2010 11th. Int. Conf. Control, Automation, Robotics and Vision
Singapore, 5-8th December 2010

Omnidirectional Video Stabilisation On a Virtual Camera Using Sensor Fusion

Thomas Albrecht*, Tele Tan†, Geoff A.W. West‡, and Thanh Ly§
*Department of Computing
†DEBI Institute
‡Department of Spatial Sciences
Curtin University of Technology, Western Australia
§Defence Science and Technology Organisation, Australia

***Abstract*—This paper presents a method for robustly stabilising omnidirectional video given the presence of significant rotations and translations by creating a virtual camera and using a combination of sensor fusion and scene tracking. Real time rotational movements of the camera are measured by an Inertial Measurement Unit (IMU), which provides an initial estimate of the ego-motion of the camera platform. Image registration is then used to refine these estimates. The calculated ego-motion is then used to adjust an extract of the omnidirectional video, forming a virtual camera that is focused on the scene. Experiments show the technique is effective under challenging ego-motions and overcomes deficiencies that are associated with unimodal approaches making it robust and suitable to be used in many surveillance applications.**

***Index Terms*—Omnivision, Camera Stabilisation, Virtual Camera, Sensor Fusion, Active Vision**

I. Introduction

Video-based surveillance systems have become prevalent in modern day society as a result of the increased security threat. Traditionally, surveillance systems use multiple fixed or pan-tilt-zoom (PTZ) cameras to enable observation of areas and/or personnel. Fixed and PTZ cameras have only limited fields of view at any one time and thus require several cameras to cover larger urban areas. To address this, mobile platforms using omnidirectional cameras have been introduced. Mobile omnidirectional camera platforms minimise the need for significant surveillance infrastructure as a single camera system can cover a vast area. They also increases the operational versality with the overall improvement to situational awareness.

One of the major issues with mobile platforms is video stabilisation. In surveillance, the stabilisation is typically in relation to the target, which means that a target, once acquired, should always stay in the same position within the view, regardless of the platform's ego-motion. Ego-motion of a platform is a combined motion of rotation and translation. To compensate for ego-rotation, a traditional approach is the use of hydraulic or electro-mechanic tilt platforms on which the camera system is mounted. The platform is tilted based on the measurement of an accelerometer or gyroscope, contained within an Inertial Measurement Unit (IMU). If ego-translation is present, the target's bearing changes. To compensate for this, the translational component needs to be estimated. However, affordable IMUs, implemented as microelectromechanical systems (MEMS) cannot measure translational motion due to errors introduced by double integration of the accelerometer readings. Digital image stabilisation resolves this by registering features [1] across frames over time, finding the optimum affine transformation that minimises the error between the features, and applying it to the subsequent frames. This can be computationally expensive as the search window has to be potentionally very large, especially if camera ego-rotation is occurring since this can easily cause the field-of-view to move drastically. When using omnidirectional cameras, the correspondence between the cameras needs to be fused into a single image domain. This can be done using projections like the common cylindrical panorama, fisheye, or cubical panorama that are used to map full 3D world scenes onto 2D images [2]. The most intuitive representation of 3D world data is the spherical projection with the omnidirectional camera situated in the centre of the sphere, because it represents the scene as it is captured; it has recently been made popular by Google's Streetview Services. Mapping all images onto a unit sphere yields a representation in a continuous 3D domain. However, well-established imaging algorithms work with 2D datasets and cannot be easily adapted to function with 3D datasets. In many situations, it is not necessary to process the entire spherical view as the region of interest, at any one time, typically only occupies a small fraction of the overall view. These regions of interest can be very efficiently extracted using appropriate inverse projection techniques [3]. Processing in this fashion attracted much interest in the late 1980s and early 1990s and went by the general term *Active Vision* [4].

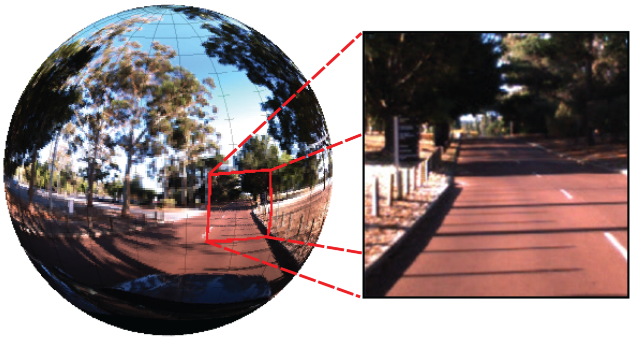


Fig. 1. Virtual Camera extracted from omnidirectional video.

