Robotica, Vol. 35, Issue 9, September 2017, pp. 1864-1887
DOI:10.1017/S0263574716000552

Scale robust IMU-assisted KLT for stereo visual odometry solution

L. Chermak*, N. Aouf and M. A. Richardson

Centre of Electronic Warfare, Cranfield University, Shrivenham, SN6 8LA, UK
E-mails: n.aouf@cranfield.ac.uk, m.a.richardson@cranfield.ac.uk

(Accepted July 23, 2016)

SUMMARY
We propose a novel stereo visual IMU-assisted (Inertial Measurement Unit) technique that extends to large inter-frame motion the use of KLT tracker (Kanade–Lucas–Tomasi). The constrained and coherent inter-frame motion acquired from the IMU is applied to detected features through homogenous transform using 3D geometry and stereoscopy properties. This predicts efficiently the projection of the optical flow in subsequent images. Accurate adaptive tracking windows limit tracking areas resulting in a minimum of lost features and also prevent tracking of dynamic objects. This new feature tracking approach is adopted as part of a fast and robust visual odometry algorithm based on double dogleg trust region method. Comparisons with gyro-aided KLT and variants approaches show that our technique is able to maintain minimum loss of features and low computational cost even on image sequences presenting important scale change. Visual odometry solution based on this IMU-assisted KLT gives more accurate result than INS/GPS solution for trajectory generation in certain context.

KEYWORDS: Computer vision, Navigation, IMU-KLT feature tracking, Visual odometry, Double dogleg.

1. Introduction

The process of estimating camera motion from the analysis of feature correspondences within a sequence of images over time is called visual odometry (VO).[1] In recent years, many research works have contributed to the success and establishment of this methodology.[2–5] VO does not necessarily use computer vision techniques only. Indeed, improvement, assistance, or fusion to visual processing techniques using sensors such as an IMU (Inertial Measurement Unit) or a GPS (Global Positioning System) have a positively contributed to VO solutions. This work concentrates on feature tracking in a VO context. Accurate estimation of the camera motion during navigation depends on the ability to successfully track features over successive images. KLT (Kanade–Lucas–Tomasi) is one of the most popular features tracking technique.[6–8] KLT considers local information derived from small search windows surrounding each of the interest points. This local process constrains template image analysis in time, space, and brightness. This is acceptable for image sequences with small appearance change i.e., high frame rate, coherent motion, stable illumination, etc. As a consequence, it is relatively efficient on image sequences with small changes in appearance between subsequent frames. However, it becomes very difficult and sometimes impossible for KLT tracker to deal with large inter-frame motions inducing a substantial optical flow. In order to deal with this problem, a pyramidal implementation of the KLT algorithm was developed.[9] The latter runs the KLT algorithm iteratively on a local template window through successive multi-resolution layers starting from the top of the pyramid (lowest resolution) until recovering the initial image size. This allows larger motions to be caught with less difficulty by breaking the distance through this multi-scale approach. Although it is reasonably efficient, it also implies a certain computational cost which makes its use difficult for

* Corresponding author. E-mail: l.chermak@cranfield.ac.uk

real-time applications, especially in VO context where feature tracking is just a stage among others in the VO pipeline.

In general, the majority of KLT-based variants modify the warping function using an affine model to adapt the template to the different conditions that might occur between two successive images, such as change in illumination, rotation, and scale.[10–12] The work of Hwangbo[13] presents a gyro-aided KLT method. The instantaneous gyro angles are used to get inter-image rotation that serves to compute the homography matrix between two consecutive images.

The obtained homography matrix is used to update the parameters of an affine photometric model for the warping function. The affine photometric model has eight parameters allowing robust tracking to camera rotation and outdoor illumination. However, this model leads to a significant computational cost.

Another work[14] proposes a gyro-aided feature tracking solution for video stabilization. In this contribution, gyroscope measurements from IMU sensor and intrinsic camera parameters are also used to obtain the homography matrix. In contrast to Hwangbo,[13] for computation complexity reasons a translational model is preferred to the affine one for the KLT warping function. While these two precedent works are mono-camera-based approaches,[13,14] Tanathong[15] presents a stereo-camera-based approach. In this work, stereo cameras are mounted on a UAV, and GPS/INS subsequent position information are used to determine the rotation angle of stereo rig between these two positions. Two affine models for warping function using orientation information are proposed here. Assuming a constant distance between the UAV and the ground (i.e., camera facing the ground with no scale changes), the angular information is either used to rotate the entire current images into the same orientation as the previous images; or to rotate the tracking windows into the same orientation as the previous images. Feature tracking is based on pyramidal KLT.

In these three contributions,[13–15] the benefit of gyroscope information is significant allowing the KLT to cope with sharp rotation where it usually fails. However, this remains possible only at the condition of a quasi-pure or a pure camera rotation. Hence, it is assumed a negligible inter-frame translation regarding the scene depth (i.e., very small scale change). High frame rate enables to fulfil this condition, and can be easily achieved due to computational efficiency of the KLT translational model.[14,15] On the other hand, the approach used by Hwangbo[13] requires a parallel processing implementation to achieve high frame rate. Also, by using GPS/INS, the method proposed by Tanathong[15] is still dependent from an external source of information.

Our motivation is to propose an innovative and computationally efficient IMU-assisted KLT tracker, using the full IMU information; robust against rotation changes similarly as[13–15] but also against important scaling between consecutive images. This allows the KLT to be partially released of its spatial constraint allowing low frame rate processing, which is not the case for gyro-only solutions. To enable a continuous and efficient use of accelerometer measurements, the IMU information has to be updated over time. This is why our IMU-assisted KLT tracker technique is an integral part of a VO algorithm, which is the second contribution of this work. Indeed, at each new image the inter-frame pose resulting from our VO initialises the IMU. The overall solution is independent of any external source (e.g., GPS).

This paper is organized as follows: Section 2 describes the overall VO algorithm. Section 3 focuses on IMU-assisted feature tracking and adaptive local tracking window concept. Section 4 gives the detail of double dogleg (DDL) algorithm for motion estimation. Results and discussion are presented in Section 5. Finally, we draw conclusions.

2. System Overview

2.1. Visual odometry and IMU association

Limitations of classical VO have been well identified in the literature.[16] Hence, many fusion works with various sensors have been investigated in order to find the most complementary combination. IMU is particularly cheap and easy to implement with vision system. Its high frequency provides precious motion information filling the interval gap of lower frequency associated vision sensors.[17–19] It is well known that inertial measurements drift with time if there is no update from an external source.[20] Associating it with sensors such as GPS or camera(s), allows the correction of the IMU absolute position, and consequently prevents it to drift over time. IMU information can be used to

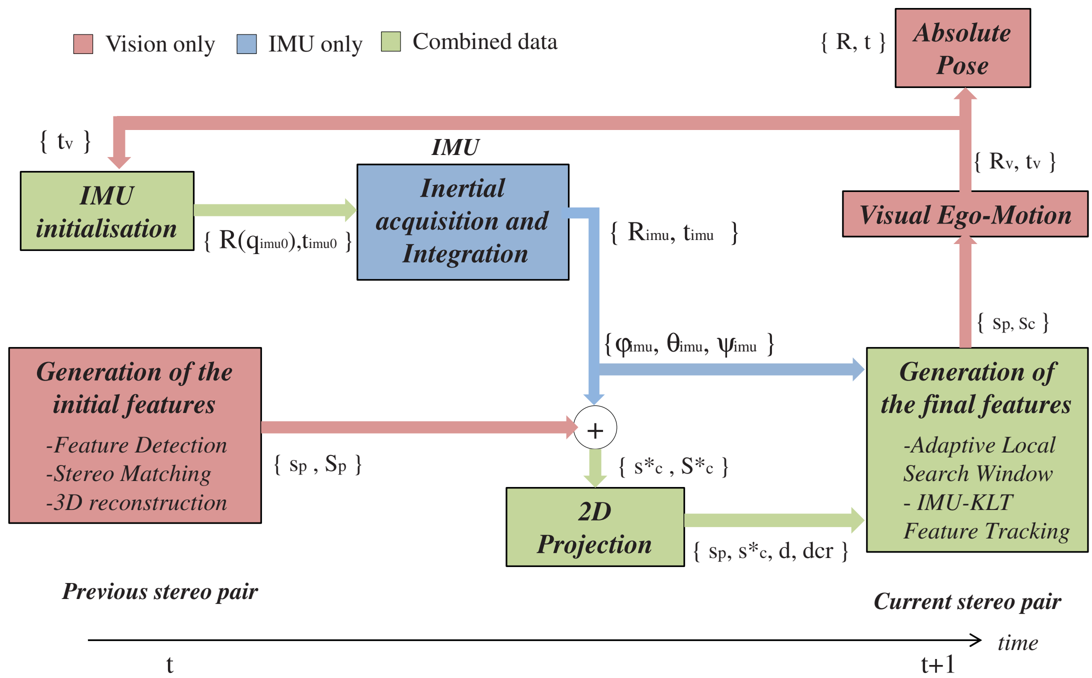

Fig. 1. Illustration of IMU-assisted feature tracking principle in the visual odometry framework.

provide the gravity vector, resulting in a reduction of motion parameters.[21] Those are then combined with Hough Transform to estimate the ego-motion without feature correspondence.

