The International Journal of Robotics Research
30(14) 1755–1774

DOI: 10.1177/0278364911416391
ijr.sagepub.com

Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and GPU implementation

Myung Hwangbo, Jun-Sik Kim and Takeo Kanade

Abstract

When a camera rotates rapidly or shakes severely, a conventional KLT (Kanade–Lucas–Tomasi) feature tracker becomes vulnerable to large inter-image appearance changes. Tracking fails in the KLT optimization step, mainly due to an inadequate initial condition equal to final image warping in the previous frame. In this paper, we present a gyro-aided feature tracking method that remains robust under fast camera–ego rotation conditions. The knowledge of the camera's inter-frame rotation, obtained from gyroscopes, provides an improved initial warping condition, which is more likely within the convergence region of the original KLT. Moreover, the use of an eight-degree-of-freedom affine photometric warping model enables the KLT to cope with camera rolling and illumination change in an outdoor setting. For automatic incorporation of sensor measurements, we also propose a novel camera/gyro auto-calibration method which can be applied in an in-situ *or on-the-fly fashion. Only a set of feature tracks of natural landmarks is needed in order to simultaneously recover intrinsic and extrinsic parameters for both sensors. We provide a simulation evaluation for our auto-calibration method and demonstrate enhanced tracking performance for real scenes with aid from low-cost microelectromechanical system gyroscopes. To alleviate the heavy computational burden required for high-order warping, our publicly available GPU implementation is discussed for tracker parallelization.*

Keywords

Auto-calibration, GPU parallel programming, KLT feature tracking, visual/inertial sensor fusion

1. Introduction

Feature tracking is a front-end process for many vision applications from optical flow to object tracking to 3D scene reconstruction. Therefore, higher-level computer vision algorithms require robust tracking performance no matter how a camera moves. The KLT (Kanade–Lucas–Tomasi) feature tracker is one of the most prevalent tracking methods that use template image alignment techniques. It has been extensively studied in the seminal work of Lucas and Kanade (1981), Shi and Tomasi (1994), and in the unifying framework of KLT variants by Baker and Matthews (2004).

The basic assumption of KLT feature tracking is that image appearance changes slowly over time. Hence, KLT is naturally vulnerable to fast rotation or severe camera shaking. Figure 1 shows two challenges of interest in feature tracking. The first occurs when a camera's out-of-plane rotation (pan or tilt) induces large optical flow, a frequent occurrence with hand-held cameras. The translation amount is located out of KLT's convergence region. Even a multi-resolution approach does not have the capability to account for such a large motion. The second challenge occurs when a camera's in-plane rotation (roll) produces template rotation in addition to large translation during the banking turn of an airplane. Rather than a simple two-degree-of-freedom (2-DOF) translation model, a higher-order template motion model (image warping function), such as a 6-DOF affine model, is preferable for describing this template rotation. Nonetheless, additional computational load resulting from higher warping parameters should be handled properly if realtime performance is needed.

This fragility of the KLT under large camera motion inherently results from its local searching process. Mathematically, the KLT is a gradient search for warping parameters in the nonlinear minimization of difference between the template and a new image. Initial parameters

Robotics Institute, Carnegie Mellon University, Pittsburgh PA, USA

Corresponding author:
Myung Hwangbo, Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
Email: myung@cs.cmu.edu

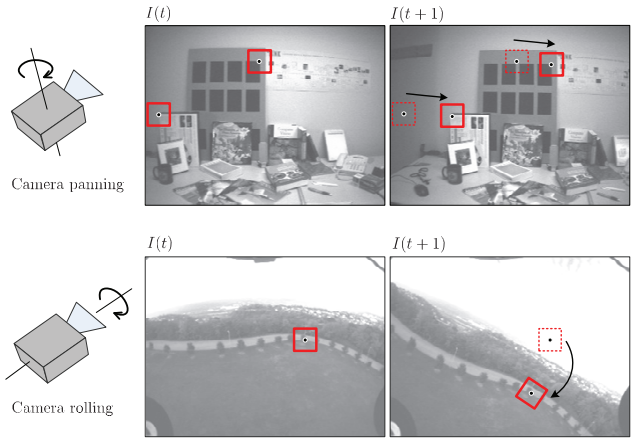

Fig. 1. Two common challenges of KLT feature tracking during severe camera ego-rotation. Top: large image translation caused by rapid camera panning; bottom: rotational image warping caused by camera rolling.

are usually set to the same values of the last iteration in a previous time frame. If large camera motion causes this initial condition to fall out of the convergence region of the current frame, feature tracking will fail. In addition, when the template uses a high-order motion model, the initial condition becomes more important in order to avoid parameter overfitting in a high-dimensional search space. Therefore, adequate estimation of initial parameters is critical for coping with severe image deformation.

We show that the knowledge of a camera's inter-frame rotation obtained from an IMU (inertial measurement unit) can significantly improve the initial condition of the KLT. When a camera rotates quickly, the gyro angle can propose a better local search region in which the parameters are more likely to converge to a true solution. Even a low-cost MEMS (microelectromechanical system) IMU is known to provide acceptable instantaneous camera rotation once its inertial sensor elements are calibrated properly. In this paper, the two main issues we address are (i) how to enhance the *search region* and *tracking motion model* when inertial gyroscopes are available, and (ii) how to *automatically* calibrate a camera and gyroscopes for accurate sensor fusion.

For the first issue, we suppose that the image sequence experiences a severe appearance change in a short time by a full 3D camera rotation such as that shown in Figure 1. A perspective transformation is the most general spatial deformation but it is prone to overfitting in the case of small template size. An affine model is sufficient for explaining local deformation in most cases. To deal with template deformation as well as illumination change, we choose the 8-DOF affine photometric model proposed by Jin et al. (2001), which employs both affine transformation and scale-and-offset illumination variation. We explain how to integrate IMU information in this high-order warping model and demonstrate how much performance improvement can be achieved.

For the second issue, each sensor's intrinsic parameters and their relative orientation need to be identified for gyro-aided feature tracking. It would be most desirable to have *in-situ* or *on-the-fly* calibration capability. Even when a sensor setting requires occasional configuration changes or when improvised sensor deployment occurs with no prior knowledge about the sensors, calibration can easily be run whenever needed. We will present an auto-calibration procedure that depends only on feature tracking and raw gyro measurements. There is no need for a calibration device such as a checkered board. Instead a set of homographies induced by natural landmarks is sufficient for calibration input.

In this paper, we assume that *rotation* is a dominant camera motion apparent in hand-held devices or dynamic aerial maneuvers, while translation could be also significant in some ground-vehicle applications. Since the camera rotation induces more rapid and severe image deformation, we do not include the accelerometer in the sensor fusion. This is also based on the following considerations: the camera's translational component is not directly measurable from the accelerometer as static gravity needs to be removed via accurate IMU attitude estimation; optical flow by camera translation is involved with feature depth, which demands complicated 3D scene reconstruction. Our goal is to maximize incorporated performance with a low-grade IMU while maintaining a concise and directly coupled fusion structure for a low-level vision processing, such as feature tracking.

The 8-DOF affine photometric model has not been popular in practice due to high computational complexity, while its tracking performance would be more robust. We alleviate this problem through restrained Hessian update in KLT as well as GPU (graphical processing unit) acceleration by means of parallel computing, which allows for tracking of up to 1,024 features at video rate.

2. Related work

Fusion between vision and inertial sensors has been an active research area. Like human visual and vestibular sensing (Corke et al. 2007), complementary characteristics in accuracy, frequency, and interaction with the environment make these sensors a prevalent choice for designing multi-sensor integration. For example, their cooperation in robotic applications can be found in vision-aided inertial navigation systems used for precise vehicle state estimation (Davison et al. 2007; Hol et al. 2007), and, conversely, in inertial-aided image understanding for non-textural scenes (Lobo and Dias 2003). One of the most tightly coupled systems of visual–inertial sensing is image stabilization, which removes unwanted motion in a video sequence by estimating image warping from registered key features.

Zhang et al. (2008) evaluated and compared several whole-image stabilization techniques.

Feature tracking or feature matching in a visual–inertial system is typically combined with predicted inter-image camera motion through a high-rate inertial sensor. In the fields of visual simultaneous localization and mapping (SLAM) (Davison et al. 2007) and augmented reality (You et al. 1999; Yokokohji et al. 2000; Bleser et al. 2006; Chandaria et al. 2007), the latest estimates of reconstructed 3D landmark position and camera pose are used as a starting point for their own designated feature matching process. As low-level front-end process for other vision algorithms, however, these Kalman filter-based schemes are too complex and computationally expensive in estimating feature and camera states with their covariances. On the other hand, IMU data has been more deeply and succinctly coupled with a feature tracker. Makadia and Daniilidis (2005) used a measured gravity direction to reduce the number of model parameters and run the Hough transform to estimate camera motion without feature correspondence. Gray and Veth (2009) compensated for a feature space of scale invariant feature transform (SIFT) descriptor using the IMU to achieve robust correspondence matching with low computational overhead. Our proposed method also incorporates gyroscopes but we proactively update KLT warping parameters using instantaneous gyro angles in order to deal with larger deformation.

Camera/IMU calibration is essential in correctly fusing raw measurements in a common frame of reference. In the literature, most visual–inertial calibration methods demand external calibration devices in order to explicitly provide known 3D feature points for camera or motion references of an IMU (Lang and Pinz 2005; Lobo and Dias 2007; Mirzaei and Roumeliotis 2008; Kelly and Sukhatme 2008; Hol et al. 2010). Lobo and Dias (2007) took a multi-step approach for calibrating an IMU intrinsic shape using a pendulum, relative camera/accelerometer orientation using vertical features and the gravity, and relative translation using a turntable and a planer target, sequentially. Although this method is comprehensive enough to obtain both intrinsic and extrinsic parameters, significant expertise and labor are required with precision equipment. A Kalman filter (KF)-based approach has also been explored to cast calibration as a state estimation problem. Mirzaei and Roumeliotis (2008) formulated a vision-aided inertial navigation system with augmented states for the relative pose and IMU bias, and then estimate the state progressively from measurement updates of known 3D features using an extended Kalman filter (EKF). Kelly and Sukhatme (2008) followed a similar approach based on an unscented Kalman filter. Taking the squared sum of prediction errors in EKF as a cost function, Hol et al. (2010) employed a gray-box system identification technique that obtains parameters from nonlinear optimization of the cost. The common limitation of these KF-based methods is that a good initial parameter is essential, known feature points from a checkered board are required, and none of the intrinsic sensor parameters are included in the estimation other than IMU bias.

Our proposed calibration procedure takes a different approach since an online and natural feature-based solution is targeted; a known feature object is no longer needed and, furthermore, intrinsic parameters of the camera and gyroscopes are estimated together with no requirement for an initial guess. We use a set of image homographies obtained during tracking of natural scene points at a distance, not only for the camera calibration but also for the gyroscope calibration. The rotating-camera calibration (Hartley 1997) is still valid for the case where camera translation is relatively small with respect to the average distance to scene points. The rotation magnitudes veiled in the homographies can provide sufficient constraints for the IMU self-calibration method (Hwangbo and Kanade 2008) to recover the gyroscope scale and alignment. The advantage is that all of the relevant parameters are linearly solvable with no prior and guaranteed to be an optimal solution for Gaussian noise.

Many KLT variants have been implemented on GPUs (Hedborg et al. 2007; Sinha et al. 2007; Ohmer and Redding 2008; Zach et al. 2008) to accelerate realtime performance in a parallel computing architecture since trackers are algorithmically independent of each other. The time complexity of KLT is dominated by the Hessian update $\mathcal{O}(n^3)$ where n is the number of warping parameters. We deal with the affine photometric model ($n = 8$) which requires at least 64 times more computation than the translational model ($n = 2$), which many other GPU-KLTs have chosen.

In addition to our earlier publication (Hwangbo et al. 2009) and the details on realtime GPU implementation (Kim et al. 2009), this paper introduces a new auto-calibration method partially based on our IMU calibration (Hwangbo and Kanade 2008), and presents more comprehensive analysis on the performance of our gyro-aided KLT feature tracking.

3. Background: KLT

KLT feature tracking is a sequence of search operations that locates an identical feature along incoming images. Given small appearance changes between temporally adjacent images, KLT is formulated as a nonlinear optimization problem (1) and searches for a change of warping parameters, $\delta\mathbf{p}$, to minimize the intensity difference e between the template T and a new image I_{t+1}:

$$e = \sum_{\mathbf{x}\in\mathcal{A}} s(\mathbf{x}) \left[T(\mathbf{x}) - I_{t+1}(\mathbf{w}(\mathbf{x}; \mathbf{p}_t, \delta\mathbf{p})) \right]^2 \qquad (1)$$

where $\mathbf{x} = (x, y)^\top$ is a pixel coordinate, $\mathbf{w}(\cdot)$ is a tracking motion model (image warping function), $\mathbf{p}_t$ is a vector of

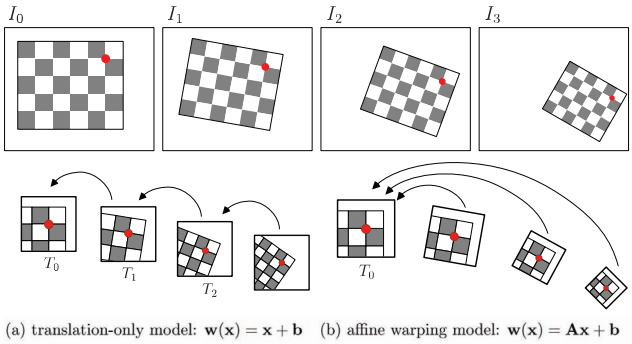

Fig. 2. Image alignments in the inverse KLT method and comparison of template updates for different warping models: (a) a low-order warping function (e.g. translation-only model: $\mathbf{w}(\mathbf{x}) = \mathbf{x} + \mathbf{b}$) is subject to frequent template updates from T_0 to T_1 to T_2; (b) a high-order warping function (e.g. affine warping model: $\mathbf{w}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$) is able to maintain the same template T_0 with rotational and scale deformations.

warping parameters at time $\mathbf{t}$, $\mathcal{A}$ is the area of a template window, and $s(\mathbf{x})$ is a weight function, and is simply a constant or a Gaussian-like function meant to emphasize the central region of $\mathcal{A}$ (Shi and Tomasi 1994).