Being able to selectively choose where to look and what to look at was recognised as a way to speed up computerised scene and image interpretation. Many systems were built using motor-powered and servo-controlled cameras and some impressive results were achieved, but it was recognised that high performance was needed for such systems to function e.g. a rotational speed of $500°s^{-1}$ and an acceleration of $5000°s^{-2}$ [5], which was expensive to obtain. However, using an omnidirectional camera overcomes these constraints as the camera system does not have any moving parts that restrict the angular velocity.

We propose a novel technique of sensor fusion for scene stabilised viewing of a target, where stabilisation is robust to both rotation and translation of the capturing platform, by extracting a region with a limited field-of-view from the omnidirectional video, effectively a '*virtual camera*' (Figure 1). The use of a virtual camera further reduces the search window size of the feature tracker. Using sensor fusion of an IMU and a particle filtered feature tracker, the orientation of the virtual camera is subsequently adjusted to keep the target centred in the view. We show that the fusion of omnidirectional video and IMU data increases tracking accuracy and eliminates unrecoverable tracking failures even during fast ego -motion. Our technique is also able to use hardware devices that are not closely synchronised in terms of their frame capture.

This paper is organised as follows: section II discusses related work, whilst in section III virtual cameras and the approach is described. The experimental methodology and the results are discussed in section IV with a conclusion given in section V.

II. Related Work

Video stabilisation has been extensively studied, e.g. Yang and Schonfeld [6] described video stabilisation using a particle filter based on SIFT [7] features, Battiato et al. [8] reported a video stabilisation systems based on tracking SIFT features through consecutive frames. cameras. However, both systems stabilise the image using features of the entire image, and do not stabilise on a single object. A tracking approach, where the stabilisation is emphasised on a target is presented by Kumar et al. [9], who used a PTZ camera and a CAMSHIFT [10] filter to track an object within a frame. Their system mechanically adjusts the orientation of the camera platform according to the position of the target object. Sun et al. [12] used a virtual camera to detect and track a person in a wide angle panoramic video; designed for indoor lecture halls, the camera was kept static at all times. Quaritsch et al. [13] reported a multicamera tracking system where targets are handed over between cameras. A system for omnidirectional image stabilisation by computing the 3D camera trajectory using a structure from motion approach was described by Torii et al. [11]; based on just image features, the system needs to process the whole image as the view point can change drastically between consecutive frames. A sensor fused approach using gravity as a vertical reference for 3D-mapping of planar image points for robot guidance is described by Lobo and Dias [14]; the approach relies on the estimation of vanishing lines that are difficult to obtain in outdoor scenes. Armesto et al. [15] used a Kalman filter to fuse vision and inertial measurements for tracking ego-motion of only a monocular-camera.

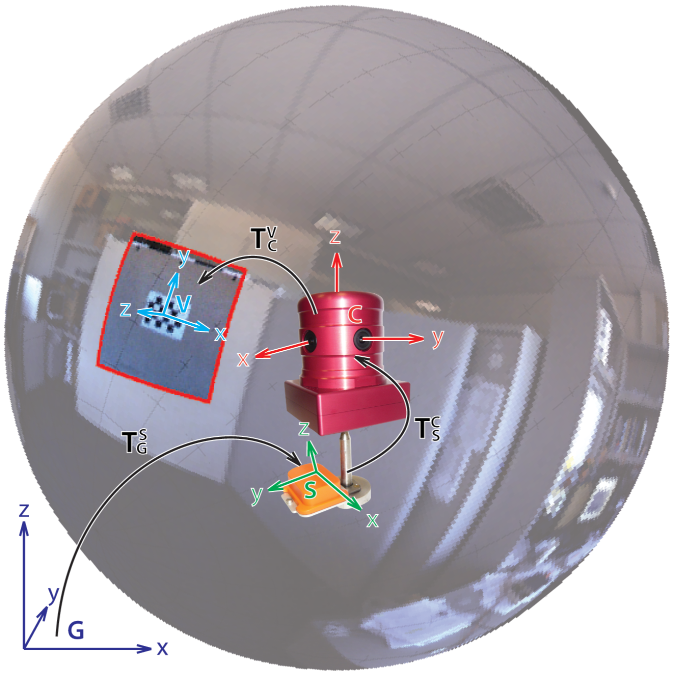


Fig. 2. **System Platform.** Augmented with the coordinate systems of the virtual camera (V), the omnidirectional camera (C), the sensor (S), and the global coordinate system (G). Note that $\mathbf{T}_S^C$ is fixed as camera and sensor are rigidly connected. The size of the virtual camera is defined by its field-of-view, α_t, and vertical resolution, w_t at time step t.

