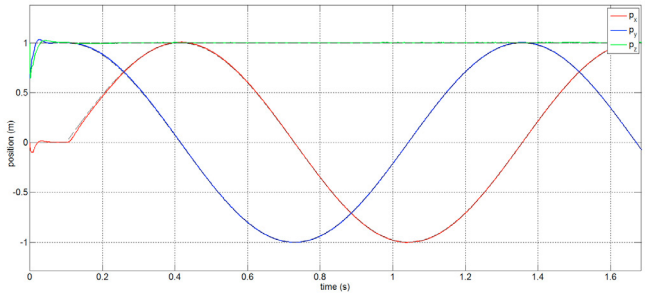

Fig. 2. Position estimate compared to true position (dotted lines) over 1.8 seconds of manoeuvre

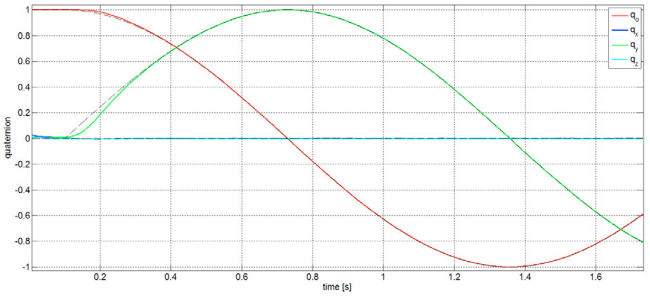

Fig. 3. Quaternion estimate compared to true quaternion (dotted lines) over first 1.8 seconds of manoeuvre

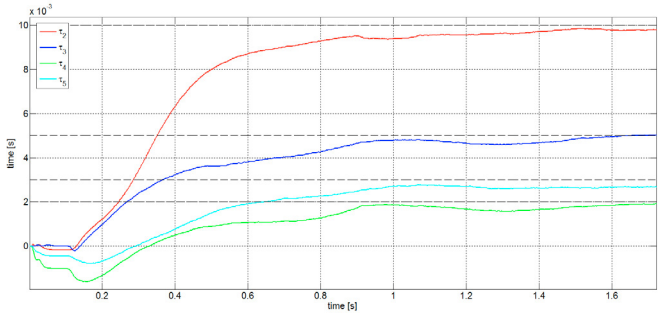

Fig. 4. Shutter delay estimate compared to true shutter delay (dotted lines) over first 1.8 seconds of manoeuvre

8.2 Experimental validation

As a simple proof of concept the camera system is made to track a single feature moving at varying speeds in a fixed circle in the x^i-y^i plane. The cameras are purposefully

initialised at different times. Fig. 5 shows the filter's position estimates and corresponding error ellipses, with and without delay estimation, over a single rotation, with the feature moving at $v_f = 2.6$ m/s. Fig. 6 shows the same information, but with the feature moving at $v_f = 7$ m/s. Table 2 contains the root-mean-square (RMS) error of the two tests, with and without shutter delay estimation. In the case of no delay estimation, uniform noise of ± 1 frame, correctly scaled by the projected velocity estimate, is appropriately added to the camera equations.

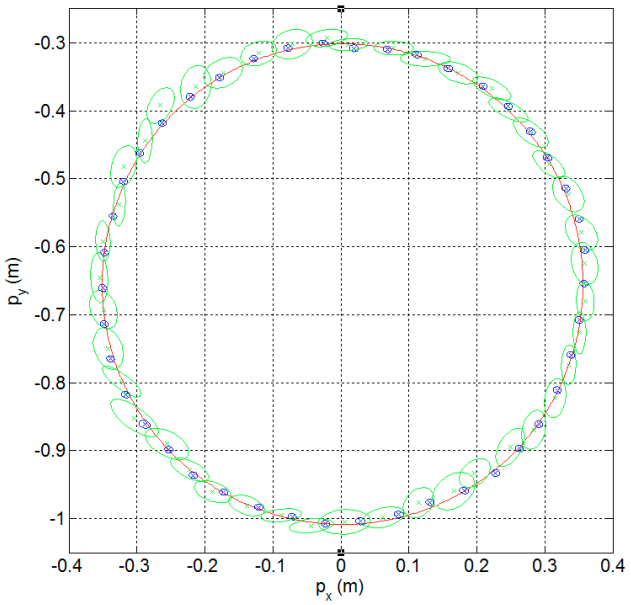

Fig. 5. x^i-y^i position estimate with delay estimation (blue) and without delay estimation (green). The error ellipses are also shown for each iteration. The true path is shown in red for a feature moving clockwise at 2.6 m/s