There exist two main iterative approaches for finding the $\delta\mathbf{p}$ that align two image patches for template matching: *forward* and *inverse* methods (Baker and Matthews 2004). Both are equivalent up to a first-order approximation of the cost function (1). Key differences lie on which image remains fixed and how to update a warping parameter during iterations. The original Lucas–Kanade algorithm (Lucas and Kanade 1981) is the forward method that recomputes the Hessian at every iteration from the gradient of the warped image of I_{t+1}. In contrast, the inverse method illustrated in Figure 2 uses a fixed Hessian from the gradient of the template T, and consequently has significant computational efficiency over the forward method.

3.1. Inverse compositional image alignment

This alignment method begins with switching the roles of the image I_{t+1} ($=I$) and the template T from (1) to (2). The Gauss–Newton method to solve for $\delta\mathbf{p}$ involves two steps: (i) linearize the cost e by the first-order Taylor expansion and (ii) find a local minimum by taking the partial derivative with respect to $\delta\mathbf{p}$. Linearizing at the current warping parameter, i.e. $\delta\mathbf{p} = \mathbf{0}$, yields

$$e = \sum_{\mathbf{x}\in\mathcal{A}} \left[I(\mathbf{w}(\mathbf{x};\mathbf{p}_t)) - T(\mathbf{w}(\mathbf{x};\delta\mathbf{p})) \right]^2 \tag{2}$$

$$\approx \sum_{\mathbf{x}\in\mathcal{A}} \left[I(\mathbf{w}_{\mathbf{x}}(\mathbf{p}_t)) - T(\mathbf{w}_{\mathbf{x}}(\mathbf{0})) - \mathbf{J}(\mathbf{x})\,\delta\mathbf{p} \right]^2 \tag{3}$$

where $\mathbf{w}(\mathbf{x};\cdot) = \mathbf{w}_{\mathbf{x}}(\cdot)$ for brevity and $s(\mathbf{x}) = 1$ here. The Jacobian $\mathbf{J} = \frac{\partial T}{\partial \mathbf{p}}\big|_{\mathbf{p}=\mathbf{0}}$ is referred to as the steepest decent image. The partial derivative of (3) with respect to $\delta\mathbf{p}$ is

$$\frac{\partial e}{\partial \delta\mathbf{p}} = 2\sum_{\mathbf{x}\in\mathcal{A}} \mathbf{J}^\top \left[I(\mathbf{w}_{\mathbf{x}}(\mathbf{p}_t)) - T(\mathbf{x}) - \mathbf{J}(\mathbf{x})\,\delta\mathbf{p} \right] = 0. \tag{4}$$

Rearranging (4) yields a closed-form solution for $\delta\mathbf{p}$ at a local minimum. It iterates until $\|\delta\mathbf{p}\| < \epsilon$ for a small ϵ due to a linearization error:

$$\delta\mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x}\in\mathcal{A}} \mathbf{J}^\top \left[I(\mathbf{w}_{\mathbf{x}}(\mathbf{p}_t)) - T(\mathbf{x}) \right] \tag{5}$$

$$\mathbf{w}_{\mathbf{x}}(\mathbf{p}_{t+1}) \leftarrow \mathbf{w}_{\mathbf{x}}(\mathbf{p}_t) \circ \mathbf{w}_{\mathbf{x}}^{-1}(\delta\mathbf{p}) \tag{6}$$

where $\mathbf{H} = \sum \mathbf{J}^\top\mathbf{J}$ is the *Hessian* (precisely a first-order approximation to the Hessian). The $\mathbf{J}$ and $\mathbf{H}$ remain unchanged until the template T is updated.

3.2. Affine photometric warping model

The choice of a tracking motion model determines the level of allowable image deformation of a template window. We employ the affine-photometric warping proposed by Jin et al. (2001) for more robust tracking to camera rotation and outdoor illumination conditions. This model has a total of eight parameters, $\mathbf{p} = (a_1, \ldots, a_6, \alpha, \beta)$ in (7). The affine warp $(\mathbf{A}, \mathbf{b})$ deals with spatial deformation, including template translation and rotation. The scale-and-offset model (α, β) treats contrast variations in the template due to illumination change. The scale α compensates for the change of ambient light and the bias β does so for the change of directed light:

$$T(\mathbf{x};\mathbf{p}) = (\alpha + 1)\, T(\mathbf{A}\mathbf{x} + \mathbf{b}) + \beta \tag{7}$$

$$\text{where } \mathbf{A} = \begin{bmatrix} 1 + a_1 & a_2 \\ a_3 & 1 + a_4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}.$$

The Jacobian $\mathbf{J}$ with respect to the warping parameter $\mathbf{p}$ is computed at $\mathbf{p} = \mathbf{0}$ using the chain rule:

$$\mathbf{J} = \frac{\partial T}{\partial \mathbf{p}}\bigg|_{\mathbf{p}=\mathbf{0}} = \begin{bmatrix} \frac{T}{\mathbf{a}}\big|_{\mathbf{a}=\mathbf{0}} & \frac{T}{\alpha} & \frac{T}{\beta} \end{bmatrix} = \begin{bmatrix} \frac{T}{\mathbf{a}}\big|_{\mathbf{a}=\mathbf{0}} & T & 1 \end{bmatrix} \tag{8}$$

$$\frac{\partial T}{\partial \mathbf{a}}\bigg|_{\mathbf{a}=\mathbf{0}} = \frac{\partial T}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial \mathbf{a}}\bigg|_{\mathbf{a}=\mathbf{0}} = \frac{\partial T}{\partial \mathbf{x}}\frac{\partial \mathbf{w}}{\partial \mathbf{a}} = \nabla T \frac{\partial \mathbf{w}}{\partial \mathbf{a}}$$

$$= \nabla T \begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{bmatrix}. \tag{9}$$

The spatial template gradient $\nabla T = [G_x, G_y]$ gives

$$\mathbf{J} = \begin{bmatrix} xG_x & yG_x & xG_y & yG_y & G_x & G_y & T & 1 \end{bmatrix}. \tag{10}$$

In (5), $\delta\mathbf{p}$ requires the inversion of the symmetric 8×8 Hessian ($\mathbf{H} = \sum_{\mathcal{A}} \mathbf{J}^\top\mathbf{J}$) which has a computation complexity

$\mathcal{O}(n^3)$. Finally, the update rule (6) for the affine photometric model at each iteration is given by

$$(\mathbf{A}, \mathbf{b})_{t+1} \leftarrow (\mathbf{A}_t \delta\mathbf{A}^{-1},\ \mathbf{b}_t - \mathbf{A}_t \delta\mathbf{b}), \quad (11)$$

$$\alpha_{t+1} \leftarrow (\alpha_t + 1)/(\delta\alpha + 1), \quad (12)$$

$$\beta_{t+1} \leftarrow \beta_t - (\alpha_t + 1)\,\delta\beta. \quad (13)$$

3.3. Template update and multi-resolution pyramid

Once a template is registered at the feature selection step, ideally no change of the template is preferred since a template update inevitably causes an accumulation of feature localization errors. A template update is, however, often necessary to reset severe template warping in order to maintain a long tracking period for updated features believed to be identical. There exists a computational trade-off between template update frequency and warping model order. Figure 2 shows two extreme cases which differ in terms of whether the template T should be renewed for every frame or can remain unchanged across images. For the higher-order warping model, no template update is necessary under severe template deformation in an image sequence. Nonetheless, a Hessian inversion in the template update is far more expensive if it occurs.

We deliberately determine the moments expected for template updating by monitoring how accurate the current tracking is. Three quality measures we use to declare tracking success are (i) sum of squared error, (ii) normalized cross correlation, and (iii) the degree of shearness of the template window (implemented as the area ratio). If any of the measures falls below a predetermined threshold, the window around the last feature position is registered as a new template, since we track a point feature rather than the template itself.

Since the linearity in (3) is only valid for a small $\delta\mathbf{p}$, a multi-resolution KLT tracker is a *de facto* implementation which is not only able to handle larger image motion, but also increases its accuracy over multiple tracking iterations at each image scale (Bouguet 2000). A multi-resolution pyramid can be efficiently executed on the GPU and runs on three to five levels depending on the template size.

4. Gyroscope fusion for robust tracking

One fundamental assumption of the KLT is that small appearance changes occur between images. For example, it is empirically observed that a (15×15) template window for a corner feature may cover up to five-pixel translation. The image-only KLT simply begins with the parameters at the previous time step, i.e. $\mathbf{p}^0_{t+1} = \mathbf{p}_t$. Furthermore, the success of feature tracking critically depends on whether or not an initial parameter $\mathbf{p}^0_{t+1}$ is positioned in a convergence region of the nonlinear optimization in (2).

Our goal in gyroscope fusion is to increase the chance of convergence by relocating $\mathbf{p}^0_{t+1}$ at a better initial point with the aid of gyroscopes.

4.1. Gyroscope model

Suppose that gyroscopes are a sensor cluster consisting of m single-axis MEMS inertial components ($m \geq 3$). We model each component as a scale-and-offset response ($z = a\omega + v$) to an external angular rate ω aligned to its axis (Analog Devices 2010). In a 3D space, the measurement z_i of an ith component can be expressed as the sum of projection of a motion $\boldsymbol{\omega}$ on a sensitivity axis $\mathbf{s}_i$, a non-zero bias v_i, and a measurement noise η_i; i.e. $z_i = \mathbf{s}_i^\top \boldsymbol{\omega} + v_i + \eta_i$ for $i = 1, \ldots, m$. The raw gyro measurement $\mathbf{z} \in \mathcal{R}^{m\times 1}$ is modeled as

$$\mathbf{z} = \mathbf{S}^\top \boldsymbol{\omega} + \mathbf{v} + \boldsymbol{\eta} \quad (14)$$

where a shape matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_m] \in \mathcal{R}^{3\times m}$ and a bias vector $\mathbf{v} = [v_1, \ldots, v_m]^\top \in \mathcal{R}^{m\times 1}$ are intrinsic parameters of the gyroscopes. Here, $\boldsymbol{\eta}$ is given as a Gaussian noise and $\mathbf{S}$ represents the scale factor ($a_i = \|\mathbf{s}_i\|$), and alignment ($\mathbf{s}_i/\|\mathbf{s}_i\|$ in a frame of reference) per each component.

A tri-axial configuration ($m = 3$) is commonly used as a minimal setup for sensing 3D motion. However, note that we make no particular assumption regarding the internal sensor configuration and each component can be arbitrarily placed as shown in Figure 3. The auto-calibration method for $(\mathbf{S}, \mathbf{v})$ described in the next section is applicable for any sensor configuration and a redundant configuration ($m > 3$) is also allowed. See Hwangbo and Kanade (2008) for more details on the gyro model.

4.2. Convergence region

A convergence region $\mathcal{C}_t$ is defined as a set of all the initial $\mathbf{p}^0_t$ such that an iterative update $\delta\mathbf{p}$ can make these values converge to a true $\mathbf{p}^*_t$. The region $\mathcal{C}_t$ is limited to a concave surface where the first-order approximation in (4) is valid enough to provide a correct update direction of $\delta\mathbf{p}$ in (5).

It is very hard to quantify this convergence region since it is affected by numerous factors, for example, feature saliency, template size, optimization details and more. In Section 6, therefore, we empirically evaluate a distribution of $\mathcal{C}_t$ as an average tracking success rate in a function of $\mathbf{p}^0_t - \mathbf{p}^*_t$.

4.3. Camera ego-motion compensation

Figure 3 illustrates the way that the IMU gyroscopes make feature tracking more robust to large optical flows. Suppose that two images I_t and I_{t+1} are captured during a pure camera rotation $\mathbf{R}^{t+1}_t$. Then, the motion of all of the image pixels

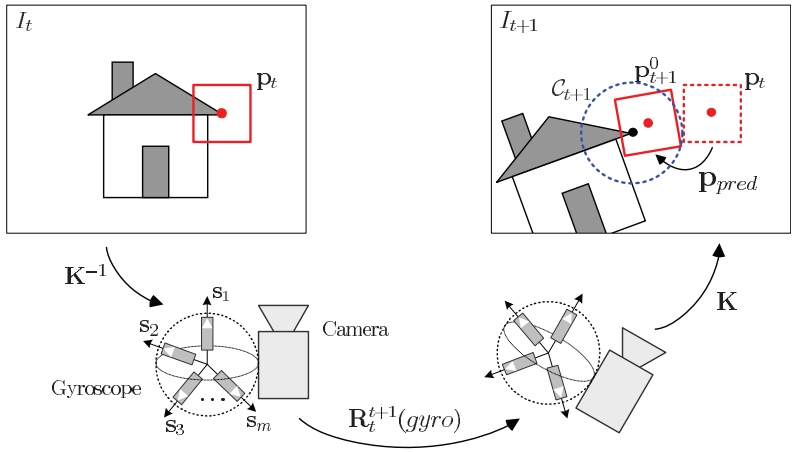

Fig. 3. Gyroscope fusion in KLT: tracking for a new image I_{t+1} starts at $\mathbf{p}^0_{t+1}$ instead of $\mathbf{p}_t$ at the previous frame as it may not be in the convergence region $\mathcal{C}_{t+1}$. Here $\mathbf{p}^0_{t+1}$ is revised using 2D homography $\mathcal{H}$ which is computed from an instantaneous gyro rotation $\mathbf{R}_t^{t+1}$ and a camera calibration matrix $\mathbf{K}$.

is described by a single 2D homography, $\mathcal{H} = \mathbf{K}\,\mathbf{R}_t^{t+1}\,\mathbf{K}^{-1}$, when a camera calibration matrix $\mathbf{K}$ is known (Hartley and Zisserman 2004). This projective transformation taking I_t to I_{t+1} can be computed by integrating the angular rate $\boldsymbol{\omega}$ in (14) from the gyro output $\mathbf{z}$ when a gyro shape and bias $(\mathbf{S}, \mathbf{v})$ and a relative orientation $\mathbf{R}_{ic}$ between camera and gyro frames of reference are also known:

$$\mathcal{H} = \mathbf{K}\,\mathbf{R}_{ic}\mathbf{R}_{\text{gyro}}\,\mathbf{K}^{-1} \tag{15}$$

$$\text{where}\quad \mathbf{R}_{\text{gyro}} = \int_t^{t+1} (\mathbf{S}^\top)^{\#}\,(\mathbf{z}(\tau) - \mathbf{v})\,d\tau.$$

Here, $(\mathbf{S}^\top)^{\#}$ is a pseudo-inverse of $\mathbf{S}^\top$ for $m > 3$ and equal to $\mathbf{S}^{-\top}$ for $m = 3$.