The use of an Extended Kalman Filter (EKF) or one of it variants has been favoured and extensively employed to fuse inertial and vision data, essentially to resolve pose estimation problem.[22–24] Despite showing relatively good efficiency, Kalman Filter approaches based on the pinhole camera model involve costly update of the covariance matrix. This is especially the case for the ones including features in the state vector.

The novel VO approach proposed here is not a pure fusion, but more a complementary association with the IMU. VO benefits from a better quality of features with the IMU-assisted KLT, whilst the IMU is initialised from visual motion estimation. Our approach aims to handle large motion and computational burden by exploiting high-frequency measurements from the IMU.

2.2. Algorithm description

Figure 1 illustrates the whole framework of our VO algorithm, which includes IMU-assisted KLT feature tracking. The different stages composing the algorithm are divided into three categories: Vision only, IMU only, Combined Vision and IMU information. The presented algorithm forms a closed loop designed to maintain the balance between visual and inertial information.

2.2.1. Generation of the initial features. The algorithm starts with feature detection on previous left and right images IpL, IpR at time t. Matches that do not validate epipolar and disparity constraints between I_{pL} and I_{pR} are rejected. The remaining good correspondences form a set s_p of 2D points. These are then reconstructed into 3D by triangulation using stereo calibration parameters forming a set S_p of 3D points.

2.2.2. IMU initialisation. In parallel to the generation of initial features, the IMU is initialised. The inertial rotation matrix R_{imu} is simply initialised with a 3×3 identity matrix as we just need instantaneous gyro-angle. On the other hand, the inertial translation vector t_{imu} is initialised with $v_V = t_V/\Delta_v$ as initial speed (t_V the latest resulting visual inter-image pose translation vector and Δ_v the time elapsed between two stereo frames).

2.2.3. Inertial acquisition and integration. Assuming a uniform acceleration during a short interval of time [*t; t + 1*], acquired accelerations a and angular velocities ω are then, respectively, doubled and simply integrated. Resulting positions and angles are steeply accumulated at each new reading until a new image is acquired. Note that gravity compensation is applied to accelerometer data before being integrated to get the free acceleration. At the end, we obtain the full inertial motion between previous and current stereo pair represented by $R(q_{\text{imu}})$ and t_{imu}.

2.2.4. Initial 3D feature and inertial data combination/2D projection. When the new stereo pair images I_{cL}, I_{cR} are acquired the inertial inter-frame motion ($R(q_{\text{imu}})$ and t_{imu}) between previous and current stereo image pairs is combined with the formed set S_p of 3D points. This results into a new set S_c^* of post inertial motion 3D points that are projected into the current stereo pair images I_{cL}, I_{cR}. This gives a set s_c^* of 2D initial inertial guess.

2.2.5. Generation of the final se of points. An adaptive local tracking window representing inertial guess neighbourhood is calculated for each point of the sets s_p and s_c^*. Individual size of the windows is function of the matches' disparity d and their distance from the centre dcr as well as the measured angles $|\varphi_{\text{imu}}|$ $|\theta_{\text{imu}}|$ $|\psi_{\text{imu}}|$ from the inertial rotation matrix. KLT is then run between related adaptive local tracking windows in previous and current stereo images pairs resulting in a set s_p, s_c of tracked features. Tracked features have to validate the same epipolar and disparity conditions that have been set earlier in the stereo matching stage.

2.2.6. Visual ego-motion. Visual motion estimation is computed from the correct set of tracked features by minimizing the sum of their re-projection errors into the camera coordinate system. We adopt a frame to frame approach in which we are looking for motion parameters forming the inter-image rotation matrix R and the translation vector t that minimize the re-projection error. The solution is generated using DDL trust region method. In an eventual case of failure in the motion estimation process, the pose and the observed velocity is taken from the inertial motion parameters that served earlier in the IMU-KLT feature tracking stage. The absolute pose cumulates the inter-image rotation matrix R_v and the translation vector t_v with the previous ones. The loop is closed by using t_v to get the initial speed v_V for the initialisation of the IMU.

3. IMU-Assisted Feature Tracking

Techniques using orientation information from an external sensor such as IMU orientation information[13,14] or angular information form to GPS/INS positons[15] have been developed in order to cope with fast camera rotations or severe shakes, which usually break KLT conditions. In this work, we developed a technique that also copes with sharp camera rotations and which additionally gives the ability to KLT to deal with severe scale changes.

3.1. IMU features projection via stereo 3D reconstruction

The singularity of our technique resides in the use of stereoscopic properties in order to combine visual and inertial data. In contrary to similar works[13,14] that are based on 2D transform image operation, we use 3D geometry to project initial features with the knowledge of inertial inter-image transform ($R(q_{\text{imu}})$ and t_{imu}) into current stereo images.

Figure 2 summarizes our idea and highlights five keys steps:

(i) *Stereo matching:* The detected features are matched between right and left previous images giving a set $s_p = p_{pL(j)}, p_{pR(j)}$ of n correct stereo correspondences (*j = 1, ..., m, m the number of points*).

(ii) *3D reconstruction:* Features from the set s_p are reconstructed in 3D using stereo calibration parameters by triangulation[25] resulting into a set $S_p = P_{(j)}$ of 3D points representing the position of the stereo correspondences in the space.

(iii) *Inertial motion matrix estimation:* IMU information (accelerometer and gyroscope) are acquired and integrated in order to form the inertial inter-frame relative motion composed of $R(q_{\text{imu}})$ and t_{imu}.

(iv) *Calculation of 3D inertial guesses*: This inertial motion matrix is then combined with S_p in order to obtain the 3D inertial guesses following the equation of motion (1) described below:

$$P_{\text{imu}(j)}\left(t'\right) = R(q_{\text{imu}})_{(k)} P_{(j)}\left(t\right) + t_{\text{imu}(k)} \tag{1}$$

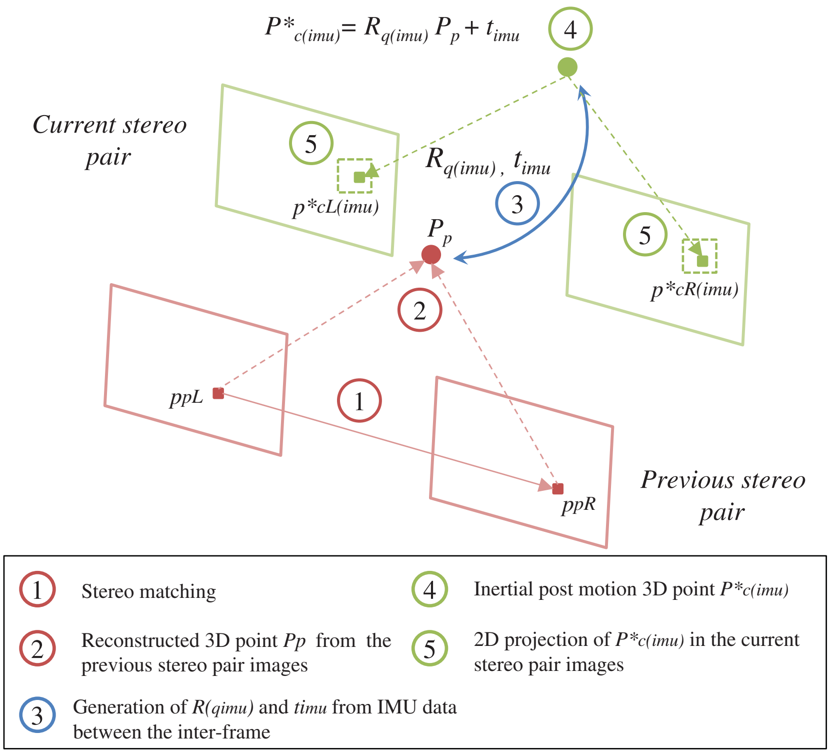

Fig. 2. Illustration of the main steps of IMU-assisted feature tracking principle for one point including: stereo matching, 3D reconstruction, IMU motion transformation, and projection into image plane.