Table 2. Root-mean-square error for the two single-feature tests, with and without shutter delay estimation

v_f (m/s)	$e_{\tau=0}$ (mm)	$e_{\tau\neq 0}$ (mm)
2.6	8.3	4.2
7	19.7	5.5

Both versions of the estimators in Fig. 5 and 6 capture the true path of the object within their respective error ellipses, but the estimator which includes the shutter delay as a state has a much smaller error ellipse. Fig. 7 shows the shutter delay estimates over the first 1300 iterations. Fig. 8 shows the diagonal position error standard deviations, which is obtained by square rooting the diagonal position error covariances. Noticeably, the z^i-axis errors in Fig. 8 are larger than that of the x^i- and y^i-axis errors. This can be attributed to the camera orientations. The majority of the cameras' y^c-axes, which have a lower resolution than the x^c-axes, are roughly aligned with the z^i-axis. Additionally, the closest camera to the test rig has its image plane parallel to the x^i-y^i plane, which gives little to no information in the z^i-axis. The spikes in Fig. 8 are a result of artificially increasing the output noise covariance of a camera when a specific feature is invalid, as explained in Section 7.

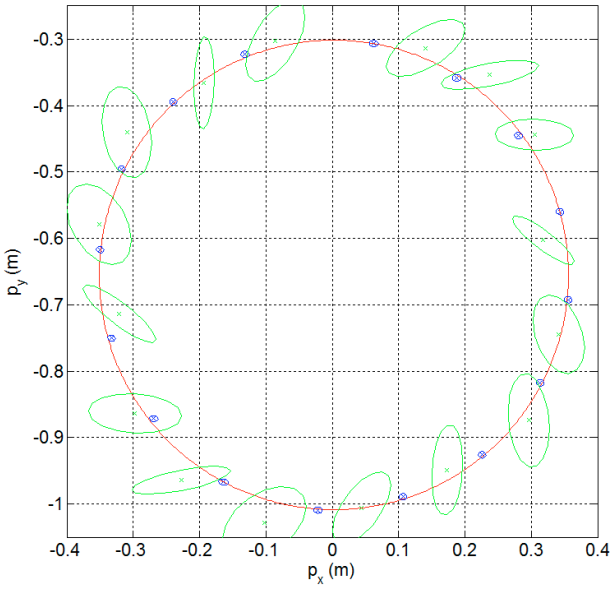

Fig. 6. x^i-y^i position estimate with delay estimation (blue) and without delay estimation (green). The error ellipses are also shown for each iteration. The true path is shown in red for a feature moving clockwise at 7 m/s

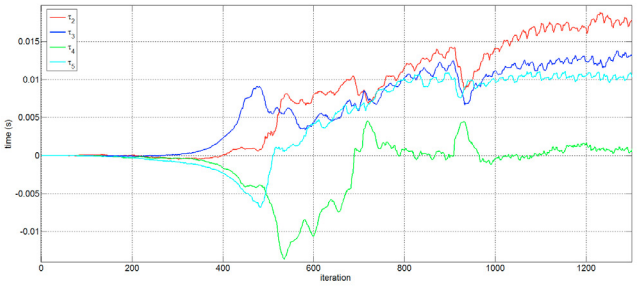

Fig. 7. Shutter delay estimates over first 1300 iterations for test with $v_f = 7$ m/s

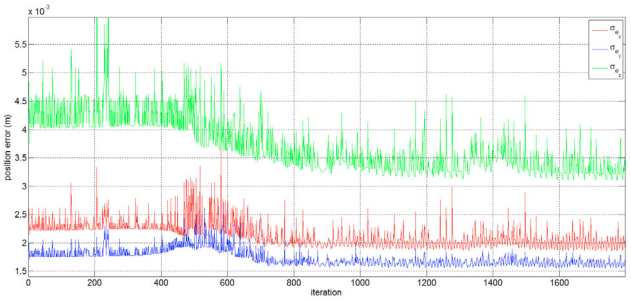

Fig. 8. Diagonal position error standard deviations during first 26 seconds for test with $v_f = 7$ m/s

8.3 Multi-feature tracking

As a final demonstration of the camera's full state estimation, a quadrotor mock-up, with three features located on its body, is used to emulate a 360° flip manoeuvre. The quadrotor starts near the inertial-frame origin and then rises to an altitude of about 1 metre. It then proceeds to perform a fast 360° roll manoeuvre, before landing back

near the origin. The same noise values from Table 1 are used in the experiment. Fig. 9 shows the inertial-frame position estimate of the quadrotor, and Fig. 10 shows the inertial-frame attitude estimate, described using intrinsic 3-2-1 Euler angles.