The homography $\mathcal{H}$ is a higher-order deformation than the affine warp we use for the spatial warping model. Hence, the predictive affine warp $(\mathbf{A}_{\text{pred}}, \mathbf{b}_{\text{pred}})$ needs to be extracted component-wise from $\mathcal{H}$. At first, the homography is divided by $\mathcal{H}_{33}$ to make $\mathcal{H}_{33} = 1$. In order to obtain all affine components (rotation, scaling and shear) except projectivity, an upper 2×2 matrix of $\mathcal{H}$ is set to the linear transformation part $\mathbf{A}_{\text{pred}}$ of the affine warp. The translation part $\mathbf{b}_{\text{pred}}$ is the same as the position change by the 2D homography. The difference of $\mathbf{x}$ and $\mathbf{x}_{\mathcal{H}}$ is set to $\mathbf{b}_{\text{pred}}$ where $\mathbf{x}_{\mathcal{H}}$ is a transferred position of $\mathbf{x}$ by $\mathcal{H}$, $[\mathbf{x}_{\mathcal{H}}\ 1]^\top \equiv \mathcal{H}[\mathbf{x}\ \ 1]^\top$. The photometric parameters $(\alpha_{\text{pred}}, \beta_{\text{pred}})$ remain unaffected:

$$\mathbf{A}_{\text{pred}} = \mathcal{H}_{2\times 2}, \tag{16}$$

$$\mathbf{b}_{\text{pred}} = \mathbf{x}_{\mathcal{H}} - \mathbf{x}, \tag{17}$$

$$\alpha_{\text{pred}} = \beta_{\text{pred}} = 0. \tag{18}$$

Once $\mathbf{R}_t^{t+1}$ are obtained from the gyroscopes, the initial parameter $\mathbf{p}^0_{t+1}$ is computed from the forward composition of the current warping parameter $\mathbf{p}_t$ with the predictive warping parameter $\mathbf{p}_{\text{pred}}$ in the inverse compositional KLT method:

$$\mathbf{w}(\mathbf{x}; \mathbf{p}^0_{t+1}) = \mathbf{w}(\mathbf{w}(\mathbf{x}; \mathbf{p}_{\text{pred}}),\ \mathbf{p}_t) \tag{19}$$

$$i.e.,\quad \mathbf{A}^0_{t+1} = \mathbf{A}_t\,\mathbf{A}_{\text{pred}},\quad \mathbf{b}^0_{t+1} = \mathbf{A}_t\,\mathbf{b}_{\text{pred}} + \mathbf{b}_t, \tag{20}$$

$$\alpha^0_{t+1} = \alpha_t,\quad \beta^0_{t+1} = \beta_t. \tag{21}$$

Basically the KLT is tolerant to an initial parameter error as long as the parameter remains in $\mathcal{C}_{t+1}$. The gyro-aided KLT starts at $\mathbf{p}^0_{t+1}$ under the expectation that the probability of $\mathbf{p}^0_{t+1} \in \mathcal{C}_{t+1}$ would be greater than that of $\mathbf{p}_t \in \mathcal{C}_{t+1}$. Algorithm 1 shows the procedure of the gyro-aided KLT feature tracking with a multi-resolution pyramid. See Kim et al. (2009) for more details on the GPU implementation.

4.4. Prediction error and gyro noise

The error in $\mathbf{p}_{\text{pred}}$ is caused by the following three main factors: a modeling error, calibration error and gyro noise. The modeling error means that camera translation is excluded when $\mathbf{p}_{\text{pred}}$ is established from $\mathcal{H}$. We here focus on how much IMU noise and error are allowable when other error sources are negligible. The modeling error will be discussed in more detail in Section 5.

Suppose that the camera is a pin-hole camera of a focal length f and a pixel scale factor k, and rotates with a constant angular rate $\boldsymbol{\omega}$ during an image sampling period Δt. Let $\lambda = kf$ and $\boldsymbol{\omega}\Delta t = \left[\theta_x\ \theta_y\ \theta_z\right]^\top$. When a small camera rotation is given around each axis ($\sin\theta \approx \theta$), the

homography $\mathcal{H}$ in (15) is approximated as follows:

$$\mathcal{H}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\lambda\theta_x \\ 0 & \lambda^{-1}\theta_x & 1 \end{bmatrix}, \quad \mathcal{H}_y = \begin{bmatrix} 1 & 0 & \lambda\theta_y \\ 0 & 1 & 0 \\ \lambda^{-1}\theta_y & 0 & 1 \end{bmatrix},$$
$$\mathcal{H}_z = \begin{bmatrix} 1 & -\theta_z & 0 \\ \theta_z & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{22}$$

Feature translation magnitudes from (17) are approximated respectively as $\|\mathbf{b}\| = \lambda\theta_i$ for camera panning/tilting ($i \in (x,y)$) and $\|\mathbf{b}\| = r\theta_z$ for rolling, where r is the distance to a feature from a camera center in pixels. When the gyroscopes are assumed to have the same noise σ_ω for each axis, the variances of $\|\mathbf{b}\|$ are given as

$$\sigma^2_{\|\mathbf{b}\|} = \begin{cases} (kf)^2\ \sigma^2_\omega\ \Delta t_s & : \text{pan/tilt}, \\ \sigma^2_\omega\ \Delta t_s & : \text{roll}. \end{cases} \tag{23}$$

For the effect of a gyro noise $\boldsymbol{\eta}$ in a practical case, let the camera be a 1/4" CCD sensor with 640 × 480 resolution ($k = 126.0$ pixel mm^{-1}), a 60° field-of-view lens ($f = 2$ mm), and sampled at 30 Hz. A typical Allan deviation of the MEMS gyroscope is $\sigma_\omega = 0.005$ rad s^{-1} when operating at 10 Hz (Analog Devices 2010; El-Sheimy et al. 2008). Thus, the translational variances of the prediction are $\sigma_{\|\mathbf{b}\|} = 0.23$ pixels for pan/tilt motion and $\sigma_{\|\mathbf{b}\|} = 0.12$ pixels for roll motion when a feature is 120 pixels apart from the center ($r = 120$). Therefore, the errors due to the gyro noise are much smaller than the translation convergence region, which is typically a third of the template size (e.g. 5 pixels for a 15 × 15 template).

A typical calibration error in a camera/gyro system can be assumed to be 10% when manufacturing error is considered for given nominal parameters and a manual boresight alignment is done. When a camera rotates at 2 rad s^{-1} with 10% error in focal length or gyro scale, it yields 1.6 and 0.8 pixel errors in $\|\mathbf{b}\|$ for pan and roll motions, respectively. So we can conclude that the calibration error typically causes a larger error on the prediction warp $\mathbf{p}_{\text{pred}}$.

Algorithm 1: Gyro-aided KLT feature tracking on a GPU.

Set $n_{\min}$, $p_{\max}$, n_{iter}, e_{SSE}, e_{NCC}, e_{shear}.
while *tracking* **do**
  Grab a new image $\mathbf{I}_{t+1}$ and transfer it to GPU.
  Compute the gradient $\nabla\mathbf{I}_{t+1}$
  if $n_s < n_{\min}$ **then**
    Select new ($n_{\min} - n_s$) Harris corner points from $\nabla\mathbf{I}_{t+1}$.
    Compute Hessian $\mathbf{H}$ and $\mathbf{H}^{-1}$ of new ($m \times m$) templates.
    Fill in empty slots of the feature table.
  Integrate IMU gyro measurements to obtain $\mathbf{R}_{\text{cam}} = \mathbf{R}_{ic}\mathbf{R}_t^{t+1}(gyro)$.
  Compute image homography $\mathcal{H} = \mathbf{K}\mathbf{R}_{\text{cam}}\mathbf{K}^{-1}$.
  for *i=1* **to** $n_{\min}$ **do**
    Update initial warp $\mathbf{w}(\mathbf{x}, p^0_{i,t+1})$ from $\mathcal{H}$ using (21).
    for *pyramid level* = $p_{\max}$ **to** 0 **do**
      Warp image $I_{t+1}(\mathbf{w}(\mathbf{x}, \mathbf{p}_{i,t+1}))$.
      Compute error $e_i = \sum T - I_{t+1}(\mathbf{w}(\mathbf{x}, \mathbf{p}_{i,t+1}))$.
      Compute update direction $\delta\mathbf{p}_i$ using (5).
      for $k = 1$ **to** n_{iter} **do**
        Search for the best line scale s^* that minimizes e_i.
      Update parameters with $s^*\delta\mathbf{p}_i$ using (11)-(13).
    Compute tracking quality measures.
    Remove the feature if ($sse > e_{\text{SSE}}$) ∨ ($ncc < e_{\text{NCC}}$) ∨ ($shear < e_{\text{shear}}$).
  n_s = number of remaining features.

5. Camera/gyroscope auto-calibration

In the sensor fusion of the gyro-aided KLT, homography computation (15) needs to know a camera intrinsic parameter $\mathbf{K}$, IMU shape and bias parameters $(\mathbf{S}, \mathbf{v})$ and their relative orientation $\mathbf{R}_{ic}$. Instead of obtaining these parameters using calibration devices such as a checkered board, we present an auto-calibration method which only uses KLT outputs, i.e. tracks of natural landmarks. Despite unknown camera motion and unknown 3D landmark positions, both the camera and gyroscopes can be directly calibrated using the set of homographies obtained from their tracks across images.

The first step is to prepare a set of homographies from KLT feature tracks on the condition that camera rotation is dominant. Camera translation $\mathbf{t}$ does not need to be zero, but we assume that translation magnitude $\|\mathbf{t}\|$ is relatively small when compared with distances to feature points. The distance ratio ρ is defined as the distance d to the majority of features divided by the degree of camera translation $\|\mathbf{t}\|$, i.e. $\rho = d/\|\mathbf{t}\|$ in Figure 4. A higher ρ produces more accurate calibration results. After n pieces of rotation-dominant camera motion $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_n)$ are applied, feature correspondences $(\mathbf{x}_1, \mathbf{x}'_1), \ldots, (\mathbf{x}_n, \mathbf{x}'_n)$ are collected using the KLT for each motion piece during Δt_i. For example, we randomly rotate a camera in roll, pitch and yaw directions while their rotation axes and angles are completely unknown. Still, each camera motion can be

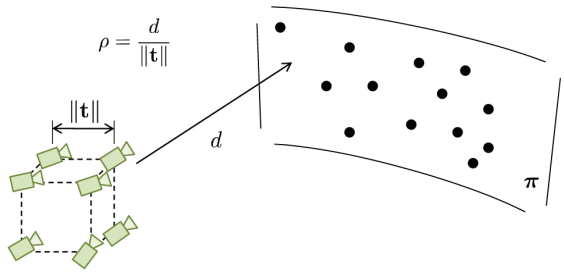

Fig. 4. Simulation setup for camera/gyro auto-calibration: a camera rigidly attached to gyroscopes randomly rotates with a non-zero translation $\mathbf{t}$, and the majority of landmarks are located at distance d from the camera. The distance ratio ρ determines how far the landmarks are located with respect to the camera translation amount $\|\mathbf{t}\|$.

abstracted by the 2D homography $\mathcal{H}_i$ estimated from the feature set $(\mathbf{x}_i, \mathbf{x}'_i)$.

The second step is to calibrate the camera and gyroscopes from this homography set. For the camera, it is well known that camera self-calibration is possible using more than three homographies induced by pure camera rotations (Hartley 1997). Although non-zero camera translation causes poor initial results, we observe that subsequent nonlinear refinement yields a far more accurate $\mathbf{K}$ when reprojection errors are taken into consideration. For the gyroscopes, we use the factorization method (Hwangbo and Kanade 2008) in which rotation magnitudes obtained from the homographies are sufficient to reconstruct a gyro shape matrix $\mathbf{S}$. The gyroscope bias $\mathbf{v}$ can be estimated separately in a static condition.

The last step is to obtain a relative orientation $\mathbf{R}_{ic}$ between both sensor frames of reference. Using the $\mathbf{K}$, $\mathbf{S}$ and $\mathbf{v}$ we find, two clouds of rotation axes can be recovered in their own frames of reference and then compared. In addition, nonlinear refinement further polishes all of the parameters simultaneously.

It is noteworthy that calibration accuracy can be improved as the whole procedure is repeated with a greater rotation speed $\|\boldsymbol{\omega}\|$. Because the gyro calibration assumes a constant angular velocity during data collection time Δt_i, the period Δt_i should be small to have less deviation in $\boldsymbol{\omega}_i(t)$ if rotated manually. The gyro-aided KLT allows a shorter Δt_i as more accurate calibration is returned. At the beginning of the calibration step, no parameters are known, thus $(\mathbf{x}_i, \mathbf{x}'_i)$ are collected by the image-only KLT (i.e. $\mathbf{S} = \mathbf{0}$). Then, steady and slow rotational motion is preferred to obtain a homography induced by large optical flows, which makes the camera calibration more accurate. Once initial calibration is performed, the entire calibration procedure is repeated with the gyro-aided KLT and faster camera motions. Consequently, overall calibration performance is progressively enhanced as more diverse angular rates are provided.