The set of 3D post inertial motion features is called $S_c^* = P_{c(\mathrm{imu})}^*$ with $P_p = [X_p, Y_p, Z_p]^{\mathrm{T}}$ and $P_{c(\mathrm{imu})}^* = [X_{c(\mathrm{imu})}, Y_{c(imu)}, Z_{ci(\mathrm{imu})}]^{\mathrm{T}}$.

(v) *Projection into 2D image plane:* The components of S_c^* are projected into the current stereo pair images using the stereo camera parameters as described below:

$$\begin{cases} p_{cL(\mathrm{imu})}^* = \begin{bmatrix} u_{cL(\mathrm{imu})}^* \\ v_{cL(\mathrm{imu})}^* \end{bmatrix} = \begin{bmatrix} f\frac{X_{c(\mathrm{imu})}}{Z_{c(\mathrm{imu})}} + u_0 \\ f\frac{Y_{c(\mathrm{imu})}}{Z_{c(\mathrm{imu})}} + v_0 \end{bmatrix} \\ p_{cR(\mathrm{imu})}^* = \begin{bmatrix} u_{cR(\mathrm{imu})}^* \\ v_{cR(\mathrm{imu})}^* \end{bmatrix} = \begin{bmatrix} f\frac{(X_{c(\mathrm{imu})}-B)}{Z_{c(\mathrm{imu})}} + u_0 \\ f\frac{Y_{C(\mathrm{imu})}}{Z_{c(\mathrm{imu})}} + v_0 \end{bmatrix} \end{cases} \tag{2}$$

where f is the focal length, u_0 and v_0 are the central pixel's coordinates, and B the stereo baseline.
A set $s_c^* = p_{cL(\mathrm{imu})}^*, p_{cR(imu)}^*$ of 2D post inertial motion features is then formed in which a high-confidence tracking area can be built.

3.2. Adaptive local tracking windows

Most of the time, inertial guesses are giving to the KLT a relatively fair estimation of the features to be tracked. That said, and in order to take the full advantage of it, we aim to maximise the use of the KLT by restraining the tracking area to the strict neighbourhood of the inertial guesses. Indeed, restraining the tracking area has two major advantages. First, it reduces the probability of tracking wrong features. Secondly, it increases KLT computational efficiency because of the great reduction in size of the tracking search area. We call these newly defined tracking areas: Adaptive Local Tracking Windows (ALTW) illustrated in Fig. 3.

(1) Let us call $s_{pc}^* = p_{pL}, p_{pR}; p_{cL(\mathrm{imu})}^*, p_{cR(}^*$imu) the set of features regrouping the initial stereo points and their related inertial projections.

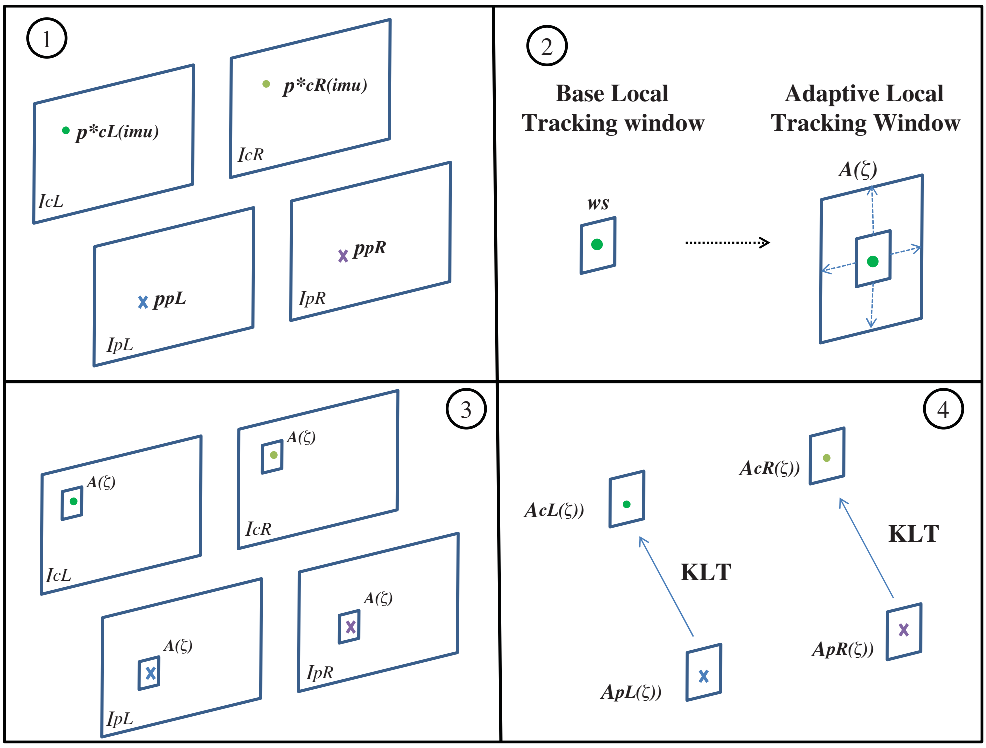

Fig. 3. Illustration of the ALTW concept in four main steps: inertial projection, size calculation of adaptive local tracking windows, sub-set images extraction, and feature tracking with KLT.

(2) Sizes of the adaptive local tracking windows are calculated according to global and local parameters forming the vector ζ:

$$\zeta = \left[\, |\varphi_{\text{imu}}| \; |\theta_{imu}| \; |\psi_{\text{imu}}| \; d \; dcr \,\right]^T \tag{3}$$

Global parameters $[|\varphi_{\text{imu}}| \; |\theta_{\text{imu}}| \; |\psi_{\text{imu}}|]^{\text{T}}$ are the Euler angles resulting from the inertial inter-image information from $R(q_{\text{imu}})$. These are influencing all the features in a same manner.

Local parameters $[d, dcr]^{\text{T}}$ represent, respectively, the disparity and the distance of the feature from the image centre. These are specific to each feature relatively to its position in the image. According to the parameters ζ, adaptive local tracking window size $A(\zeta)$ is calculated for each component of s^*_{pc} following the conditions below:

- If the velocity *vehicle* > 3 m/s and ($|\varphi_{imu}|$ or $|\theta_{imu}|$ or $|\psi_{imu}|$) > 0.009 rads then

$$A(\zeta) = ws + g(|\varphi|, |\theta|, |\psi|) + l(d, \; dcr) \tag{4}$$

- Otherwise

$$A(\zeta) = ws + l(d, \; dcr) \tag{5}$$

where *ws* is a constant 9 × 9 base size of the local tracking window. This initial size can be widen gradually according to the score obtained with the local and global sub-functions, respectively *l(d, dcr)* and $g(|\varphi|, |\theta|, |\psi|)$. For instance, typical values 0.5°,1°, 2°, 3°, 4° correspond to 7, 9, 16, 27, 41 pixels. The sub-functions g and l are experimentally expressed as follow:

$$g(|\varphi|, |\theta|, |\psi|) = \frac{(1 + 10 \times \max(|\varphi|, |\theta|, |\psi|))^4}{0.2} \tag{6}$$

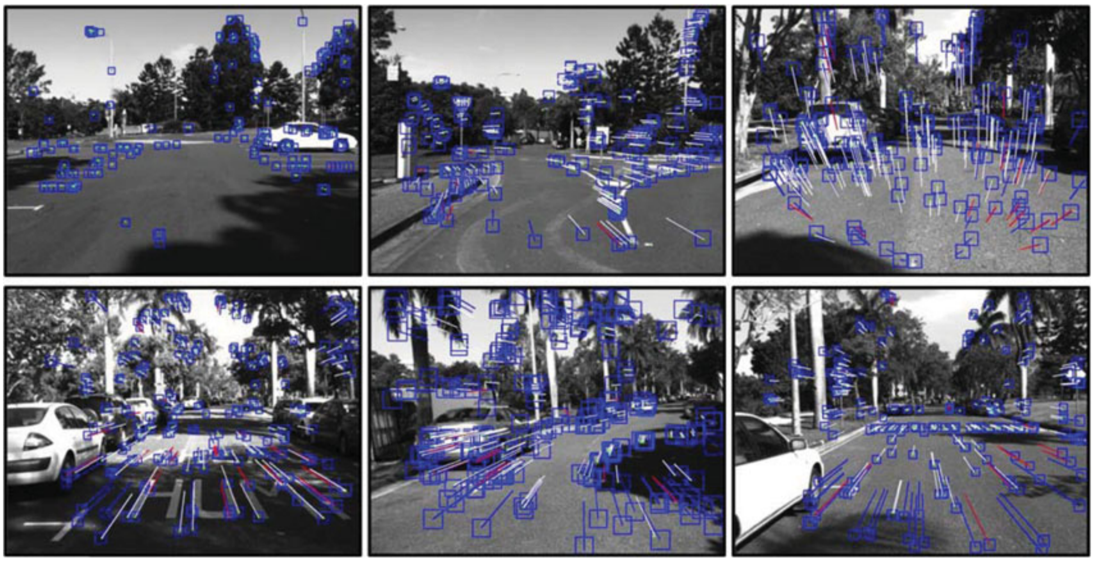

Fig. 4. Illustration of adaptive local tracking windows (blue squares). Blue lines—inertial optical flow, red lines—outliers, white lines—inliers.