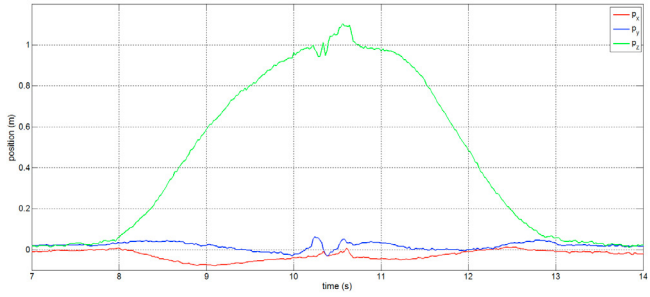

Fig. 9. Inertial-frame position estimate of quadrotor during a 360° flip manoeuvre

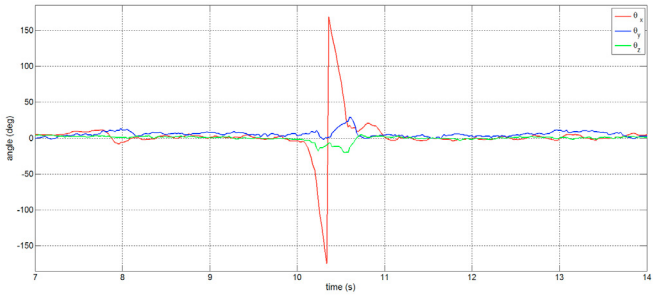

Fig. 10. Inertial-frame pose estimate of quadrotor during a 360° flip manoeuvre

9. CONCLUSION

A low-cost visual tracking system was developed, which reliably estimated pose, translation, and the shutter delays between the different cameras. With simple tests it was shown that the system is able to localise single features with an RMS error of around 5 mm, even under poor camera synchronisation and fast manoeuvres.

Although not extensively investigated, multi-feature object tracking appeared to yield even better position accuracy with 2° attitude errors. The low per-camera cost and simple tracking algorithm means that the camera system can easily be scaled to cover larger capture volumes and/or achieve higher accuracy. Notably, imperfect camera calibration can manifest itself as a small shutter delay bias, which varies depending on the feature's pixel location. With extensive calibration methods this bias can be reduced to an inconsequential size.

REFERENCES

Achtelik, M., Zhang, T., Kuhnlenz, K., and Buss, M. (2009). Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors. In *2009 International Conference on Mechatronics and Automation*, 2863–2869. IEEE.

Bouguet, J.Y. (2015). Camera calibration toolbox for matlab. Website. URL `https://www.vision.caltech.edu/bouguetj/`. Last checked: 31.10.2016.

Brown, R.G. and Hwang, P.Y. (1997). Introduction to random signals and applied kalman filtering: with matlab exercises and solutions. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997.*, 1.

Charmed Labs (2016). Pixycam product page. Website. URL `http://charmedlabs.com/default/pixy-cmucam5/`. Last checked: 05.11.2016.

Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16), 1–35.

Fu, C., Carrio, A., and Campoy, P. (2015). Efficient visual odometry and mapping for unmanned aerial vehicle using arm-based stereo vision pre-processing system. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, 957–962. IEEE.

Gao, F. and Shen, S. (2016). Online quadrotor trajectory generation and autonomous navigation on point clouds. In *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*, 139–146. IEEE.

Gupte, S., Mohandas, P.I.T., and Conrad, J.M. (2012). A survey of quadrotor unmanned aerial vehicles. In *Southeastcon, 2012 Proceedings of IEEE*, 1–6. IEEE.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Huang, A.S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., and Roy, N. (2017). Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Robotics Research*, 235–252. Springer.

Leishman, R.C., Macdonald, J.C., Beard, R.W., and McLain, T.W. (2014). Quadrotors and accelerometers: State estimation with an improved dynamic model. *IEEE Control Systems*, 34(1), 28–41.

Nemra, A. and Aouf, N. (2009). Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering.

Omari, S. and Ducard, G. (2013). Metric visual-inertial navigation system using single optical flow feature. In *Control conference (ECC)*, 1310–1316.

Reddi, Y. and Boje, E. (2014). System identification for low-cost small-scale helicopters. In *World Congress*, volume 19, 8831–8836.

Richards, J.G. (1999). The measurement of human motion: a comparison of commercially available systems. *Human movement science*, 18(5), 589–602.

Schmitz, A., Ye, M., Shapiro, R., Yang, R., and Noehren, B. (2014). Accuracy and repeatability of joint angles measured using a single camera markerless motion capture system. *Journal of biomechanics*, 47(2), 587–591.

Vicon Motion Capture Systems Ltd. (2016). Vicon vantage product page. Website. URL `http://www.vicon.com/products/camera-systems/`. Last checked: 01.11.2016.

Windolf, M., Götzen, N., and Morlock, M. (2008). Systematic accuracy and precision analysis of video motion capturing systems,exemplified on the vicon-460 system. *Journal of biomechanics*, 41(12), 2776–2780.