III. System Description

We utilise the Ladybug 2, an omnidirectional camera manufactured by Point Grey Research and an IMU, MTi, manufactured by Xsens. The Ladybug 2 camera consists of six individual cameras each capturing 1024×768 pixels at 30fps. Five cameras are horizontally aligned in a ring, with the sixth pointing upwards. In this setup the system can capture about 80% of the whole sphere. It is pre calibrated, and the geometry between the cameras is provided by the manufacturer. This allows for fast and precise spherical mapping, negating the need to register all images. The MTi IMU has acceleration, gyroscopic, and magnetic sensors that are fused using a hardware-based Kalman filter. It outputs calibrated measurements of acceleration and angular velocity, as well as a drift-free 3D orientation with a static accuracy of $\leq 1.0°$, at a maximum sample rate of 100Hz.

A. Notation

In this paper, the following notation is used: a point in a coordinate system (A) is denoted as p^{A}, features and objects as f^{A} and o^{A} respectively. The orientation of A with respect to another coordinate system (B) is denoted as the 3×3 rotation matrix $\mathbf{R}_{\mathrm{B}}^{\mathrm{A}}$. The translational offset between A and B is denoted as the 3×1 vector $\mathbf{t}_{\mathrm{B}}^{\mathrm{A}}$ of A in respect to B. Both are combined using homogeneous coordinates resulting in the 4×4 transformation matrix $\mathbf{T}_{\mathrm{B}}^{\mathrm{A}}$

$$\mathbf{T}_{\mathrm{B}}^{\mathrm{A}} = \begin{pmatrix} \mathbf{R}_{\mathrm{B}}^{\mathrm{A}} & \mathbf{t}_{\mathrm{B}}^{\mathrm{A}} \\ 0 & 1 \end{pmatrix}. \tag{1}$$

The inverse transformation $(\mathbf{T}_{\mathrm{B}}^{\mathrm{A}})^{-1}$ is subsequently defined as $\mathbf{T}_{\mathrm{A}}^{\mathrm{B}}$.

B. Coordinate Systems

Four coordinate systems are used (Figure 2):

1) the global coordinate system (G),
2) the sensor coordinate system (S), which is defined for the IMU,
3) the camera coordinate system (C) is defined with respect to S as the camera and IMU are rigidly attached, its origin lies in the centre of the camera, and
4) the virtual camera coordinate system (V), which is defined at the centre of a virtual camera and defined with respect to C (sec. III-C).

The MTi sensor outputs its orientation as the rotation between G, S, and time t, denoted as the homogeneous transformation $\mathbf{T}_{\mathrm{G},t}^{\mathrm{S}}$. With the fixed relative transformation between S and C, $\mathbf{T}_{\mathrm{S}}^{\mathrm{C}}$, the transformation of a point p^{C} into p^{G} is

$$p^{\mathrm{G}} = \mathbf{T}_{\mathrm{S},t}^{\mathrm{G}}\, \mathbf{T}_{\mathrm{C}}^{\mathrm{S}}\, p^{\mathrm{C}}. \tag{2}$$

C. Virtual Camera

A virtual camera (Figure 1) is a sub-window extracted from a full spherical view. The virtual camera is defined by its orientation $\mathbf{R}_{\mathrm{C},t}^{\mathrm{V}}$ with respect to C, the vertical resolution w_t, and the field-of-view α at time step t. Applying the perspective projection with z_1 and z_2 as the projection's far and near clipping respectively yields the transformation from C to the virtual camera coordinates V as $\mathbf{T}_{\mathrm{C},t}^{\mathrm{V}}$

$$\mathbf{T}_{\mathrm{C},t}^{\mathrm{V}} = \begin{pmatrix} \cot\frac{\alpha_t}{2} & 0 & 0 & 0 \\ 0 & \cot\frac{\alpha_t}{2} & 0 & 0 \\ 0 & 0 & \frac{z_1+z_2}{z_2-z_1} & \frac{2\,z_1\,z_2}{z_2-z_1} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R}_{\mathrm{C},t}^{\mathrm{V}} & 0 \\ 0 & 1 \end{pmatrix}. \tag{3}$$