and

$$l\,(d,\ dcr) = 4 \times \frac{d}{f \times B} + 2 \times \left(\frac{dcr - 400}{100}\right) \qquad (7)$$

where f the focal length, and B the stereo baseline. Notably, a maximum size of 40 square pixels is fixed even if the score of $A(\zeta)$ is found to be higher.

(3) Depending on the local parameters, a maximum of four different ALTW sizes can be obtained. However, we only keep the adaptive local tracking windows with the larger size which is applied to all. Thus, each individual tracking set s^{*}_{pc} has a unique ALTW size $A(\zeta)$ surrounding each of its components.

(4) For each component of s^{*}_{pc}, we extract a region of interest, which has the size of the calculated ALTW resulting in a sub-set of four images $A_{\mathrm{ALTW}} = A_{pL}(\zeta), A_{pR}(\zeta); A_{cL}(\zeta), A_{cR}(\zeta)$. Figure 4 illustrates use of ALTW on different images.

Before running KLT, a phase correlation stage is done between $A_{pL}(\zeta)$ and $A_{pR}(\zeta)$ and, respectively, between $A_{cL}(\zeta)$ and $A_{cR}(\zeta)$.

Phase correlation stage is a fast frequency-based approach giving an estimate of an eventual translational offset between two similar images. If an offset is found, it is then use to refine the position of inertial guesses. Discrete Fourier Transform (DFT) is calculated for the two related ALTW and serves then to compute the cross-power spectrum C as such

$$C = \frac{G_a G_b^{-}}{\left|G_a G_b^{-}\right|} \qquad (8)$$

where

$$\begin{cases} G_a = F\left\{A_{pL}\,(\zeta)\right\} \text{ and } G_b = F\left\{A_{cL}\,(\zeta)\right\} \text{ for left pair} \\ G_a = F\left\{A_{pR}\,(\zeta)\right\} \text{ and } G_a = F\left\{A_{cR}\,(\zeta)\right\} \text{ for right pair} \end{cases}$$

where F is the forward DFT and the exponent “ $^{-}$ ” indicates the conjugate of the DFT and F^{-1} is the inverse DFT.

Then, the cross correlation (8) is converted back to the time domain and the translational shift υ is deduced from the peak location:

$$\left(\upsilon_x, \upsilon_y\right) = \underset{(x,y)}{\text{argmax}} \left(F^{-1}\{C\}\right) \tag{9}$$

3.3. KLT formulation

The KLT tracker is defined as a non-linear optimisation problem that aims to minimise the squared sum of the intensity difference e between two successive images I_p and I_c. This operation is done over a small patch of size $(2w_x + 1) \times (2w_y + 1)$ centred, respectively, at the position $x = [u, v]^T$ and the tracking motion model *w(*x*;p)*.

$$e = \sum_{-w_x}^{w_x} \sum_{w_y}^{w_y} \left[I_p\,(\mathrm{x}) - I_p(w\,(\mathrm{x};\mathrm{p})\right]^2 \tag{10}$$

where w_x and w_y are two integers usually set to 7,8,10,... 21 according to Eq.(9). The local patches inner pixels are used to create an over constrained system.

The tracking motion model, also called *warping function w(x;p)*, is composed of $\mathrm{x} = [u, v]^T$ a pixel coordinate and p a vector of warping parameters.

The KLT starts with an initial value of p, and iteratively searches for the δp that aligns the two image patches such that Eq.(10) is minimised. The warping functions can be written following two different models. First, the translational model expressed as

$$w\,(\mathrm{x}; p) = w\,(\mathrm{x} + b) \tag{11}$$

The second one is the affine model expressed as:

$$w\,(\mathrm{x}; p) = w\,(\mathrm{Ax} + b) \tag{12}$$

where

$$A = \begin{bmatrix} 1 + \eta_{xx} & \eta_{yx} \\ \eta_{xy} & 1 + \eta_{yy} \end{bmatrix} \text{and } b = \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix}$$

The translational model has the advantage to be fast. However, it is only reliable if the appearance change between subsequent images remains small. p needs to be initialised with a pixel position close enough to its target in order to allow translational-based KLT to cope with large optical flows.

On the other hand, the affine model (or also called *pyramidal* KLT) gives more options to deal with spatial deformation of the patches. However, it is computationally more expensive than the translational model. For this implementation, we opted for the translational model (11). In our case, the warping parameter p takes as initial points, an inertial guess:

$$p = b = \begin{cases} \begin{bmatrix} u^*_{CL(\mathrm{imu})} \\ v^*_{CL(\mathrm{imu})} \end{bmatrix} \text{for left pair} \\ \begin{bmatrix} u^*_{CR(\mathrm{imu})} \\ v^*_{CR(\mathrm{imu})} \end{bmatrix} \text{for right pair} \end{cases} \tag{13}$$

If the quality of the IMU measurements are good, inertial guesses have a high probability to be close enough to their related targets. These provide then an ideal initial condition to minimize (10).

The latter is re-written with the knowledge of the ALTW (Eqs. (4) and (5), and Fig. 3) as

$$e \begin{cases} \sum_{-w_x}^{w_x} \sum_{-w_y}^{w_y} \left[A_{\boldsymbol{pL}(\zeta)}(\mathrm{x}) - A_{\boldsymbol{cL}(\zeta)}(w(\mathrm{x}+\mathrm{b}))\right]^2 \\ \sum_{-w_x}^{w_x} \sum_{-w_y}^{w_y} \left[A_{\boldsymbol{pR}(\zeta)}(\mathrm{x}) - A_{\boldsymbol{cR}(\zeta)}(w(\mathrm{x}+\mathrm{b}))\right]^2 \end{cases} \tag{14}$$

Instead of searching on the entire image for each correspondence, the KLT only focus on limited areas defined by adaptive local tracking windows calculated earlier (Fig. 3). This presents many advantages:

- First, the inertial projection features hold the full inter-image information (rotation plus translation), which enables the KLT to remain robust to scale changes.
- Secondly, the sub-images only extract the close neighbourhood of the points composing a set s^*_{pc}. Consequently, it prevents the KLT from false tracking.
- Thirdly, the computational time of the KLT drops considerably because of the small size of the tracking areas defined by the calculated ALTW.

All these advantages are made possible by the combination of full inertial information combined with 3D geometry and stereoscopy allowing a precise prediction of the tracked features location (see Fig. 2). This is the main difference with related work,[13,14] where inertial features are calculated from the rotation matrix R_{gyro} only using 2D image homography transformation as follows:

$$\begin{cases} b = \mathrm{x}_{\mathrm{gyro}} = H_{\mathrm{x}} \\ \quad \text{with} \\ H = K R_{\mathrm{gyro}} K^{-1} \end{cases} \tag{15}$$

where H is the 3×3 homography matrix and K is the camera calibration matrix. Features are rotated in the image plane according to the gyroscope information and serve as initial conditions to minimize Eq.(10). The warping function of Ryu[14] follows the translational model and consequently it only requires b (11) as input for the warping parameter p. On the other hand, Hwangbo[13] uses a more advanced affine photometric model with eight parameters $p = (\eta_{xx}, \eta_{xy}, \eta_{yx}, \eta_{yy}, \eta_x, \eta_y, \alpha, \beta)$, where α and β deal with illumination change. For these two techniques[13,14] based only on rotation information scale change remains an issue.

3.4. Inertial dynamic features rejection

The use of full IMU information combined with 3D geometry and stereoscopy brings to each feature a coherent motion behaviour. Additionally, the rigid vehicle-camera-IMU structure jointly moves towards the scene in a unique manner. This gives to the inertial guesses similar projection trend whether if the initial detected were belonging to static or dynamic objects.