Algorithm 2 shows the procedure of our camera/gyro auto-calibration method. More details on each calibration step are described in the following.

5.1. Camera calibration

Suppose that n homographies $(\mathcal{H}_1, \ldots, \mathcal{H}_n)$ are estimated from feature tracks of rotating cameras. From (15), the goal is to find an upper triangular matrix $\mathbf{K}$ that produces a proper rotation matrix $\mathbf{R}_i = \mathbf{K}^{-1}\mathcal{H}_i\mathbf{K}$ for all i. As presented already by Hartley (1997), the orthonormality of a rotation matrix, $\mathbf{R}_i\mathbf{R}_i^\top = (\mathbf{K}^{-1}\mathcal{H}_i\mathbf{K})(\mathbf{K}^{-1}\mathcal{H}_i\mathbf{K})^\top = \mathbf{I}$, generates a set of linear constraints for the entries of a symmetric matrix $\mathbf{C} = \mathbf{K}\mathbf{K}^\top$:

$$\mathbf{C} = \mathcal{H}_i\,\mathbf{C}\,\mathcal{H}_i^\top \quad \xrightarrow[i=1,\ldots,n]{\text{stacking}} \quad \mathbf{L}\mathbf{c} = \mathbf{0} \tag{24}$$

where an unknown $\mathbf{c} \in \mathcal{R}^{6\times 1}$ is a vector of upper triangular elements of $\mathbf{C}$ and $\mathbf{L}$ is a stack of a rank-four matrix $\mathbf{L}_i(\mathcal{H}_i) \in \mathcal{R}^{6\times 6}$ which is a set of linear constraints on $\mathbf{c}$ given $\mathcal{H}_i$. From at least two homographies ($n \geq 2$), $\mathbf{c}$ is solvable from a $(6n \times 6)$ homogeneous linear form $\mathbf{L}$. The Cholesky decomposition then returns the estimate $\widehat{\mathbf{K}}$ from $\mathbf{C}$. This method has been reported to be stable for high noisy levels, e.g. a 5% focal length error in 2 pixel image noise (Hartley 1997).

Practically it is hard to restrain a camera motion from translation during *in-situ* or on-the-fly calibration. When a camera is manually rotated, for example, the camera center is inevitably translated around. In the presence of camera translation $\mathbf{t}$, a RANSAC-based homography estimation tends to find a planar homography $\mathcal{H}_p = \mathbf{K}(\mathbf{R}+\mathbf{t}\mathbf{n}^\top/d)\mathbf{K}^{-1}$ induced by a 3D plane $\boldsymbol{\pi}$ on which a majority of features are closely located, where $\mathbf{n}$ is a plane normal and d is a depth to $\boldsymbol{\pi}$ in Figure 4. Non-zero $\mathbf{t}$ causes the input $\mathcal{H}_i$ to violate the constraint (24) and leads to the system modeling error. Hence, the accuracy of the linear solution in (24) becomes the worse for the smaller $\rho = d/\|\mathbf{t}\|$.

However, we observe that a subsequent nonlinear refinement step significantly improves initial camera parameters. This improvement can be explained as follows: For a general camera motion $(\mathbf{R}, \mathbf{t})$, the mapping of an image point is $\mathbf{x}' = \mathbf{K}\mathbf{R}_i\mathbf{K}^{-1}\mathbf{x}+\mathbf{K}\mathbf{t}_i/z$, where z is a point depth. The refinement minimizes the squared sum of reprojection errors, $\sum e^2 = \sum(\mathbf{x}' - \mathcal{H}_i\mathbf{x})^2$, for all feature correspondences. The error e can be approximated by $e = \mathbf{K}\mathbf{t}/z$ when tracking noise is ignored. On the condition that $(\mathbf{t}_1, \ldots, \mathbf{t}_n)$ have diverse directions with a random magnitude, the distribution of e can be approximated to an unbiased Gaussian since e does not depend on the image position $\mathbf{x}$. The simulation results are provided in Figure 7 to verify this argument.

5.2. Gyroscope calibration

First, a gyro bias $\mathbf{v}$ is separately estimated during the stationary condition ($\boldsymbol{\omega} = \mathbf{0}$). Then, the recovery of a gyro shape $\mathbf{S}$ begins with the fact that a homography $\mathcal{H}$ and its corresponding rotation matrix $\mathbf{R}$ are *similar* matrices for the transformation $\mathbf{K}$ under the assumption $\mathbf{R} = \mathbf{K}^{-1}\mathcal{H}\mathbf{K}$. One of the similar matrix properties provides that $\mathcal{H}$ and $\mathbf{R}$ share the same eigenvalues, i.e. $\mathtt{eig}(\mathcal{H}) = \mathtt{eig}(\mathbf{R}) = (1, e^{j\theta}, e^{-j\theta})$, although their eigenvectors are generally different (Golub and Loan 1996). Let θ_i be a phase angle of the complex eigenvectors of $\mathcal{H}_i$. Suppose that $\mathcal{H}_i$ is obtained when the camera rotates at angular velocity $\boldsymbol{\omega}_i^{\text{cam}}$ during Δt_i. Then, the corresponding IMU angular rate $\boldsymbol{\omega}_i^{\text{gyro}}$ should have a magnitude equal to an angular speed, i.e. $\|\boldsymbol{\omega}_i^{\text{gyro}}\| = \|\boldsymbol{\omega}_i^{\text{cam}}\| = \theta_i/\Delta t_i$, although its rotation axis is unknown. As a result, the goal is to recover the gyro shape $\mathbf{S}$ using known angular speeds $(\|\boldsymbol{\omega}_1^{\mathcal{H}}\|, \ldots, \|\boldsymbol{\omega}_n^{\mathcal{H}}\|)^\top$, which are derived from $(\mathcal{H}_1, \ldots, \mathcal{H}_n)$ using $\|\boldsymbol{\omega}_i^{\mathcal{H}}\| = \theta_i/\Delta t_i$.

When the bias $\mathbf{v}$ is compensated, a collection of n gyro measurements $\mathbf{D}$ can be described as a product of *motion* matrix $\mathbf{W}$ and *shape* matrix $\mathbf{S}$:

$$\mathbf{D} = \mathbf{WS} \tag{25}$$

where $\mathbf{W} = [\boldsymbol{\omega}_1^\top; \ldots; \boldsymbol{\omega}_n^\top] \in \mathcal{R}^{n\times 3}$, $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_m] \in \mathcal{R}^{3\times m}$, and each entry of $\mathbf{D} \in \mathcal{R}^{n\times m}$ is a bias-compensated raw gyro measurement, i.e. $d_{ij} = z_{ij} - v_j = \boldsymbol{\omega}_i^\top \mathbf{s}_j$.

The factorization method (Hwangbo and Kanade 2008) finds a unique decomposition into $\mathbf{W}$ and $\mathbf{S}$ that satisfies the given $\boldsymbol{\tau}$. This method begins with $\mathbf{D} = \mathbf{U}\Sigma\mathbf{V}^\top$ by SVD (singular value decomposition) which yields $\mathbf{W}^* = \mathbf{U}_p\Sigma_p^{1/2}$ and $\mathbf{S}^* = \Sigma_p^{1/2}\mathbf{V}_p^\top$ where Σ_p is the top left 3×3 block of Σ, and $\mathbf{U}_p$ and $\mathbf{V}_p$ are the first three columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. This solution is only unique up to any invertible $\mathbf{A} \in \mathcal{R}^{3\times 3}$, since $\mathbf{D} = (\mathbf{W}^*\mathbf{A})(\mathbf{A}^{-1}\mathbf{S}^*)$. From $\boldsymbol{\omega}_i^* = \mathbf{A}\widehat{\boldsymbol{\omega}}_i$, the known $\|\boldsymbol{\omega}_i^{\mathcal{H}}\|$ provides a set of constraints to resolve the ambiguity $\mathbf{A}$ as follows:

$$\widehat{\boldsymbol{\omega}}_i^\top \mathbf{Q}\widehat{\boldsymbol{\omega}}_i = \|\boldsymbol{\omega}_i^{\mathcal{H}}\|^2 \quad \xrightarrow[i=1,\ldots,n]{\text{stacking}} \quad \mathbf{Bq} = \boldsymbol{\tau}^2 \tag{26}$$

where $\mathbf{Q} = \mathbf{A}^\top\mathbf{A}$, $\mathbf{q}$ is a vector of upper triangular elements of $\mathbf{Q}$, $\boldsymbol{\tau} = (\|\boldsymbol{\omega}_1^{\mathcal{H}}\|, \ldots, \|\boldsymbol{\omega}_n^{\mathcal{H}}\|)^\top$, and $\mathbf{B} \in \mathcal{R}^{n\times 6}$. From at least six camera motions ($n \geq 6$), $\mathbf{q}$ is solvable. Once $\mathbf{A}$ is found by the Cholesky decomposition, the linear solution is reconstructed as $\widehat{\mathbf{W}} = \mathbf{W}^*\mathbf{A}$ and $\widehat{\mathbf{S}} = \mathbf{A}^{-1}\mathbf{S}^*$. Note that this solution is still up to a rotation matrix $\mathbf{R}$ from $\widehat{\mathbf{W}}\widehat{\mathbf{S}} = (\widehat{\mathbf{W}}\mathbf{R})(\mathbf{R}^{-1}\widehat{\mathbf{S}})$. When $\hat{\mathbf{s}}_i$ is a normalized column vector of $\widehat{\mathbf{S}}$, we can determine $\mathbf{R} = [\hat{\mathbf{s}}_1, (\hat{\mathbf{s}}_1 \times \hat{\mathbf{s}}_2) \times \hat{\mathbf{s}}_1, \hat{\mathbf{s}}_1 \times \hat{\mathbf{s}}_2]$ such that $\mathbf{R}^{-1}\widehat{\mathbf{S}}$ becomes an upper triangular matrix.

Compared with $4n$ independent constraints for the camera calibration in (24), only n constraints exist for the gyroscope calibration in (26). Therefore, the reliable estimate $\widehat{\mathbf{S}}$ requires greater numbers of homographies than the camera calibration.

5.3. Relative orientation

It is straightforward to compute the relative orientation $\mathbf{R}_{ic}$ between both calibrated sensors. We compare two bundles of rotation axes $\{\widehat{\boldsymbol{\omega}}^{\text{cam}}, \widehat{\boldsymbol{\omega}}^{\text{gyro}}\}$ expressed in terms of their respective frames of reference. Here $\widehat{\boldsymbol{\omega}}_i^{\text{cam}}$ is extracted from $\mathbf{R}_i = \widehat{\mathbf{K}}^{-1}\mathcal{H}_i\widehat{\mathbf{K}}$ and $\widehat{\boldsymbol{\omega}}_i^{\text{gyro}}$ is obtained directly from the ith column of $\widehat{\mathbf{W}}$. The best fit of $\mathbf{R}_{ic}$ can be estimated from the SVD of $\mathbf{H} = \sum_{i=1}^n \widehat{\boldsymbol{\omega}}_i^{\text{gyro}}(\widehat{\boldsymbol{\omega}}_i^{\text{cam}})^\top$ such that $\widehat{\boldsymbol{\omega}}_i^{\text{cam}} = \mathbf{R}_{ic}\,\widehat{\boldsymbol{\omega}}_i^{\text{gyro}}$ for all i. The least squares solution is found as $\widehat{\mathbf{R}}_{ic} = \mathbf{UV}^\top$ given $\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^\top$ (Arun et al. 1987).

5.4. Nonlinear refinement

The calibration parameters from linear methods in (24) and (26) can be improved by nonlinear refinements that minimize more meaningful error measures. In a *per-sensor refinement* that finds best parameters for individual sensors, the camera calibration minimizes the reprojection error, $\sum(\mathbf{x}' - \mathbf{KR}_i\mathbf{K}^{-1}\mathbf{x})^2$, with $(5 + 3n)$ parameters including $\mathbf{K}$ and each $\mathbf{R}_i$. The gyroscope calibration minimizes the squared sum of constraints, $\sum(\|\boldsymbol{\omega}_i^{\mathcal{H}}\| - \|\mathbf{S}^{-\top}\mathbf{d}_i\|)^2$, with six parameters of an upper triangular matrix $\mathbf{S}$.

In a *total refinement* that simultaneously polishes all of the parameters $(\mathbf{K}, \mathbf{S}, \mathbf{v}, \mathbf{R}_{ic})$, the reprojection error involved with $\mathbf{R}_{\text{gyro}}$ in (27) is minimized given raw measurements $(\mathbf{x}, \mathbf{z})$. The gyro rotation $\mathbf{R}_{\text{gyro}}^i$ is numerically integrated over a time period Δt_i using corresponding gyro measurements $\mathbf{z}_i$:

$$E_{\text{total}} = \sum_{i=1}^{n}\sum_{j=1}^{n_i}(\mathbf{x}'_{ij} - \mathbf{KR}_{ic}\mathbf{R}_{\text{gyro}}^i\mathbf{K}^{-1}\mathbf{x}_{ij}) \tag{27}$$

$$\text{where} \quad \mathbf{R}_{\text{gyro}}^i = \int_{\Delta t_i} \mathbf{S}^{-\top}(\mathbf{z}_i - \mathbf{v})\,dt.$$

In the case of a tri-axial gyroscope and camera, the Levenberg–Marquardt iterative nonlinear optimization finds the total number of 17 $(= 5 + 6 + 3 + 3)$ parameters in $(\mathbf{K}, \mathbf{S}, \mathbf{v}, \mathbf{R}_{ic})$.

5.5. Camera/gyro time synchronization

In addition to spatial calibration, temporal synchronization is also necessary for combining inertial and visual sensing. High-precision time synchronization requires hardware-level external triggering based on a precise clock. When commodity sensors which have no external sync inputs are used, however, the exact times of sampling instances become obscured. They are hidden by unknown delays in the data bridge, such as communication or buffering between low-level embedded systems.