A point p^{V} can thus be transformed into global coordinates as

$$p^{\mathrm{G}} = \mathbf{T}_{\mathrm{S},t}^{\mathrm{G}}\, \mathbf{T}_{\mathrm{C}}^{\mathrm{S}}\, \mathbf{T}_{\mathrm{V},t}^{\mathrm{C}}\, p^{\mathrm{V}}. \tag{4}$$

D. Particle Filter Framework

We use a particle filter framework [16] to track the target object in global coordinates and use the current predicted position p_t^{G} to adjust the orientation of the virtual camera. The state -space model of the particle filter can be described with a state vector $\mathbf{x} = (x\,, y\,, z\,, \dot{x}\,, \dot{y}\,, \dot{z})^T$ containing the global 3D-projection onto the unit sphere and velocity of the target.

E. Synchronisation and Calibration

To utilise data captured by a sensor in another sensor's domain, the spatial transformation and temporal offset between both sensors needs to be known. This means that IMU and camera need to be calibrated, i.e. $\mathbf{T}_{\mathrm{S}}^{\mathrm{C}}$ needs to be estimated. We use a method proposed by [18] to estimate the relative pose and orientation. However, as the Ladybug 2 camera does not possess a hardware trigger interface, the synchronisation between the camera and the IMU is not accurate, resulting in a deviation between video frames and inertial measurements. To cope with this offset, we increase the uncertainty of the position estimation, so that our approach can also deal with an inaccurate calibration between camera and IMU.

F. Stabilisation

After selecting an object, we initialise the stabilisation process by registering features within the object's region. In subsequent time steps, stabilisation is achieved by iteratively adjusting the orientation of the virtual camera to stabilise the object at the centre of the image using sensor fusion of the IMU and the feature tracker.

1) Initialisation: At the first time step, we select an object to track by selecting a region and use the methods proposed by [19] to register good features $f_{1..N,t}^{\mathrm{V}}$, where $t = 0$ and N is the number of features registered. The position of the object o_t^{G} at time step t is then estimated using eq. (4) as the centroid formed by the features as

$$o_t^{\mathrm{G}} = \mathbf{T}_{\mathrm{V},t}^{\mathrm{G}}\, \frac{1}{N} \sum_{n=1}^{N} f_{n,t}^{\mathrm{V}}. \tag{5}$$

To centre the virtual camera onto o_t^{G}, the orientation of the virtual camera, $\mathbf{R}_{\mathrm{C},t}^{\mathrm{V}}$ can then be computed as the rotation of the unit vector $\mathbf{u}_0^{\mathrm{C}} = (1\,, 0\,, 0)^T$ towards o_t^{C} using Rodrigues' rotation formula

$$\mathbf{R}_{\tilde{\mathbf{\Omega}}}(\beta) = \mathbf{I}_3 + \sin\beta \cdot \tilde{\mathbf{\Omega}} + (1 - \cos\beta) \cdot \tilde{\mathbf{\Omega}}^2, \tag{6}$$

where the skew symmetric matrix $\tilde{\mathbf{\Omega}}$ is defined as

$$\tilde{\mathbf{\Omega}} = \begin{pmatrix} 0 & -\tilde{\omega}_3 & \tilde{\omega}_2 \\ \tilde{\omega}_3 & 0 & -\tilde{\omega}_1 \\ -\tilde{\omega}_2 & \tilde{\omega}_1 & 0 \end{pmatrix}, \text{ with } \tilde{\omega} = \mathbf{u}_0^{\mathrm{C}} \times \mathbf{o}_t^{\mathrm{C}}, \tag{7}$$

and β is the angle between both position vectors that can be computed as the l^2-norm of the cross product for small, or the $\arccos$ of the dot-product for large angles. The transformation, $\mathbf{T}_{\mathrm{C},t}^{\mathrm{V}}$, to adjust the virtual camera can then be recomputed using eq. (3).

2) Sensor Fusion: Sensor fusion is performed in three steps: first, the IMU is used to roll forward the position estimation of each feature into the next time frame. Second, the features are refined to compensate for translational movement. Third, the orientation of the virtual camera is adjusted to keep the object centred.