As the principle our methods is to limit tracking to the strict neighbourhood of the inertial guesses, the calculated ATWs have a high probability to not contain anymore the dynamic feature. This, results from the fact that dynamic object motion will certainly differ from the rigid vehicle-camera-IMU structure. Consequently, KLT fails to find a similar pattern which automatically eliminates a dynamic feature. Figure 5 illustrates a case where dynamic features belonging to a moving track are discarded.

In Fig. 5 (bottom left), red lines represent the optical flow of tracked features with Affine KLT between the top images. On the other hand, in Fig. 5 (bottom middle) blue lines represent the optical flow between the detected features on the before motion and the inertial guesses projected in the after motion image. As the truck motion differ from the vehicle motion, inertial guesses do not point to the area where the truck moved. Consequently, tracking of features belonging to the truck will fail.

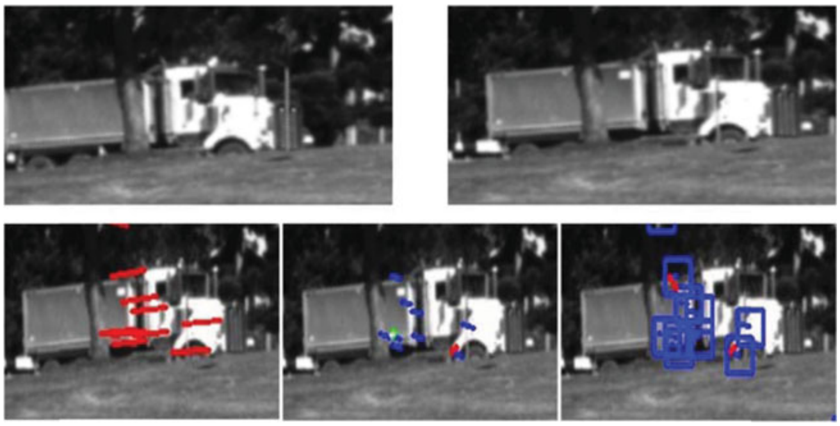

Fig. 5. Illustration of the effect of inertial guesses on features belonging to dynamic objects: Top: Consecutive images illustrating a moving truck before (left) and after motion (right); Bottom: Optical flow plotted on the after motion image between detected and: left—tracked feature using Affine KLT; middle—inertial guesses; right—inertial guesses with their respective ALTWs displayed.

4. Motion Estimation

4.1. Local bundle adjustment and trust region method

In this section, the presented solution for motion estimation is a minimalist version of the local bundle adjustment involving only two consecutive stereo pair images. It only focuses on the optimisation of the relative motion parameters between these two views.[26] Several contributions[1,16,18] demonstrated that the use of multiple frames bundle adjustment approaches increase the accuracy of VO final position error over long-distance navigation. However, it was also shown that the use of a two frames approach bundle adjustment with a robust features association scheme [4,26] achieve a position error of <1% of the travelled distance while being computationally efficient. This two views scheme fits perfectly our feature tracking strategy, where features over are not tracked over more than two consecutive frames in order to avoid drift in image feature localization.

The non-linear objective function f to minimise is the image re-projection error function of the motion parameter vector κ expressed as follows:

$$\min \sum_{i=1}^{N} \left\| p_{cL(i)} - f\left(P_{p(i)};\kappa\right)\right\|^2 + \left\| p_{cL(i)} - f\left(P_{p(i)} - B;\kappa\right)\right\|^2 \tag{16}$$

where

$$\kappa = \begin{bmatrix} q_0 & q_1 & q_3 & q_4 & t_x & t_y & t_z \end{bmatrix}^T \tag{17}$$

This motion parameter vector κ to be optimised is a 1×7 vector which consists of the four quaternion elements for the orientation and the translational elements on the three axis (x, y, z).

Non-linear re-projection function f takes as input, P_p a 3D triangulated feature from the previous stereo pair and the motion parameter κ (also the baseline B for right pair features). The link between spatial and planar representations is obtained with the help of the rectified camera matrix K_{rect} in the way as Eq.(2). The objective is to reduce the pixel distance between the tracked features and their relative re-projected features.

Levenberg–Marquardt (LM) algorithm is widely used to solve bundle adjustment problem. [27–28] This technique that was first proposed by Levenberg[29] and then Marquardt,[30] is commonly adopted to solve non-linear least square problems because of its easy implementation and its effectiveness. It also belongs to the trust region methods family that guaranties local convergence while avoiding the non-positive definite Hessian problem in contrary to Newton's method.

In contrary to line search optimisation methods, trust region approaches set first a maximum distance before choosing a direction. Hence, the model is trusted around a restricted area Δ, which is

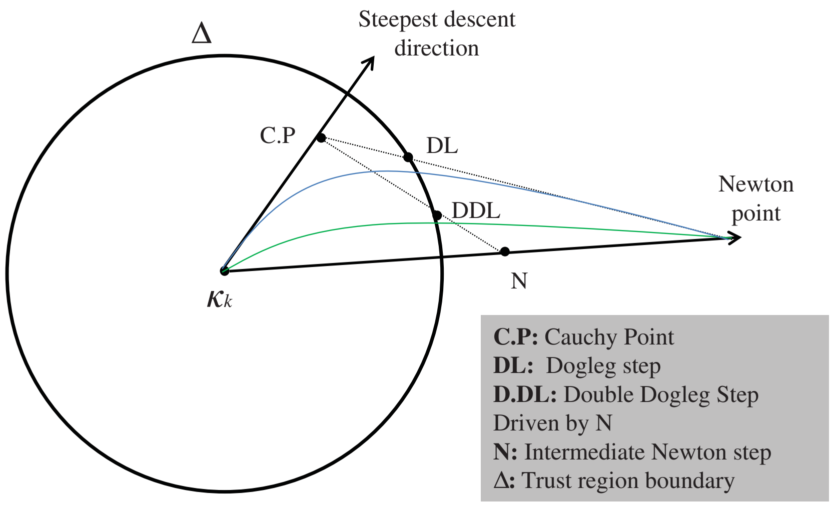

Fig. 6. Illustration of dogleg and double dogleg convergence curve.

adjusted along iterations. If the model matches the objective function f then Δ is increased, whereas it decreases if the approximation is poor.

4.2. *Double dogleg*

In this work, instead of using LM algorithm, we decided to adopt DDL trust region method,[31] which is a variant of the dogleg (DL) algorithm[32] to solve the bundle adjustment for motion estimation. It has already been shown that the use of DL trust region technique presents advantages in term of computational cost compared to Levenberg–Marquardt methods for full bundle adjustment applied to 3D structure reconstruction only.[33] DL algorithm is delineated by two lines composed of the steepest descent direction and the Newton point direction (see Fig. 6). The optimal trajectory follows the steepest descent direction until reaching the Cauchy point (CP) then converges to the Newton point passing by the DL step.

This later should be intersecting with the trust region boundary Δ. By introducing an intermediate Newton step N between the CP and the actual Newton point, the behaviour of the DDL algorithm presents a further improvement. Indeed, the optimal curve trajectory crosses the trust region before original DL. This direct control between these two lines (Steepest descent and Newton) by the mean of trust region (characterised by Δ) gives a faster optimisation to the algorithm and is also the main difference with LM algorithm.[34]

Trust region sub-problem follows a quadratic model $\varrho(\delta)$ that approximates the objective function $f\ (\kappa)$ (16) in such a way that the model is trusted within a limited region Δ_k around the current motion parameter vector κ:

$$Q_k(\delta) = f(\kappa) + \left(J^T r\right)^T \delta + \frac{1}{2}\delta^T \left(J^T J\right)^T \delta \tag{18}$$

where

$$r = x - f(\kappa) \text{ and } J = \frac{\partial f(\kappa)}{\partial \kappa}$$

Where r is the $n\times 1$ residual vector, J is the $n\times 7$ Jacobian matrix of the function $f(\kappa)$ and $J^T J$ is the $n \times n$ approximation of the Hessian matrix, and I the $n \times n$ identity matrix (n the number of correspondences).