Nonetheless, if delays from the sensors to a computer are unknown but can be assumed constant, there is an easy and simple way to identify a time lag between two measurement sequences, as shown in Figure 5. We make a

Algorithm 2: Progressive auto-calibration method for the IMU-aided KLT feature tracking.

Set $\mathbf{K} = \mathbf{I}_{3\times3}$, $\mathbf{S} = \mathbf{0}$ and $\mathbf{R}_{ic} = \mathbf{I}_{3\times3}$

$k = 0$.

while $E_{total} > E_{th}$ *and* $k < k_{\max}$ **do**

Estimate the gyro bias $\mathbf{v}$ when stationary.

$n \leftarrow$ the number of camera motions.

(Data Collection)

for $i = 1$ **to** n **do**

Rotate the camera around a new rotation axis with steady angular velocity.

Run IMU-aided feature tracking using current estimates of $\mathbf{K}$, $\mathbf{S}$ and $\mathbf{R}_{ic}$

Collect feature correspondence $(\mathbf{x}_i, \mathbf{x}_i')$ between $\mathbf{I}(t)$ and $\mathbf{I}(t + \Delta t_i)$.

Collect gyro raw measurements $(\mathbf{z}_i(t), \ldots, \mathbf{z}_i(t + \Delta t_i))$ during Δt_i;

(Camera Calibration)

for $i = 1$ **to** n **do**

Estimate the Homography $\mathcal{H}_i$ from $(\mathbf{x}_i, \mathbf{x}_i')$ using RANSAC.

Stack the linear constraints into $\mathbf{L}$ in (24).

Find $\mathbf{K}$ from Cholesky decomposition of the linear solution $\mathbf{c}$ of (24).

Refine $\mathbf{K}$ by minimizing the sum of reprojection errors $e = \sum(\mathbf{x}' - \mathbf{K}\mathbf{R}_i\mathbf{K}^{-1}\mathbf{x})^2$.

(Gyroscope Calibration)

for $i = 1$ **to** n **do**

Compute $\|\boldsymbol{\omega}_i^{\mathcal{H}}\| = \theta_i/\Delta t_i$ where $\theta_i = \texttt{phase}(\texttt{eig}(\mathcal{H}_i))$.

Compute $\mathbf{d}_i = \texttt{mean}(\mathbf{z}_i - \mathbf{v})$.

Factorize $\mathbf{D}$ into $\mathbf{W}^* = \mathbf{U}_p\Sigma_p^{1/2}$ and $\mathbf{S}^* = \Sigma_p^{1/2}\mathbf{V}_p^\top$ by SVD.

Stack all $\|\boldsymbol{\omega}_i^{\mathcal{H}}\|$ constraints into $\mathbf{B}$ in (26).

Find $\mathbf{A}$ from Cholesky decomposition of the linear solution $\mathbf{q}$ of (26).

Recover $\widehat{\mathbf{S}}$ from $\mathbf{A}^{-1}\mathbf{S}^*$ and make $\mathbf{S} = \mathbf{R}^{-1}\widehat{\mathbf{S}}$ an upper triangular.

Refine $\mathbf{S}$ and $\mathbf{v}$ by minimizing the sum of constraint errors $e = \sum(\|\boldsymbol{\omega}_i^{\mathcal{H}}\| - \|(\mathbf{S}^\top)^{\#}(\mathbf{z}_i - \mathbf{v}_i)\|)^2$.

(Relative Orientation)

for $i = 1$ **to** n **do**

Compute $\boldsymbol{\omega}_i^{\text{cam}}$ from $\mathbf{R}_i = \mathbf{K}^{-1}\mathcal{H}_i\mathbf{K}$ and $\boldsymbol{\omega}_i^{\text{gyro}} = \mathbf{S}^{-\top}\mathbf{d}_i$.

Compute $\mathbf{H} = \sum_{i=1}^{n}\boldsymbol{\omega}_i^{\text{gyro}}(\boldsymbol{\omega}_i^{\text{cam}})^\top$.

Find $\mathbf{R}_{ic} = \mathbf{U}\mathbf{V}^\top$ from the SVD of $\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^\top$.

(Total Refinement)

Refine $\mathbf{K}$, $\mathbf{S}$, $\mathbf{v}$ and $\mathbf{R}_{ic}$ simultaneously by minimizing E_{total} in (27)

Increase the camera rotation speed $\|\boldsymbol{\omega}\|$ in the next iteration.

$k = k + 1$.

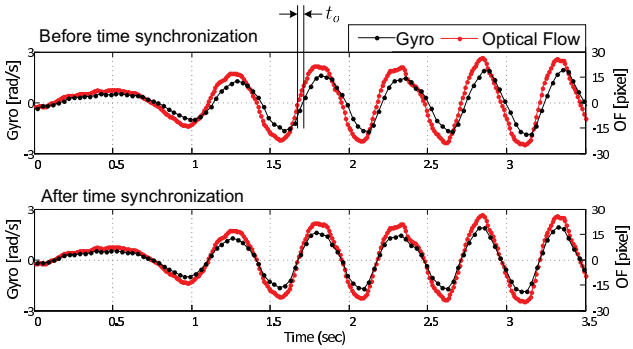

Fig. 5. Time synchronization of a camera/gyro system. Top: Given a sinusoidal panning motion, gyroscope angular speed is asynchronous with an average magnitude of optical flows. Bottom: Time offset $t_o = -0.02$ s is identified from the phase lag between two measurement sequences.

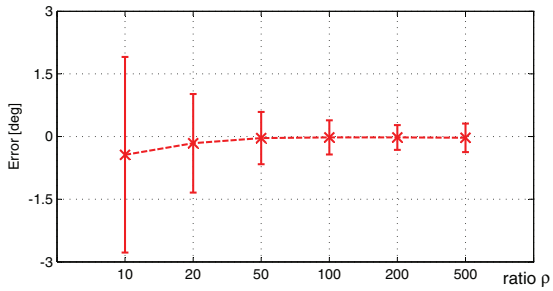

Fig. 6. Systematic error in camera rotation angle θ obtained from homography $\mathcal{H}$ ($\theta = \texttt{phase}(\texttt{eig}(\mathcal{H}))$ when $\theta_{true} = 10^\circ$: this error is induced by non-zero camera translation $\mathbf{t}$ but quickly becomes unbiased as the distance ratio $\rho = d/\|\mathbf{t}\|$ increases. The statistic of θ_{err} is derived from 100 iterations when $\mathbf{t}$ is randomly provided according to ρ.

small sinusoidal camera motion around a pan or tilt axis and then compute average magnitudes of optical flows from many feature tracks. Because the optical flow and gyroscope angle should have an identical phase, the phase lag ϕ_o between the two sensor signals reveals the time offset t_o of both sensors. Practically, we evaluate a cost function $f(t_o)$ which is a normalized cross correlation between $OF(t - t_o)$ and $\|\boldsymbol{\omega}_{\text{gyro}}(t)\|$. We then find a t_o that maximizes the cost.

5.6. Simulation

A simulation was performed to evaluate the effect of non-zero camera translation on calibration accuracy. Figure 4 shows a simulation setup that allows camera translation $\mathbf{t}$ when scene points are located at distance d from the camera. The camera was positioned in turns at each of the cube's corners. For each simulated data, the camera was translated by the amount $\|\mathbf{t}\|$ and oriented in a new angle toward features. Figure 8 shows some examples of feature tracks generated by this kind of camera pose sequence with an average

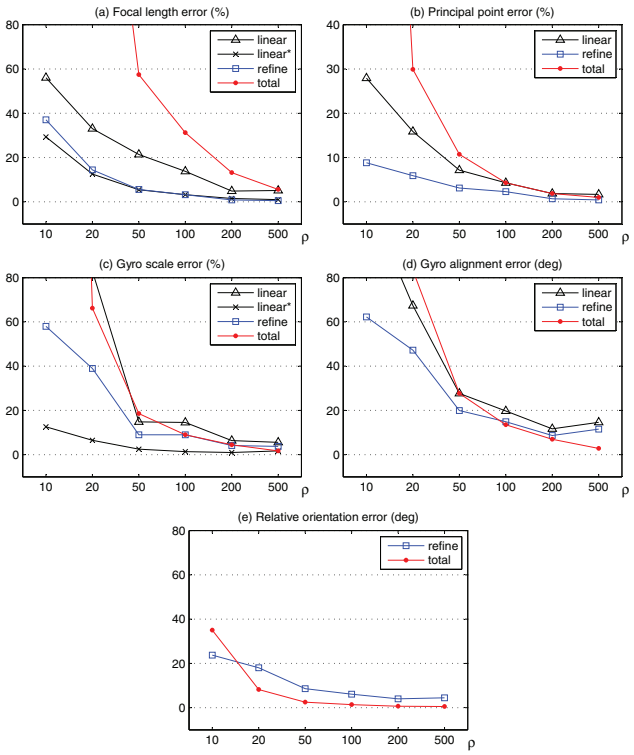

Fig. 7. Calibration accuracy according to the distance ratio ρ ($\rho = d/\|\mathbf{t}\|$ in Figure 4): `linear`* means single-parameter models ($\mathbf{K}^* = \text{diag}(f,f,1)$ and $\mathbf{S}^* = \text{diag}(s,s,s)$). `refine` refers to per-sensor nonlinear refinement with an initial obtained from `linear`. `total` refers to simultaneous total refinement of camera/gyro parameters.

10° camera rotation angle. Each $\mathcal{H}_i$ was estimated using RANSAC from an average of 150 feature tracks when a 0.5 pixel Gaussian noise was added onto 640 × 480 images. The distance ratio ρ indicates how far the landmarks are located with respect to the camera translation amount $\|\mathbf{t}\|$, i.e. $\rho = d/\|\mathbf{t}\|$. The ratio $\rho = 10$ means that the camera center moves up to 10 cm during rotations in which objects are placed 1 m ahead.

For calibration analysis, we investigate a systematic error of the angle θ obtained from a homography $\mathcal{H}$ when ρ is finite. Only when $\rho = \infty$ is there no systematic error in θ, i.e. a camera performs a pure rotation ($\|\mathbf{t}\| = 0$), or features are located at infinity ($d = \infty$). In Figure 6, the estimate of θ is biased and $\|\boldsymbol{\omega}_i^{\mathcal{H}}\|$ in (26) would be more likely to be smaller than a true angular speed. When $\rho \leq 100$, θ is expected to have more than a 4% root mean squared error.

Also evaluated in the calibration are a single-parameter camera matrix $\mathbf{K}^* = \text{diag}(f,f,1)$ and shape matrix $\mathbf{S}^* = \text{diag}(s,s,s)$. These are reasonable approximate models for most cameras and tri-axial gyroscopes. When plugged into (24) and (26), respectively, three times more linear constraints are obtained for single unknown f but the same number of constraints for s.

Figure 7 shows the calibration accuracy achieved when the distance ratio ρ increases from 10 to 500. A total of 30 homographies were used ($n = 30$). The linear calibration methods incur huge errors at a small ρ and only returns acceptable results of lower than a 5% error when $\rho \geq 200$. Even at a small ρ, however, we observe that subsequent per-sensor nonlinear refinements significantly improve calibration accuracy using initial values from the linear methods. For example, when $\rho = 50$, the factorization method returns an 18% error in scale and refinement reduces it to 8%. Owing to parameter overfitting, gyro alignments ($\angle \mathbf{s}_i\mathbf{s}_j$) have relatively large errors even when $\rho \geq 100$. Note that these simulated data were actually generated by $\mathbf{K}^*$ and $\mathbf{S}^*$ with additional Gaussian noise. Hence, the best calibration accuracy is obtained for single-parameter cases. The parameter overfitting of $\mathbf{K}$ and $\mathbf{S}$ is observed from the fact that calibration errors are critically reduced when $\mathbf{K}^*$ and $\mathbf{S}^*$ are used. Because magnitude $\|\boldsymbol{\omega}_i^{\mathcal{H}}\|$ is a relatively weak constraint for all parameters of $\mathbf{S}$, the factorization method is more sensitive to measurement noises in $\mathbf{D}$ and $\|\boldsymbol{\omega}_i^{\mathcal{H}}\|$. Hence, a larger number of calibration measurements (n) is required to restrain overfitting especially in the alignment in $\mathbf{S}$.

Total refinement, which simultaneously minimizes camera and gyro measurement errors, is compared with per-sensor refinement which independently minimizes respective measurement error. Although total refinement is far worse than per-sensor refinement at $\rho < 200$, it yields a better gyro alignment estimation at $\rho \geq 100$ and relative alignment at $\rho \geq 20$ since additional constraints from camera measurements are imposed on these parameters. Since the systematic error in θ decreases as ρ increases, total refinement is expected to return the best result at $\rho > 500$.

Figure 8 shows the prediction residuals for the gyro-aided KLT when the total refinement calibration results at $\rho = 50$ are used. When gyro assistance is supported, the average feature translation the KLT needs to track reduces from 150 pixels to 10 pixels under pan or tilt camera motions.

6. Experiments

We have tested our gyro-aided feature tracker on two different kinds of image sequences: an indoor desk scene (640×480 at 30 Hz, see Extension 1) taken in front of a desk while the camera undergoes motions, such as the shaking of a hand-held device; an outdoor aerial scene (320 × 240 at 15 Hz, see Extension 2) captured during the flight of a small fixed-wing airplane. Both datasets were acquired using the same camera/IMU system in Figure 9, but their image resolutions and frame rates were adjusted according to the processing power of their accompanying main computers.