At time step $t+1$ the new measurement of the IMU, $\mathbf{T}_{\mathrm{G},t+1}^{\mathrm{S}}$, is evaluated. The transformation $\mathbf{T}_{\mathrm{G},t+1}^{\mathrm{V}}$ is then computed as

$$\mathbf{T}_{\mathrm{G},t+1}^{\mathrm{V}} = \mathbf{T}_{\mathrm{C},t+1}^{\mathrm{V}}\, \mathbf{T}_{\mathrm{S}}^{\mathrm{C}}\, \mathbf{T}_{\mathrm{G},t+1}^{\mathrm{S}}. \tag{8}$$

The feature positions in the virtual camera at time step $t+1$ as predicted by the IMU are then

$$f^{\mathrm{V}}_{1..N,t+1} = \mathbf{T}^{\mathrm{V}}_{\mathrm{G},t+1}\, \mathbf{T}^{\mathrm{G}}_{\mathrm{V},t}\, f^{\mathrm{V}}_{1..N,t}. \tag{9}$$

We then utilise pyramidal Lucas-Kanade tracking [20] and updating the positions of the features in V and dropping non-matched features. If the number of features drop below a certain threshold (e.g. <10), new features are estimated within the predicted region of the target object using the methods described by [19]. Using the matched feature pairs, $f^{\mathrm{V}}_{1..M,t+1}$, where M is the number of matches, the position of the target object o^{G}_{t+1} is then again computed using eq. (5) and used to update the particle filter tracking. We use the distance l along the unit sphere [17] as the error function of the measurement update of the particle filter

$$l = \arccos\left(o^{\mathrm{G}}_{t,3}\, \mathbf{x}_3^{\mathrm{G}\,(i)} + \sqrt{1-\left(o^{\mathrm{G}}_{t,3}\right)^2}\sqrt{1-\left((\mathbf{x}_3^{\mathrm{G}})^{(i)}\right)^2} \times \cos\left(\arctan\frac{o^{\mathrm{G}}_{t,2}}{o^{\mathrm{G}}_{t,1}} - \arctan\frac{(\mathbf{x}_2^{\mathrm{G}})^{(i)}}{(\mathbf{x}_1^{\mathrm{G}})^{(i)}} \right)\right). \tag{10}$$

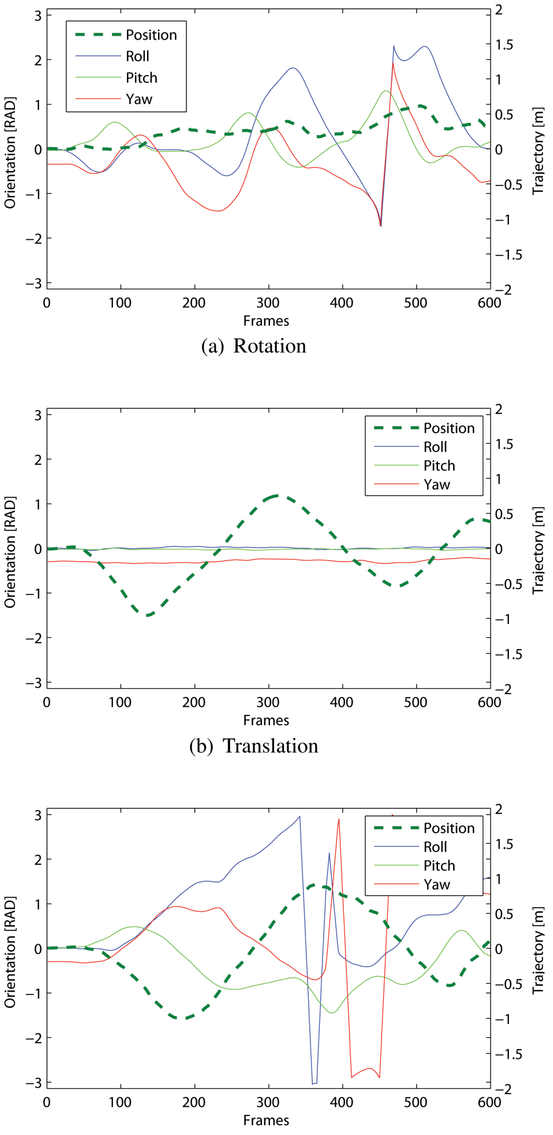
Finally, the updated position prediction of the target object o^{G}_{t+1}, as estimated by the particle filter, is used to stabilise the virtual camera by using eqs. (6) and (7) to compute the transformation, $\mathbf{T}^{\mathrm{V}}_{\mathrm{C},t+1}$, for the next time step.