Solving (18) is not straightforward. In the case where the unconstrained solution δ_N (Gauss–Newton step) is too long, the convergence trajectories are ruled out by the following equations:

$$\begin{cases} \delta_{DL} = \delta_{CP} + \lambda\,(\delta_{CP} -\ \delta_N)\ \text{for}DL \\ \delta_{DDL} = \delta_{CP} + \lambda\,(\beta\delta_{CP} -\ \delta_N)\,\text{for}DDL \end{cases} \tag{19}$$

with $\beta = 0.8\gamma + 0.2$ ($\gamma \epsilon$ [0,1]) is an adjusting parameter that fixes the position of the intermediate Newton step N in the Newton direction for the DDL, (see Fig. 6). The minimiser for the Cauchy point δ_{CP} and the Gauss–Newton step δ_N are obtained using these equations.

$$\begin{cases} \delta_{CP} = -\frac{\|b^T b\|}{\|b^T Ab\|} b \\ \delta_N = A^{-1} b \end{cases} \tag{20}$$

where approximated Hessian $A = J^T J$ and the gradient $b = J^T(x - f(\kappa))$ with the Jacobian $J = [\partial q/\partial\kappa, \partial t/\partial\kappa]^{\mathrm{T}}$ and $f(\kappa)$ is the objective function. Depending on the method chosen (19), λ must achieve

$$\delta_{DL} = \Delta \ \text{ or } \ \delta_{DDL} = \Delta \tag{21}$$

Thus, in our case the chosen method is DDL, and the current point κ_k is updated according to the constrained trust region as

$$\kappa_{k+1} = \begin{cases} \kappa_k - \frac{\Delta_k}{\delta_{CP}} \text{ if } \delta_{CP} \geq \Delta_k \\ \kappa_k + \delta_{DDL} \text{ if } \delta_{CP} \,\langle \Delta_k \text{ and } \delta_N \rangle\, \Delta_k \\ \kappa_k + \delta_N \text{ if } \delta_{CP} < \Delta_k \text{ and } \delta_N \leq \Delta_k \end{cases} \tag{22}$$

After the calculation of the new point κ_{k+1}, the trust region Δ_{k+1} is then updated according to the reduction ratio ρ_k between the actual residual r_{act} and the predicted residual r_{pred} defined below:

$$\rho_k = \frac{r_{\text{act}}}{r_{\text{pred}}} = \frac{f\,(\kappa_k) - f\,(\kappa_{k+1})}{Q_k\,(0) - Q_k\,(\delta_{DDL})} \tag{23}$$

$$\Delta_{k+1} = \begin{cases} \gamma_1 \Delta_k \text{ if } \rho_k < \eta_1 \\ \Delta_k \text{ if } \eta_1 \leq \rho_k < \eta_2 \\ \gamma_2 \Delta_k \text{ if } \rho_k \geq \eta_2 \end{cases} \tag{24}$$

where $0 < \gamma_1 < 1 < \gamma_2$ and $0 < \eta_1 \leq \eta_2 < 1$. The value of Δ_k is increased or decreased according to the quality of the model approximation of the objective function f (16). The algorithm converges when $\|b\| \leq \epsilon$.

Algorithm 1 RANSAC-based motion estimation routine

Input: s
Output: $\kappa_{\text{final}}, s_{\text{inliers}}$
Set: $\tau = 5$;
 # start of RANSAC routine for motion estimation
for $k = 0$ to 50
  $N_k = 0$
  form the set s by getting 3 random pairs from s
  initialise $\kappa_k = \kappa_0$
  # call the chosen optimisation process
  κ_{optim} = DoubleDogleg(s_{RNG}, κ_k,) (Algorithm 2)
   if(converged)
   # get the inliers correspondences
    for each pair from s do
    if($\| p_{cL} - p^*_{cL}\| + \| p_{cR} - p^*_{cR} \| < \tau$)
     $N_k = N_k + 1$
    endif

endfor

if($N_k > N_{\text{best}}$).
#create an feature the set $s_{\text{inliers}} fromtheN_k$
inliers
$N_{\text{best}} = N_k$
$\kappa_{\text{best}} = \kappa_{\text{optim}}$
endif
endif
endfor

 # refinement stage using $\kappa_{\text{best}} s_{\text{inliers}}$
$\kappa_{\text{final}} = \text{DoubleDogleg}(s_{\text{inliers}}, \kappa_{\text{best}})$ (Algorithm 2)

Algorithm 2 Double Dogleg minimisation process

Input: p_{cL}, p_{cR}, P_p, κ_0
Output: κ_{optim}
Set: $\Delta_k = 1$, $\beta = 0.6667$, $\eta_1 = 0.15$, $\eta_2 = 0.75$, $\gamma_1 = 0.5$, $\gamma_2 = 2$, $\rho_0 = 1$, $\chi = 1\text{e}^{-4}$, $k = 0$, maxiter =50;
 # start of optimisation task
while(not converged and not *maxiter*)
if($\rho_k > 0$)
Compute:
$A = J^T J$, $b = J^T r$, $\|b\|$, $\|\kappa_k\|$, $\|\|$x - $f(\kappa_k, P_p)\|$
and $\|\delta_{CP}\|$ (20)
G.N (Gauss–Newton) = false
if($\|\text{b}\|_\infty < \chi$ or $\|r\| < \chi$)
$\kappa_{\text{optim}} = \kappa_k$
converged
endif
endif

if($\|\delta_{CP}\| > \Delta_k$)
$\delta_{\text{DDL}} = -(\Delta_k/\|\delta_{CP}\|)^* \Delta_k$
else if(not G.N)
Compute δ_N (20)
endif

if($\|\delta_N\| \leq \Delta_k$)
 # take Gauss–Newton direction
$\delta_{\text{DDL}} = \delta_N$
else
Compute δ_{DDL} (19)
endif

if($\|\delta_{\text{DDL}}\| < \chi \; \|\kappa_k\|$)
$\kappa_{\text{optim}} = \kappa_k$
converged
else
Compute:
$\kappa_{\text{new}} = \kappa_{k+}\delta_{\text{DDL}}$
$A = J^T J$, $b = J(x - f(\kappa_{\text{new}}, P_p))$ *and* ρ_k (23)

if($\rho_k > 0$)
$\kappa_k = \kappa_{\text{new}}$ (22)
$k = k+1$

endif

update trust region boundary Δ (24)
endif
endwhile

4.3. Visual motion estimation algorithm

Let us consider $s = p_{pL(i)}, p_{pR(i)}; p_{cL(i)}, p_{cR(i)}$ the set holding all the feature correspondences linking consecutives stereo image pairs (*i = 1, ..., n, n is the number of correspondences*). The presented motion estimation algorithm is implemented following a RANSAC-based scheme as a part of an efficient inliers selection strategy. At each step, it starts with the selection from the set s of three random pairs of initial features and their corresponding tracked features. These three random pairs form the set $s_{\text{RNG}} = p_{pL(j)}, p_{pR(j)}; p_{cL(j)}, p_{cR(j)}(j = 1, 2, 3)$. The motion parameters of κ is initialised to $\kappa_0 = [1\ 0\ 0\ 0\ 0\ 0\ 0]^{\text{T}}$ as it is a sufficient condition for the motion parameters to converge, especially when the quality of the feature correspondences are good.

Then, the minimisation method (Algorithm 2) is called taking as input the motion parameter κ_k and s_{RNG} from which $P_{p(j)}$ is the 3D position of the previous features. In the case where the optimisation methods led to a converging solution, the resulting estimation of κ_k for the current set s_{RNG} of random points is assessed for the whole set s. Each feature correspondence from which the re-projection error falls under a fixed threshold τ is considered as inlier. At the end of the RANSAC routine, κ_{best} takes the value of the motion parameter estimation κ_k giving the largest amount of inliers. This enables us to form a new set of filtered feature correspondences $s_{\text{inliers.}}$

A refinement step runs the minimisation method (Algorithm 2) for a last time using only the set of inliers s_{inliers} obtained from κ_{best}. This gives the final solution κ_{final} from which the inter-frame rotation $R_{(q)}$ and translation t are computed. Relative motion estimations are then accumulated along the platform's (vehicle's) route in order to reconstitute the whole travelled trajectory. Algorithm 1 describes the RANSAC-based motion estimation routine. This RANSAC-based motion estimation routine optimises the motion parameter κ and at the same time discards all possible outliers that were contained in the set s.

Processing iteratively a small set of three random pairs gives to this routine a great efficiency. Furthermore, it showed an impressive robustness against outliers especially the ones belonging to dynamic objects such as vehicles, pedestrians, trams, etc. Knowing how much the outliers can be misleading, and especially how a wrong estimation of the motion can dramatically affect the whole generated trajectory, it is very valuable to have reliable motion estimation routine.