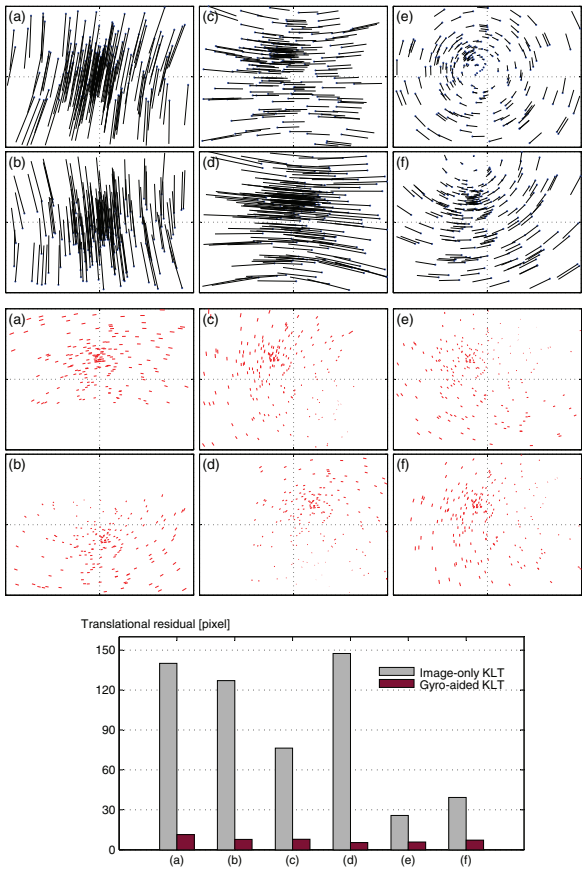

Fig. 8. Comparison of initial conditions in the image-only and gyro-aided KLTs. Top: Simulated feature tracks, equal to initial errors in the image-only KLT, are generated by the sequence of camera poses. Each rotation angle is around 10°. Middle: Prediction residuals, which are equal to initial errors in the IMU-aided KLT, are computed by auto-calibration results when $\rho = 50$. Bottom: Average initial errors in translation warping $\|\mathbf{b}\|$.

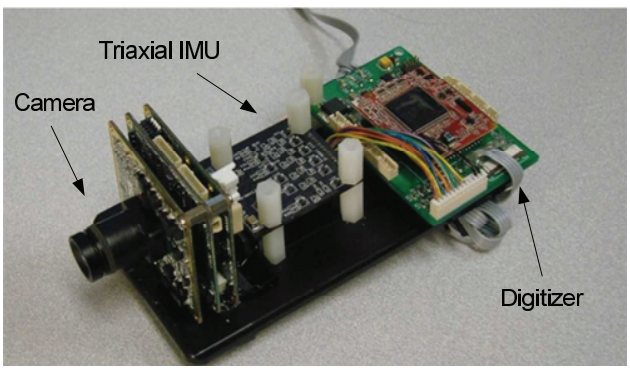

Fig. 9. A camera/IMU experimental system for the IMU-aided KLT feature tracking.

6.1. Camera/IMU system

Figure 9 shows the low-cost and lightweight camera/IMU system used in the experiments. The camera is the *Sentech* USB board camera and the IMU is an *O-Navi Gyroscube* tri-axis MEMS inertial sensor. Their outputs are connected to a main computer via USB and serial communications, respectively. Three-tuple gyroscopic values in the range of $\pm 200^\circ\ \mathrm{s}^{-1}$ are sampled at 100 Hz with a 11-bit resolution A/D converter. The IMU is rigidly attached directly behind the camera and carefully aligned with the camera to ensure that the relative orientation is zero, $\mathbf{R}_{ic} = \mathbf{I}$.

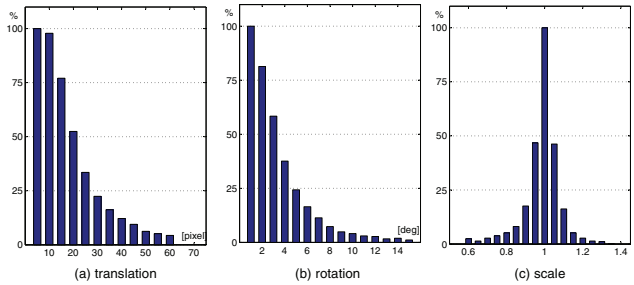

Fig. 10. Success rates of the image-only KLT according to initial warping parameter errors $\mathbf{p}_{\mathrm{err}}$: 300 Harris corner features in (21×21) template are tested with different (a) translation, (b) rotation and (c) scale errors. The zone corresponding to p-success rate indicates a gross view of the convergence region $\mathcal{C}$ such that $\mathrm{Prob}(\mathbf{p}_{\mathrm{err}} \in \mathcal{C}) = p$.

6.2. KLT convergence region

We empirically evaluate a distribution of the convergence region $\mathcal{C}$ of affine warping parameters for our KLT implementation. Since every feature has a different $\mathcal{C}$ depending on feature saliency and cost function concavity, we take a gross view of $\mathcal{C}$ in reference to how many features are successfully tracked given initial errors. Note that no gyro fusion is involved here and the performance of ordinary image-based tracking is examined.

A total of 300 Harris corner features in a 21×21 template was selected from one image in the desk scene and then tracked by a multi-resolution KLT approach with three pyramid levels. Various initial affine warping errors were repeatedly imposed in order to test whether they converge back to zero or diverge. Figure 10 shows the tracking success rates according to different translation, rotation and scale initial errors, respectively. A 50% probability zone of $\mathbf{p}_{\mathrm{err}} \in \mathcal{C}$ is found as 20 pixel translation, 3.5° rotation and 0.05 scale errors.

6.3. Gyro-aided tracking performance

The performance improvement after incorporating camera–ego motion compensation is compared with purely image-based tracking, i.e. the image-only method. For the desk scene shown in Figure 11, we shook the hand-held camera 2 m ahead of the desk and sequentially conducted three principal rotational motions (panning, tilting and

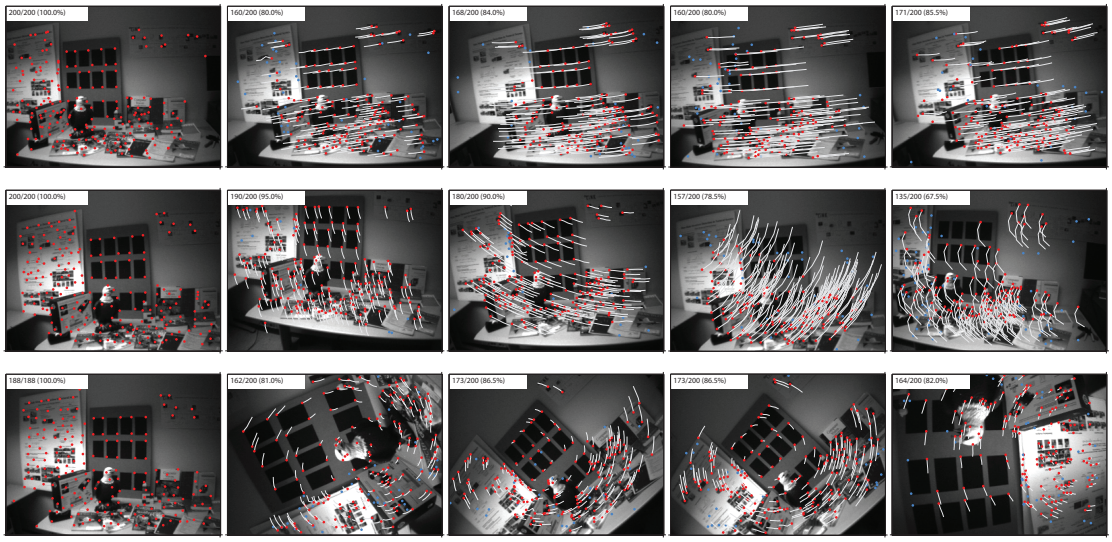

Fig. 11. Robust tracking results of the gyro-aided KLT in the 640 × 480 desk scene: each row corresponds to various types of fast camera motion (from the top, panning, tilting, and moving forward with rolling, respectively). The white tails are optical flows over the last three frames. See Extension 1 for the video demonstration.

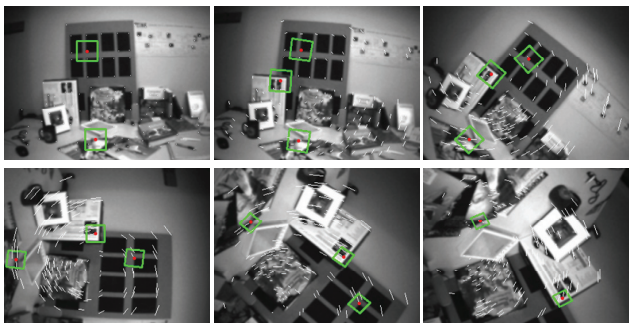

Fig. 12. Affine template warping occurred over a long period of tracking as a camera moves forward with fast rolling. See Extension 1 for the video demonstration.

rolling). Figure 13 compares the tracking performance among the following three methods: (i) image-only affine photometric model (IA = No gyro + Affine), (ii) gyro-aided translational model (GT = Gyro + Translation) and (iii) gyro-aided affine photometric model (GA = Gyro + Affine). For the sake of comparison, if features were lost, the Harris corner detector made each frame consistently start with a total of 150 features.

For slow camera rotation, there is no difference in tracking performance between IA and GA, as shown in Figure 13(a). Instead, GT is significantly inferior to GA during camera rolling since GT has no warping model for template rotation. For fast camera rotation, Figure 13(b) shows the clear effect of gyro fusion in terms of the number of lost features. Whenever pan or tilt rate is higher than 2 rad s^{-1} (corresponding to 20 pixels in translation), and roll rate is higher than 2.5 rad s^{-1} (corresponding to 4° in rotation), the number of lost features in IA becomes apparent. In contrast, GA maintains a consistent tracking performance that only loses 10–20% of features even at high angular rates. The prediction obtained by the gyroscopes restrains initial warping errors within 10 pixels and 1°, corresponding to a 90% probability region of the convergence area in Figure 10. The flat peaks of gyro roll rates indicate that its maximum range is exceeded.

Figure 14 shows the distribution of feature translations rescued by GA in the desk scene. In other words, IA fails to track these feature motions but GA is able to track them. The empty hole at the center illustrates IA's translational convergence region of around a 15 pixel radius, while GA can cover up to a 60 pixel translation. To evaluate long-term tracking capability, Figure 15 compares the tracking length of IA and GA in terms of an accumulated histogram. In this comparison, no new features are added once 300 features are registered at the first frame. GA always has a higher percentile of tracked features at any tracking length. At a 50-frame length, half of the features still survive in GA but only one quarter of the features are left in IA.

The performance of GT and GA is almost identical during pan/tilt motions but GT becomes significantly worse when roll motion is involved. This is because a translational warping model is not able to respond to out-of-plane camera rotation. Figure 12 shows all of the degrees of freedom of the affine model warping when the camera moves forward with rolling. The templates translate, rotate and scale down as the camera approaches the desk, and get sheared as the angle to the desk plane becomes oblique.

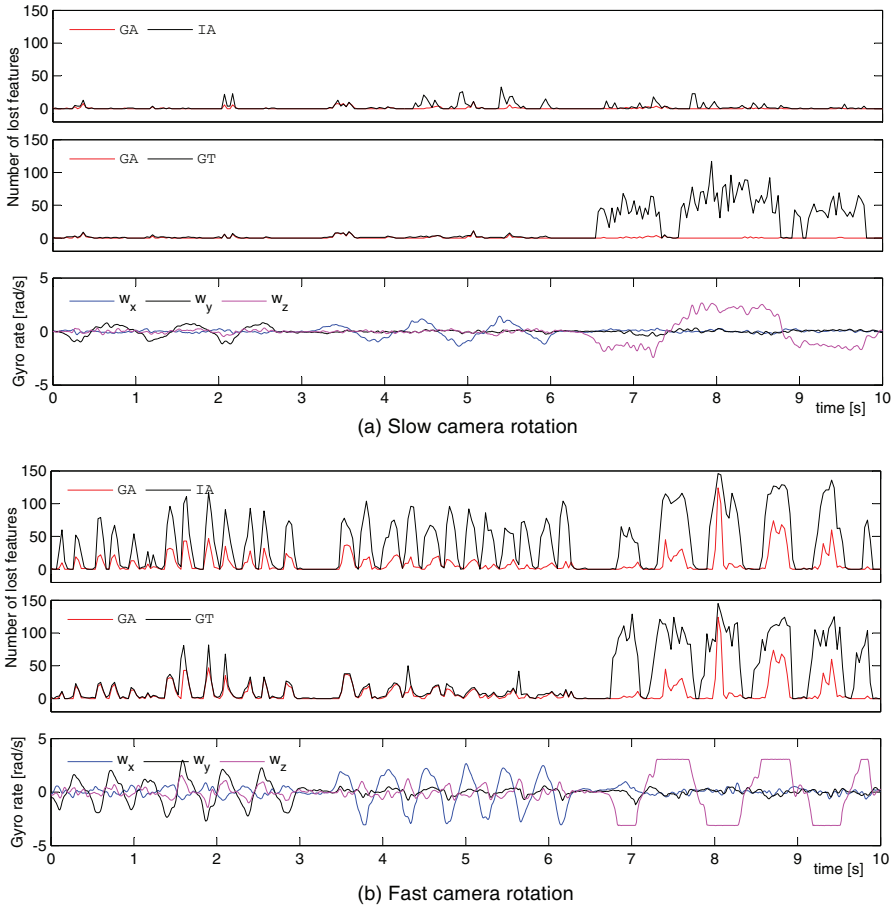

Fig. 13. Performance comparison in the desk scene between three KLT methods that use image-only affine photometric model (IA), gyro-aided translational model (GT), and gyro-aided affine photometric model (GA), respectively. When features are lost, new features are added so that every frame always starts with a total of 150 features. GA and IA has no noticeable difference in the slow rotation step, but apparently GA has superior performance to IA whenever the gyro rates reach peaks during fast rotation. GT loses more features than GA whenever camera rolling occurs.