IV. Experiments

The goal of the experiments was to test the robustness and to quantitatively evaluate the performance of stabilisation a virtual camera using sensor fusion of a feature tracker and IMU. We captured 9 hand-held sequences during which the recording platform was subject to three different types of ego-motion: 1. rotational, 2. translational, and 3. combined motion. Each sequence consists of 24 seconds with full resolution omnidirectional video data at 25fps and inertial data sampled at 50Hz. As a target object we chose a checkerboard because it allows precise error measurement. We compared our approach to unimodal stabilisation using IMU-only and feature-tracker-only. Furthermore, we applied our approach to a monocular camera to compare the performance of a virtual camera of an omnidirectional video to a monocular video. For that, we simulated a monocular camera by disabling all but the front camera of the Ladybug 2. To evaluate the performance of the proposed technique, we compute the error of the stabilisation as the Euclidean distance from the centre of the virtual camera as ϵ^V and convert it into a resolution independent error in G as

$$\epsilon^{\mathrm{G}}_{\theta} = \epsilon^{\mathrm{V}} \alpha_t\, w_t^{-1}. \tag{11}$$

The results listed in Table I indicate that the IMU is a reliable source for stabilising ego-rotation (Fig. 6(a)). However, the performance still does not match our fused approach because small translations have been present in the data due to the hand-held spinning and the fact that both sensors were not perfectly aligned and the lack of hardware synchronisation in the Ladybug 2 camera. However, when translational movement is dominant, the image drifts off as the IMU does not compensate for translation (Fig. 6(b)). Note the form of the curve which corresponds to the left-right movement of the platform (see the ground truth in Fig.3(b)). Figure 4(a) depicts the corresponding frames in the stabilised video. The small rectangle shows the centre of the virtual camera. The feature tracker coped very well with movement where translation was dominant (Fig. 6(b)) since the apparent motion of the scene is not large. Rotations, however, caused large shifts in th e field-of-view, exceeding the search window and causing an unrecoverable

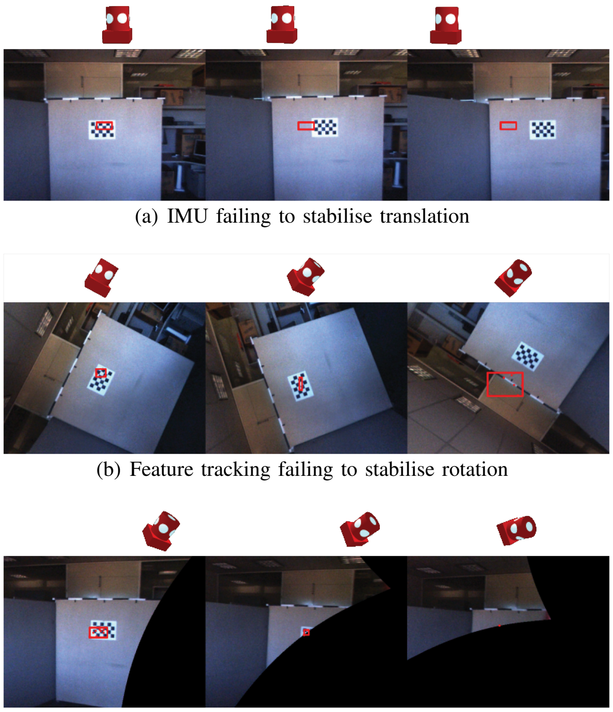


(a) Rotation

(b) Translation

(c) Combined Motion

Fig. 3. **Ground Truth.** The graphs depict the ground truth for three example test runs: (a) mainly rotation, (b) mainly translation, and (c) both rotations and translations. The position is given as the distance from the centre in meters and displayed on the right axis of the graphs. Yaw, pitch, and roll show the orientation of the platform in Euler angles on the left axis. Note, that the singularities in graph (c) are caused by the singularities that are present in the Euler notation.



(a) IMU failing to stabilise translation

(b) Feature tracking failing to stabilise rotation

(c) Monocular camera failing to stabilise out of view

Fig. 4. **Unimodal and Monocular Stabilisation.** The figures depict 2 seconds (50 frames) of the full 24 seconds of (a) translational, (b) rotational, and (c) combined motion. Unimodal stabilisation approaches are used in (a) and (b), while in (c) the sensor fused technique is applied. The relative position and orientation of the capturing platform is shown above each frame. The small rectangle indicates the predicted centre of the virtual camera.