5. Experiments

In this section, we present the results of our VO algorithm running on an urban environment dataset[35] available at https://wiki.qut.edu.au/display/cyphy/UQ+St+Lucia. In this dataset,[35] a car is equipped with stereo cameras (Point Grey® Flea2) and an IMU-GPS device (XSens® Mti-g) mounted on its roof. Visual data consist of 1024 × 768 stereo images acquired at 30 Hz. IMU data provide calibrated accelerometer and gyroscope information at 100 Hz, and GPS data at 1 Hz. INS data are also provided in this dataset. It will serve as a reference in the VO trajectory comparison tests. In the first instance, feature tracking performance only will be evaluated. Then, we will assess the impact of feature tracking on the quality of the VO generated trajectories. Finally, it is the accuracy of the VO generated trajectories focusing this time only on motion estimation techniques which will be analysed.

5.1. Feature tracking and processing time performances

In this paper, one of our aims is to assess the performances of our IMU-assisted KLT tracker in the presence of large optical flows and to show its robustness against severe change in scale. Consequently, we decided to re-sample the 30 Hz original image sequence respectively to 10 Hz, 5 Hz, and 3 Hz sequences in order to emphasise the scaling effect on consecutive images. The lower the frequency, the higher inter-frame gap between consecutive images which makes tracking of the features much more challenging.

Table I. List of KLT techniques with their characteristics.

Techniques	Abbreviation	Model	Pyramid level	Patch size
KLT (visual only)	T	Translational	0	21 × 21
gyro-assisted KLT without adaptive local tracking windows (gyro)	GT	Translational	0	21 × 21
IMU-assisted KLT without adaptive local tracking windows (gyro + accelerometer)	IT	Translational	0	21 × 21
IMU-assisted KLT (gyro + accelerometer)	ITA	Translational	0	$A(\alpha)/2 \times A(\alpha)/2$
gyro-aided KLT (gyro)[13]	GA	Affine	3	21 × 21

Table II. KLT techniques: tracking performances, processing time, and time ratio (ITA taken as reference) at different frequencies on dataset[35] full sequence.

		ITA	T	GT	IT	GA
Tracking performances (%)	10 Hz	**92.5**	54.5	56.5	88.3	67.5
	5 Hz	**85.4**	32.2	34.2	75.3	59.2
	3 Hz	**77.3**	18.7	19	62	47.8
Processing time (ms)	10 Hz	**18**	38	40	31	141
	5 Hz	**17**	34	38	31	138
	3 Hz	**17**	33	36	29	128
Time ratio	10 Hz	1	2.2	2.2	1.7	7.8
	5 Hz	1	2	2.2	1.8	8.1
	3 Hz	1	1.9	2.1	1.7	7.5

In order to highlight the importance of using full IMU information, three variations of our IMU-assisted KLT method have been tested. Table I describes abbreviations and characteristics of the compared KLT-based techniques.

- The first variation uses accelerometer and gyroscope information and adaptive local tracking windows.
- The second variation is the same as the first one but without adaptive local tracking windows.
- The third variation is the same as the second one but using gyroscope information only.

Results of our IMU-assisted KLT feature tracking in its three variation forms are compared to conventional KLT but also to the Hwangbo's gyro-aided KLT method.[13] In order to have a complete and fair comparison, we adapted their code (https://www.cs.cmu.edu/ myung/IMU_KLT/ index.html#source) to our stereo motion scenario, as it is originally designed for monocular camera systems. The evaluation criteria are the rate of inliers over tracked features and KLT processing time.

Table II presents feature tracking and time-related performances at different sequence frequencies for the techniques mentioned in Table I.

For the full sequence, a general drop of the percentage of correct tracked features can be observed when the gap between consecutive frames gets bigger (i.e., lower frequency). Besides, a decrease of the processing time decrease can be noticed. This is a logical consequence of the lower number of initial features to track at lower frequency. T loses almost 50% of features on average at 10 Hz and can only save around 20% at 3 Hz. These results are not surprising at all according to the nature of KLT, which hardly copes with scaling. GT which adds gyroscope information slightly improves the performance but remains very similar to T. When using the full IMU information (IT), the tracking performance is greatly improved compared to T and computation time gets also reduced because of the higher precision of the inertial features. Because of the affine approach GA gives better results than GT. This is even truer at low frequency. However, it remains clearly below IT. Additionally, homography-based affine model requires a much higher computation cost. Finally, ITA offers the best tracking performance with the lowest processing time.

This demonstrates the remarkable advantage of the adaptive local tracking window concept, which enables ITA to be twice faster than T and to keep a relatively high rate of correct tracked features while remaining robust to scale change. This last point is clearly illustrated in Fig. 7, by analysing the

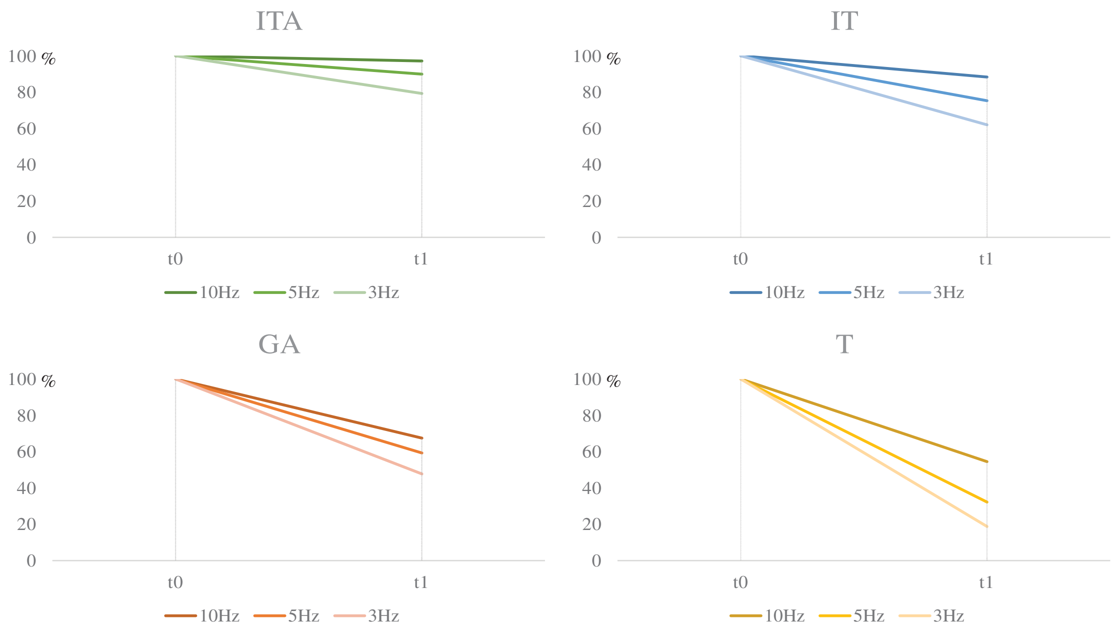

Fig. 7. Illustration of the full sequence average tracking performance drop for the different techniques as a function of the scale change (GT not displayed here as it is very similar to T).

slope of the different techniques regarding tracking performances against increasing scale change. Indeed, with 77.3% of correctly tracked features even at 3 Hz, ITA is the most robust against scale change compared to his other technique.

Table III presents results of the tracking performance for the different KLT-based techniques in three challenging areas. These happen within the full sequence where the vehicle undergoes obstacles resulting in severe rotations (pitch and roll). In these three cases the results achieved reinforce the findings in Table II.

Performance of T drops completely even at 10 Hz. On the other hand, we note that gyro-based solutions (GT and GA) are giving better performances here than on the full sequence especially at higher frequencies (10 Hz and 5 Hz). This is not the case at 3 Hz where the scale change is too severe between subsequent images. GA gives relatively close results to IT, although it is less valid at 3 Hz. ITA remains far better than the other techniques, coping impressively well with the huge scaling and rotational changes. Comparing performances between ITA and IT confirms the contribution of adaptive local tracking window in the improvement of the tracking performance.

Figures 8, 9, and 10 illustrate two images on the three cited challenging points for each KLT-based technique at 5 Hz. Most of the tracked features from GA belong to far objects. These are also the one that are less affected with scaling. Hence, they are easily mapped by homography. However, in Figs. 9 and 10, GA is severely affected by the large scaling and cannot track any features.

Conversely, ITA and to a less extent IT are able to track features close to the camera presenting large optical flows that can reach more than 100 pixels in translation. These features are really important for motion estimation as they present a significant disparity, which means that they are less subject to errors.