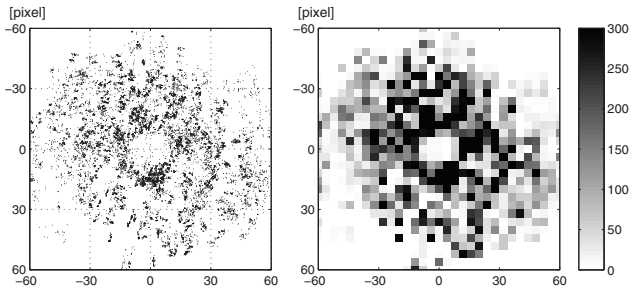

Fig. 14. Inter-frame translations of features that the gyro-aided KLT (GA) can track but the image-only KLT (IA) cannot. The distribution of feature point translations (left) and its corresponding histogram (right) in the entire desk scene. The empty hole of around 10-pixel radius at the center indicates the translational convergence region for IA. Gyro fusion enables GA to cover up to a 60 pixel feature translation.

For the aerial outdoor scene, the tracking results of IA and GA are presented in Figure 16. All of the scene points were at least 30 m away from the camera. Although feature tracking is not as accurate as that of the desk scene due to its low image resolution, GA provides more robust optical flows than IA in fast rotational motions. See Extension 1 and 2 for video demonstrations of both experimental scenes.

6.4. Camera/gyro calibration

From the desk and aerial scenes, we selected 30 camera motion pieces ($n = 30$) from each scene, either in a steady rotation or at a local maxima of motion speed. Feature correspondences were collected over two frames for the desk scene and four frames for the aerial scene. Some examples of feature tracks are shown in Figures 17

Table 1. Auto-calibration results on the desk scene experiment: a total of 30 homographies were used ($n = 30$). The final parameter estimates after the refinement had an 8.2% focal length error, 9.1% principle point error, $4.2 \pm 8.2\%$ error for three gyro scales ($\|\mathbf{s}_i\|$), a $1.1 \pm 10.6\%$ error for three gyro alignment angles ($\angle \mathbf{s}_i \mathbf{s}_j$).

	$\mathbf{K}$ (640×480)	$\mathbf{S}$	Euler ($\mathbf{R}_{ic}$) (degrees)
Ground-truth	$\begin{bmatrix} 570.25 & 0.01 & -10.59 \\ 0 & 569.32 & -22.00 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 298.51 & 0 & 0 \\ 0 & 298.51 & 0 \\ 0 & 0 & 298.51 \end{bmatrix}$	$\begin{bmatrix} 0.00 \\ 0.00 \\ 0.00 \end{bmatrix}$
Linear	$\begin{bmatrix} 550.62 & 294.88 & -21.95 \\ 0 & 677.27 & -15.57 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 261.13 & -11.93 & 65.89 \\ 0 & 326.14 & 42.73 \\ 0 & 0 & 288.11 \end{bmatrix}$	$\begin{bmatrix} -3.48 \\ 6.75 \\ -13.78 \end{bmatrix}$
Linear*	$\begin{bmatrix} 645.86 & 0 & 0 \\ 0 & 645.86 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 292.03 & 0 & 0 \\ 0 & 292.03 & 0 \\ 0 & 0 & 292.03 \end{bmatrix}$	$\begin{bmatrix} 0.09 \\ 4.55 \\ -0.35 \end{bmatrix}$
Refinement	$\begin{bmatrix} 615.79 & 2.93 & -11.09 \\ 0 & 616.85 & -0.01 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 258.23 & -30.04 & 27.23 \\ 0 & 300.82 & 4.94 \\ 0 & 0 & 296.11 \end{bmatrix}$	$\begin{bmatrix} -0.25 \\ 5.74 \\ 2.12 \end{bmatrix}$

Linear* refers to single-parameter models, $\mathbf{K}^* = \text{diag}(f, f, 1)$ and $\mathbf{S}^* = \text{diag}(s, s, s)$.

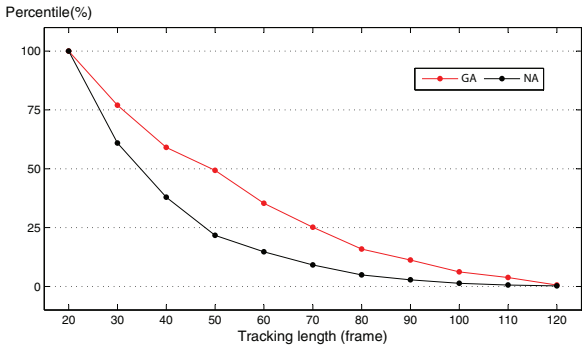

Fig. 15. Accumulated histogram of feature tracking length: 300 features are selected at the first frame and no new features are added. The percentile of each tracking length indicates how many more features are tracked compared to a given length. The gyro-aided KLT (GA) has a higher percentile at any tracking length than the image-only KLT (IA). For example, 50% of the features are tracked in at least 50 frames by GA but only in 33 frames by IA.

and 18. The distance ratio ρ is roughly 20 for the desk scene ($\|\mathbf{t}\| = 10$ cm, $d = 2$ m) and 50 for the aerial scene ($\|\mathbf{t}\| = 2$ m, $d = 100$ m). The ground truth of $\mathbf{K}$ is obtained using a planar checkered board by the DLR camera calibration toolbox (Sepp and Fuchs 2010), while that for $\mathbf{S}$ is obtained from a product datasheet. The ground truth of $\mathbf{R}_{ic}$ is measured from a mechanical drawing of the sensor plate in Figure 9 and corresponds to a zero angle.

Tables 1 and 2 show automatic calibration results of the desk and aerial scene obtained from linear methods (24), (26) and total refinement (27), respectively. As expected from the simulation results in Section 5, the refinement step significantly increases calibration accuracy compared with initial linear solutions. Note that the bias $\mathbf{v}$ is excluded in the total refinement, since $n = 30$ is not sufficient for preventing the overfitting of increased parameters. In the aerial scene, the linear camera calibration fails in Cholesky decomposition due to the high noise present in the homographies which are derived from feature tracks in low-resolution images. The alignment in the gyro shape $\mathbf{S}$ is relatively less accurate than other parameters because the quality of calibration inputs is limited; rotation angles from homographies are slightly biased and unsteady motions during Δt cause the gyro measurements to become more noisy than the feature tracks. Based on the ability of single-parameter model $\mathbf{S}^*$ to return a smaller calibration error, we can deduce that the calibration dataset is not informative enough to precisely recover the gyro's internal alignment.

It is noteworthy that camera motions in desk and aerial scenes are not specifically intended for the purpose of calibration, but for typical use of a hand-held device and aerial robot during normal operation. This fact clearly demonstrates that our method works for on-the-fly calibration when it is necessary to run gyro-aided feature tracking with no calibration priors in real scenarios. A calibration dataset collected in a principled way as the simulation input would contain more informative constraints and further improve the calibration accuracy.

Nonetheless, prediction residuals, as shown in Figures 17 and 18, demonstrate that a sufficient level of calibration accuracy is provided for gyro-aided tracking. After the calibration results are plugged back into the camera–ego motion compensation (16)–(17), translational prediction

Table 2. Auto-calibration results on the aerial scene experiment: a total of 30 homographies were used ($n = 30$). The final parameter estimates after the refinement have a 5.3% focal length error, 36% principle point error, a 1.1 ± 10.9% error for three gyro scales ($\|\mathbf{s}_i\|$), 7.4 ± 13.7% error for three gyro alignment angles ($\angle \mathbf{s}_i \mathbf{s}_j$). The Cholesky decomposition failed due to high noise in the homographies.

	$\mathbf{K}$ (320 × 240)	$\mathbf{S}$	Euler ($\mathbf{R}_{ic}$) (degrees)
Ground-truth	$\begin{bmatrix} 286.34 & 0.01 & 5.02 \\ 0 & 286.53 & -0.72 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 298.51 & 0 & 0 \\ 0 & 298.51 & 0 \\ 0 & 0 & 298.51 \end{bmatrix}$	$\begin{bmatrix} 0.00 \\ 0.00 \\ 0.00 \end{bmatrix}$
Linear	Cholesky fails	$\begin{bmatrix} 265.10 & -108.67 & -76.82 \\ 0 & 239.63 & -35.81 \\ 0 & 0 & 245.34 \end{bmatrix}$	-
Linear*	$\begin{bmatrix} 208.69 & 0 & 0 \\ 0 & 208.69 & 0 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 281.83 & 0 & 0 \\ 0 & 281.83 & 0 \\ 0 & 0 & 281.83 \end{bmatrix}$	$\begin{bmatrix} -3.45 \\ 1.94 \\ -8.06 \end{bmatrix}$
Refinement	$\begin{bmatrix} 270.38 & 3.58 & -2.19 \\ 0 & 272.28 & -44.48 \\ 0 & 0 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 259.23 & 13.01 & -78.94 \\ 0 & 274.06 & -56.90 \\ 0 & 0 & 285.88 \end{bmatrix}$	$\begin{bmatrix} 2.50 \\ 0.79 \\ -9.03 \end{bmatrix}$

Linear* refers to single-parameter models, $\mathbf{K}^* = \text{diag}(f, f, 1)$ and $\mathbf{S}^* = \text{diag}(s, s, s)$.

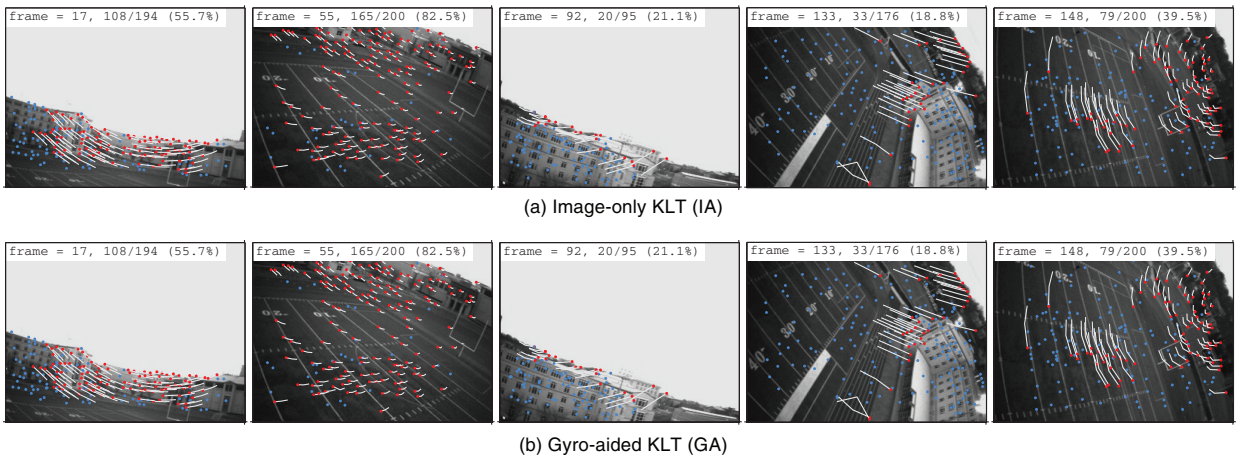

Fig. 16. Performance comparison between (a) image-only and (b) gyro-aided KLTs in the 320 × 240 aerial scene: the text in the images shows the ratio between the tracked and total number of features, and the corresponding success rate. GA achieves more robust tracking performance with higher success rates than IA. Tracked feature points are colored in red and lost features are in blue. White tails represents optical flows over the last two frames. See Extension 2 for the video demonstration.

warp residuals are within 15 and 10 pixels for the desk and aerial scene, respectively. These residuals are equal to initial conditions for the gyro-aided KLT and belong to a 75% probability zone of the convergence region of our KLT implementation in Figure 10.

6.5. GPU implementation

We use a GPU to accelerate the KLT to cope with high computational complexity of the affine-photometric warping model (7). A major bottleneck in the registration step is the Hessian inverse ($\mathcal{O}(n^3)$, where n is the number of parameters) and that in the tracking step is the warping parameter update ($\mathcal{O}(n^2)$). The source code written in the CUDA (Compute Unified Device Architecture) framework and the datasets used in Extension 1 and 2 are available at http://www.cs.cmu.edu/~myung/IMU_KLT.

Figure 19 shows the computation time tested for the desk scene. The test started with 512 features and inserted new features whenever the number of tracked features drops below 400. We use five levels of an image pyramid and a 21 × 21 template at every pyramid level. The

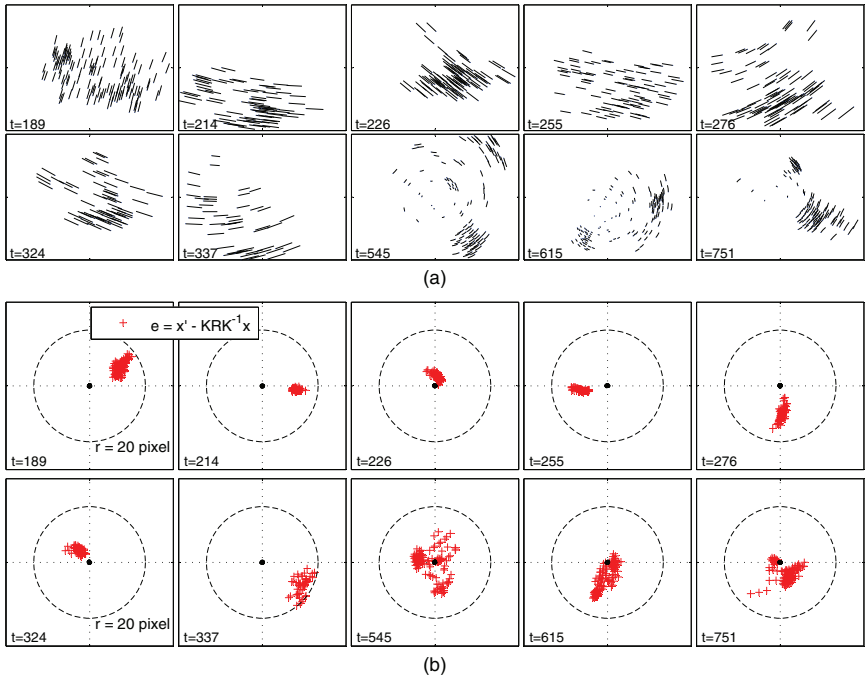

Fig. 17. Desk scene experiment (640 × 480): calibration inputs and gyro-aided prediction residuals (10 samples among 30 inputs). (a) Correspondence inputs for the RANSAC-based homography estimation: features were tracked over two image frames ($\Delta t = 0.067$ s). (b) Translational prediction warp residuals of final calibration estimates. It is equal to the reprojection error of homography predicted by gyro fusion, $e = \mathbf{x}' - \mathcal{H}\mathbf{x} = \mathbf{x}' - \mathbf{K}\mathbf{R}_{ic}\mathbf{R}_{\text{gyro}}\mathbf{K}^{-1}\mathbf{x}$.