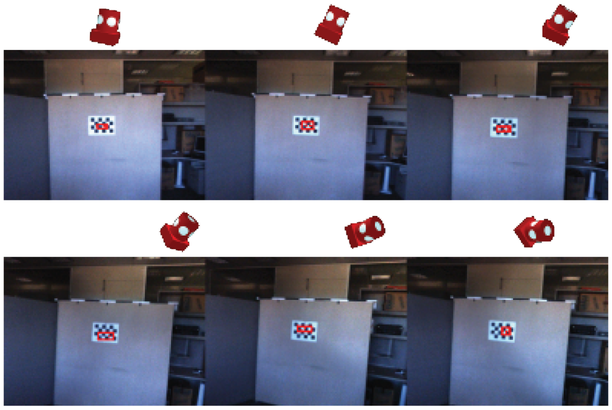
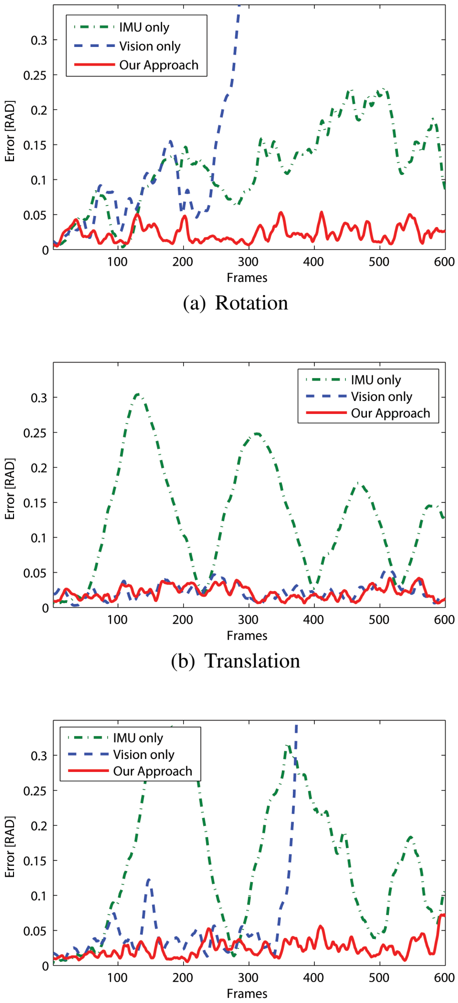


Fig. 5. **Virtual Camera with Sensor Fusion Stabilisation.** The figure depicts a 4 second (100 frames) sequence of combined motion. The stabilisation is performed using sensor fusion of the IMU and a feature tracker on a virtual camera. The relative position and orientation of the platform is show above each frame.



(a) Rotation

(b) Translation

(c) Combined Motion

Fig. 6. **Experimental Evaluation.** Graphs (a)-(c) show the error as an offset from the centre in radians measured in the global frame for the compared stabilisation techniques.

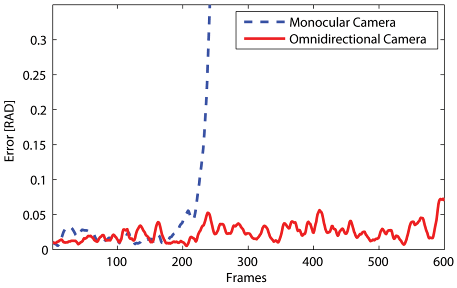


Fig. 7. **Experimental Evaluation.** The graph depicts the error for a monocular camera compared to our approach.

	1. Rotation	2. Translation	3. Combined
IMU	0.116 ±0.060	0.128 ±0.081	0.152 ±0.103
Feature-Tracker	0.088† ±0.066	0.023 ±0.012	0.045† ±0.042
Fused Monocular	-	-	0.035† ±0.053†
Our Approach	0.023 ±0.014	0.021 ±0.011	0.023 ±0.013

TABLE I
Results. The table shows the shift of the image as a relative mean error in radians. † Feature tracker unrecoverably lost the target, mean error is computed up until the lost of target.

failure at around frame 250. Figure 4(b) shows the drifting and rolling of the virtual camera in the corresponding frames. A similar problem occurred in the combined motion at frame 350 (Fig. 6(c)). Finally, when using our approach on a monocular camera (Fig. 7), the stabilisation works well until the target approaches the bounds of the field-of-view of the camera around frame 200. The corresponding frames are depict in Figure 4(c). As the field-of-view is a hardware restriction, only an omnidirectional camera can overcome the limited field-of -view. However, the computational cost of processing a whole spherical data set would be much higher than the computational cost for processing the search window of the virtual camera, which is of the same size as the monocular camera.
During the experiments the calibration and synchronisation offsets between both sensors seemed not to be an issue. Even though they presumably caused in increase in tracking inaccuracy, the particle filter assured that the tracking does not fail during fast rotations.

V. Conclusion

In this paper, we presented a method of stabilising the view of an omnidirectional camera with respect to a target object by using a virtual camera and sensor fusion. The experiments showed that the presented approach is robust and outperforms unimodal and monocular approaches. It demonstrated the benefit of sensor fusion to overcome limitations of single sensors.

Supplementary Material

Videos demonstrating the stabilisation are available online at http://www.computing.edu.au/~14133369/stabilisation.

Acknowledgment

This research is supported by the DSTO, Australia PhD scholarship program.

References

[1] B. Zitova, J. Flusser, *Image registration methods: a survey*. Image and vision computing, 2003.

[2] D. Salomon, *Transformations and projections in computer graphics*. Springer, London, 2006.

[3] S.K. Nayar, *Catadioptric omnidirectional camera*. IEEE Conference on Computer Vision and Pattern Recognition, 1997.

[4] A. Blake and A. Yuille, *Active vision*. The MIT Press Cambridge, Massachusetts, 1993.

[5] D.W. Murray, K.J. Bradshaw, P.F. McLauchlan, I.D. Reid, and P.M. Sharkey, *Driving saccade to pursuit using image motion*. International Journal of Computer Vision, 1995.

[6] J. Yang, D. Schonfeld, C. Chen, and M. Mohamed, *Online video stabilization based on particle filters*. IEEE International Conference on Image Processing, 2006.

[7] D. Lowe, *Distinctive image features from scale-invariant keypoints*. International Journal of Computer Vision, 2004.

[8] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, *SIFT Features Tracking for Video Stabilization*. International Conference on Image Analysis and Processing, 2007.

[9] P. Kumar, A. Dick and T.S. Sheng, *Real Time Target Tracking with Pan Tilt Zoom Camera*. Digital Image Computing: Techniques and Applications, 2009.

[10] G.R. Bradski, *Computer vision face tracking as a component of a perceptual user interface*. Workshop on Applications of Computer Vision, 1998.

[11] A. Torii, M. Havlena, and T. Pajdla, *Omnidirectional image stabilization by computing camera trajectory*. Advances in Image and Video Technology: Third Pacific Rim Symposium, 2009.

[12] X. Sun, J. Foote, D. Kimber, and B.S. Manjunath, *Region of Interest Extraction and Virtual Camera Control Based on Panoramic Video Capturing*. IEEE Transactions on Multimedia, 2005.

[13] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof and B. Strobl, *Autonomous Multicamera Tracking on Embedded Smart Cameras*. EURASIP Journal on Embedded Systems, 2007.

[14] J. Lobo and J. Dias, *Vision and Inertial Sensor Cooperation Using Gravity as a Vertical Reference*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003.

[15] L. Armesto, J. Tornero, M. Vincze, *Fast Ego-motion Estimation with Multi-rate Fusion of Inertial and Vision*. International Journal of Robotics Research, 2007.

[16] A. Doucet, S. Godsill, C. Andrieu, *On sequential Monte-Carlo sampling methods for Bayesian filtering*. Statistics and Computing, 2000.

[17] J.P. Snyder, *Map Projections–A Working Manual*. United States Government Printing Office, Washington, 1987.

[18] J.D. Hol, T.B. Schon and F. Gustafsson, *Modeling and Calibration of Inertial and Vision Sensors*. International Journal of Robotics Research, 2010.

[19] J. Shi and C. Tomasi, *Good Features To Track*. IEEE Conference on Computer Vision and Pattern Recognition, 1994.

[20] J.-Y. Bouguet, *Pyramidal implementation of the lucas kanade feature tracker description of the algorithm*. Intel Corporation, Microprocessor Research Labs, 2000.