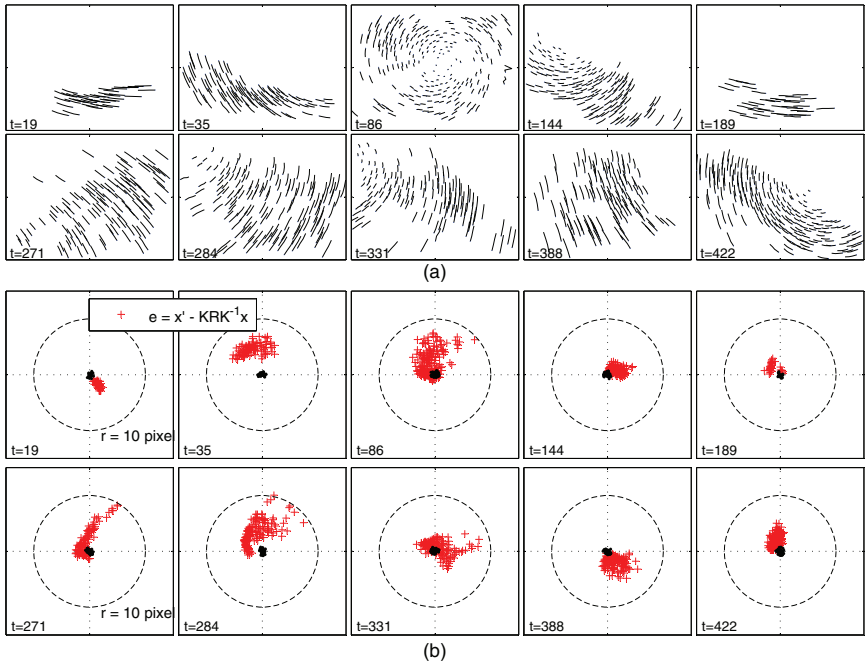

Fig. 18. Aerial scene experiment (320 × 240): calibration inputs and gyro-aided prediction residuals (10 samples among 30 inputs). (a) Correspondence inputs for the RANSAC-based homography estimation: features were tracked over four image frames ($\Delta t = 0.267$ s). (b) Translational prediction warp residuals of final calibration estimates. It is equal to the reprojection error of homography predicted by gyro fusion, $e = \mathbf{x}' - \mathcal{H}\mathbf{x} = \mathbf{x}' - \mathbf{K}\mathbf{R}_{ic}\mathbf{R}_{\text{gyro}}\mathbf{K}^{-1}\mathbf{x}$.

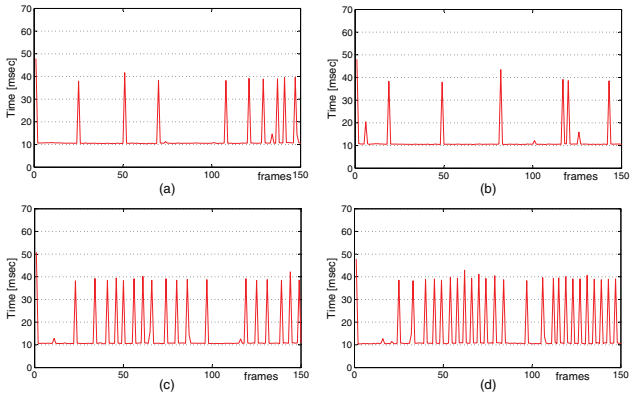

Fig. 19. Computation time of affine-photometric GPU-KLT tracking 512 features in the desk scene (tested on an NVIDIA GeForce 8800 GTX). The spikes correspond to the additional step for new feature registration occurred when 20% of features are lost. (a) Camera panning. (b) Camera tilting. (c) Camera rolling. (d) Rolling with forward move.

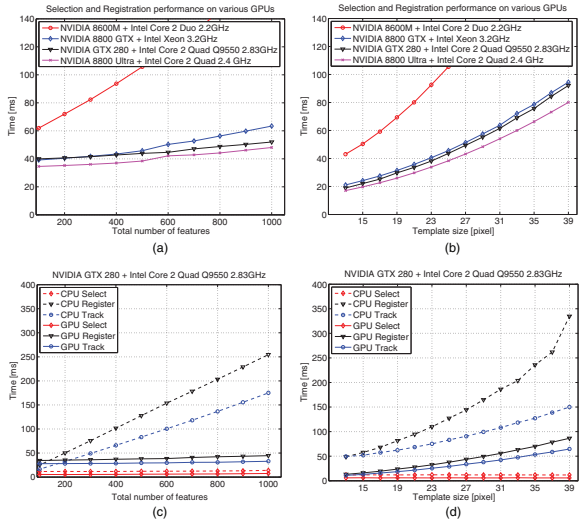

Fig. 20. Performance comparison between GPU and CPU implementations of the affine-photometric KLT in the registration and tracking steps as the number of features and template size increase (25×25 template and 512 features when they were fixed in the comparison). More detailed results are shown in Kim et al. (2009). (a) Registration versus number of features. (b) Registration versus template size. (c) Tracking versus number of features. (d) Tracking versus template size.

tracking step consistently takes 10 ms per frame. The worst consumed time for the registration step takes about 40 ms. On average, our implementation processes 63.3, 50.3, and 53.9 FPS (frames per second) during panning/tilting, random shake, and forward moving with rolling, respectively.

In Figure 20(c), tracking/registration times on the GPU remain nearly flat regardless of how many features are tracked while those on the CPU increase linearly as the number of features increases up to 1,024. Figure 20(d) shows that the computation time slightly increases linearly with the template size while that on the CPU increases quadratically. See Kim et al. (2009) for further discussion on implementation issues and the performance comparison between GPU and CPU implementations.

7. Conclusion

In this paper, we have demonstrated the effectiveness of gyro fusion in feature tracking when camera–ego motion is dominant. The knowledge of the camera's inter-frame motion can compensate for large hopping of true warping parameters. Initial warping parameters revised by instantaneous gyro rotation greatly increase the convergence rate of subsequent nonlinear optimization of KLT. Robust tracking performance has been clearly demonstrated in hostile experimental situations, such as under the conditions of heavy camera shaking or fast camera rolling.

Compared with previous work involved with sophisticated state estimation from inertial sensors, our fusion structure is a concise but efficient form in which the parameter update is tightly coupled with raw gyro measurement in a short inter-frame interval. With this feature, the structure can be easily plugged into existing image-only methods without any severe modification. In contrast with other KLT implementations, we use the affine-photometric model, which is able to deal with illumination change and is also more suitable for integration with full 3D camera motion in sensor fusion. Despite significantly higher computational burden than that of a translation-only warping model, GPU-based parallel processing enables video-rate tracking of up to 1,000 features.

We also presented an automatic online calibration of a camera/gyroscope system. The benefit is that no prior knowledge of sensor configuration is necessary and *in-situ* or on-the-fly calibration capability is provided, since tracks of natural landmarks are sufficient for a calibration dataset. The assumption of a purely rotating camera is apparently validated in the simulation in terms of the distance ratio between landmark depth and inter-frame camera translation. When the distance ratio is larger than 20, the calibration parameters from experimental data show an error rate of less than 10% and maintain the majority of prediction residuals in the KLT convergence region.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

Analog Devices (2010) *ADXRS-623 ±150°/sec MEMS Yaw Rate Gyroscope (Data Sheet)*.

Arun KS, Huang TS and Bolstein SD (1987) Least-squares fitting of two 3-D point sets. *IEEE Trans Pattern Anal Machine Intell* 9: 698–700.

Baker S and Matthews I (2004) Lucas–Kanade 20 years on: A unifying framework. *Int J Comput Vision* 56: 221–255.

Bleser G, Wohlleber C, Becker M and Stricker D (2006) Fast and stable tracking for AR fusing video and inertial sensor data. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, Plzen, Czech Republic, pp. 109–115.

Bouguet J (2000) Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm. Technical Report, Intel Corporation, Microprocessor Research Labs, OpenCV Documents.

Chandaria J, Thomas G, Bartczak B, Koeser K, Koch R, Becker M, et al. (2007) Realtime camera tracking in the MATRIS project. *SMPTE Motion Imaging J* 116: 266–271.

Corke P, Lobo J and Dias J (2007) An introduction to inertial and visual sensing. *Int J Robotics Res* 26: 519–535.

Davison A, Reid I, Molton N and Stasse O (2007) MonoSLAM: Real-time single camera SLAM. *IEEE Trans Pattern Recognition Machine Intell* 29: 1052–1067.

El-Sheimy N, Hou H and Niu X (2008) Analysis and modeling of inertial sensors using Allan variance. *IEEE Trans Instrumentation Meas* 57: 140–149.

Golub GH and Loan CFV (1996) *Matrix Computations*, 3rd edn. Baltimore, MD: The Johns Hopkins University Press.

Gray J and Veth M (2009) Deeply-integrated feature tracking for embedded navigation. In *ION International Technical Meeting Program*, Anaheim, CA.

Hartley R (1997) Self-calibration of stationary cameras. *Int J Comput Vision* 22: 5–23.

Hartley RI and Zisserman A (2004) *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge: Cambridge University Press.

Hedborg J, Skoglund J and Felsberg M (2007) KLT tracking implementation on the GPU. In *Proceedings SSBA 2007*, Linkoping, Sweden.

Hol JD, Schön TB and Gustafsson F (2010) Modeling and calibration of inertial and vision sensors. *Int J Robotics Res* 29: 231–244.

Hol JD, Schön TB, Luinge H, Slycke PJ and Gustafsson F (2007) Robust real-time tracking by fusing measurements from inertial and vision sensors. *J Real-Time Image Process* 2: 149–160.

Hwangbo M and Kanade T (2008) Factorization-based calibration method for mems inertial measurement unit. In *Proceedings IEEE International Conference on Robotics and Automation*, San Hose, CA.

Hwangbo M, Kim J-S and Kanade T (2009) Inertial-aided klt feature tracking for a moving camera. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09)*, pp. 1909–1916.

Jin H, Favaro P and Soatto S (2001) Real-time feature tracking and outlier rejection with changes in illumination. In *International Conference on Computer Vision*, pp. 684–689.

Kelly J and Sukhatme GS (2008) Fast relative pose calibration for visual and inertial sensors. In *11th International Symposium Experimental Robotics (ISER'08)*.

Kim J-S, Hwangbo M and Kanade T (2009) Realtime affine-photometric KLT feature tracker on GPU in CUDA framework. In *The Fifth IEEE Workshop on Embedded Computer Vision in ICCV 2009*, pp. 1306–1311.

Lang P and Pinz A (2005) Calibration of hybrid vision / inertial tracking system. In *Proceedings 2nd Workshop on Integration of Vision and Inertial Sensors (INERVIS'05)*.

Lobo J and Dias J (2003) Vision and inertial sensor cooperation using gravity as a vertical reference. *IEEE Trans Pattern Recognition Machine Intell* 25: 1597–1608.

Lobo J and Dias J (2007) Relative pose calibration between visual and inertial sensors. *Int J Robotics Res* 26: 561–575.

Lucas BD and Kanade T (1981) An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, Canada, pp. 674–679.

Makadia A and Daniilidis K (2005) Correspondenceless ego-motion estimation using an IMU. In *Proceedings IEEE International Conference on Robotics and Automation*, pp. 3534–3539.

Mirzaei FM and Roumeliotis SI (2008) A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Trans Robotics* 24: 1143–1156.

Ohmer J and Redding N (2008) GPU-accelerated KLT tracking with monte-carlo-based feature reselection. In *Computing: Techniques and Applications, 2008. DICTA '08. Digital Image*, pp. 234–241.

Sepp W and Fuchs S (2010) *DLR CalLab and CalDe—The DLR Camera Calibration Toolbox*. Available at: http://www.robotic.dlr.de/callab/.

Shi J and Tomasi C (1994) Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA.

Sinha SN, Frahm J-M, Pollefeys M and Genc Y (2007) Feature tracking and matching in video using programmable graphics hardware. In *Machine Vision and Applications*.

Yokokohji Y, Sugawara Y and Yoshikawa T (2000) Accurate image overlay on video see-through HMDs using vision and accelerometers. In *Proceedings IEEE Virtual Reality*.

You S, Neumann U and Azuma R (1999) Hybrid inertial and vision tracking for augmented reality registration. In *Proceedings IEEE Virtual Reality*.

Zach C, Gallup D and Frahm J-M (2008) Fast gain-adaptive KLT tracking on the gpu. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPR Workshops 2008*, pp. 1–7.

Zhang C, Chockalingam P, Kumar A, Burt P and Lakshmikumar A (2008) Qualitative assessment of video stabilization and mosaicking systems. In *IEEE Workshop on Applications of Computer Vision (WACV 2008)*, pp. 1–6.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Indoor desk scene (640 × 480 at 30 FPS)experiment for gyro-aided feature tracking as various types of rapid and random camera rotation are imposed.
2	Video	Outdoor aerial scene (320 × 240 at 15 FPS) experiment for gyro-aided feature tracking as a fixed-wing model airplane maneuvers in an urban area.