5.2. Impact of feature tracking in motion estimation

We evaluated the impact of the KLT-based techniques on the quality of motion estimation on the same dataset using as optimization method DDL for all of them. The vehicle is driven over 491 m curved trajectory in an urban environment subject to strong contrasts. Figure 11 illustrates the trajectories generated with all KLT-based techniques. T, GT, and GA, final positions are far from the actual final position F (definition of this final position is given in Section 5.3). On the other hand, IT and ITA trajectories follow the road path and their respective final position is very close to F. This confirms the correlation between the number and the quality of the features and the motion estimation accuracy.

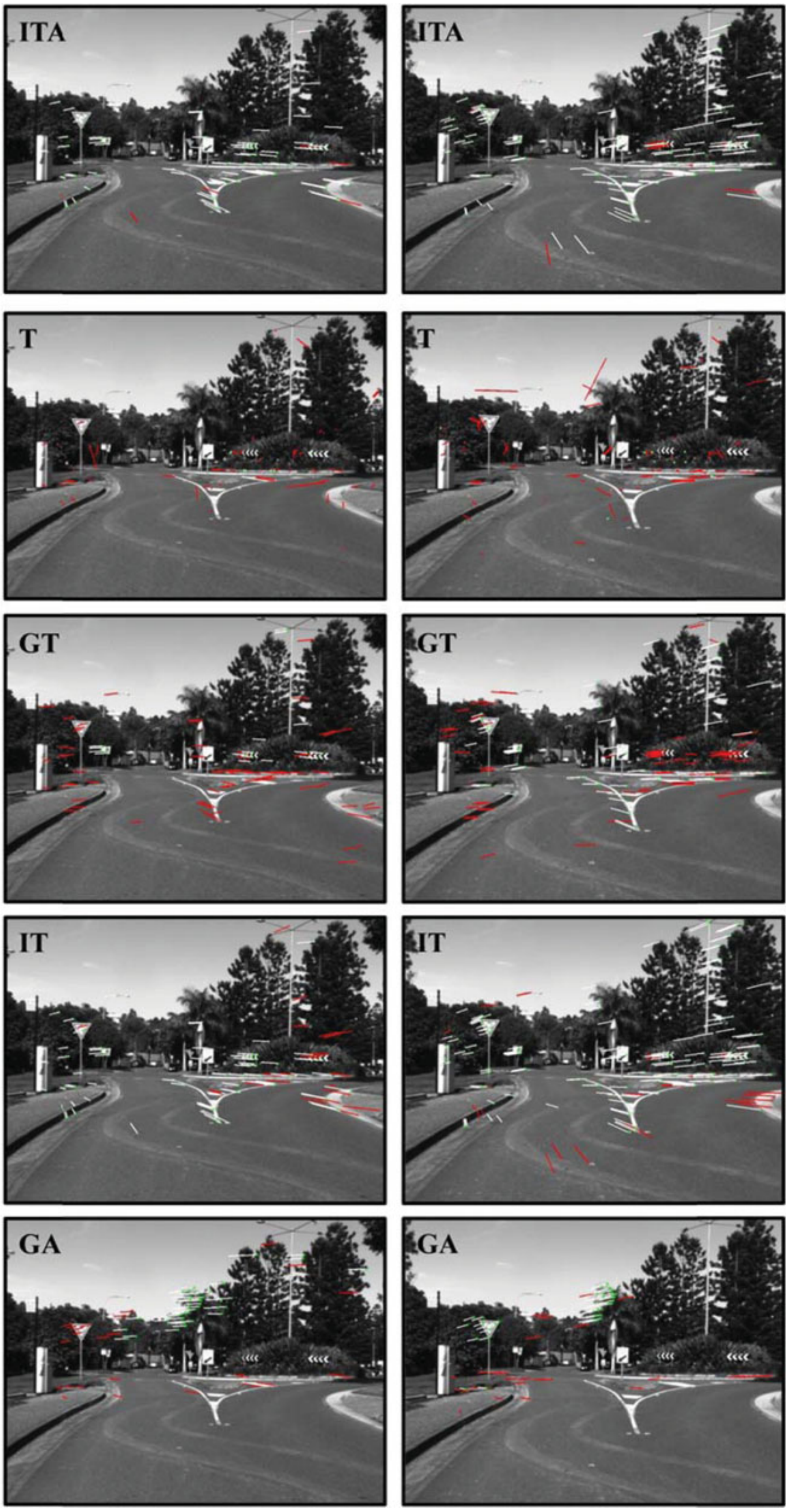

Fig. 8. Bump in the Roundabout case: white optical flow-inliers; red optical flow-outliers; green current position of the tracked feature.

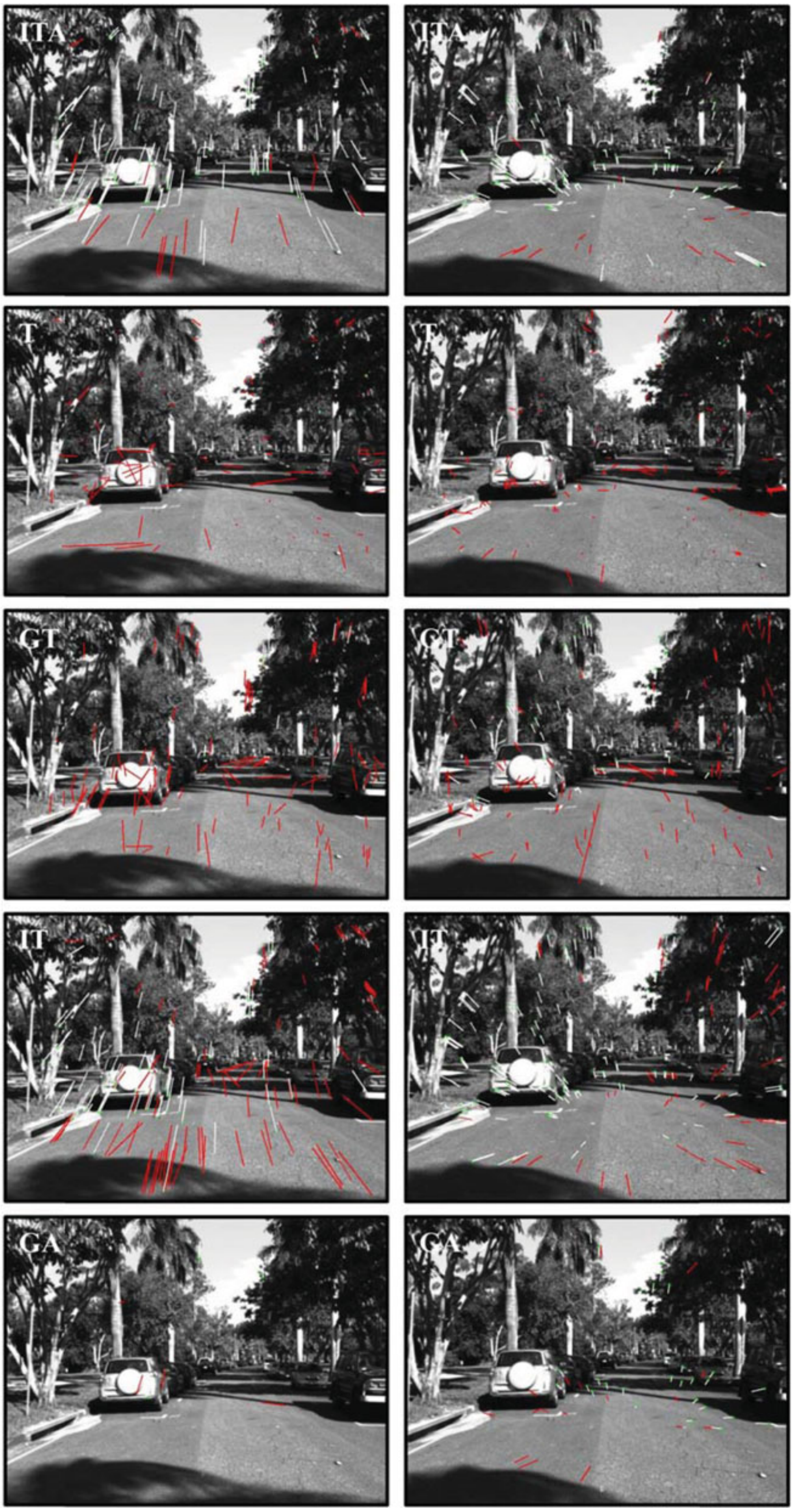

Fig. 9. 1st Humped crossing case: white optical flow-inliers; red optical flow-outliers; green current position of the tracked feature.

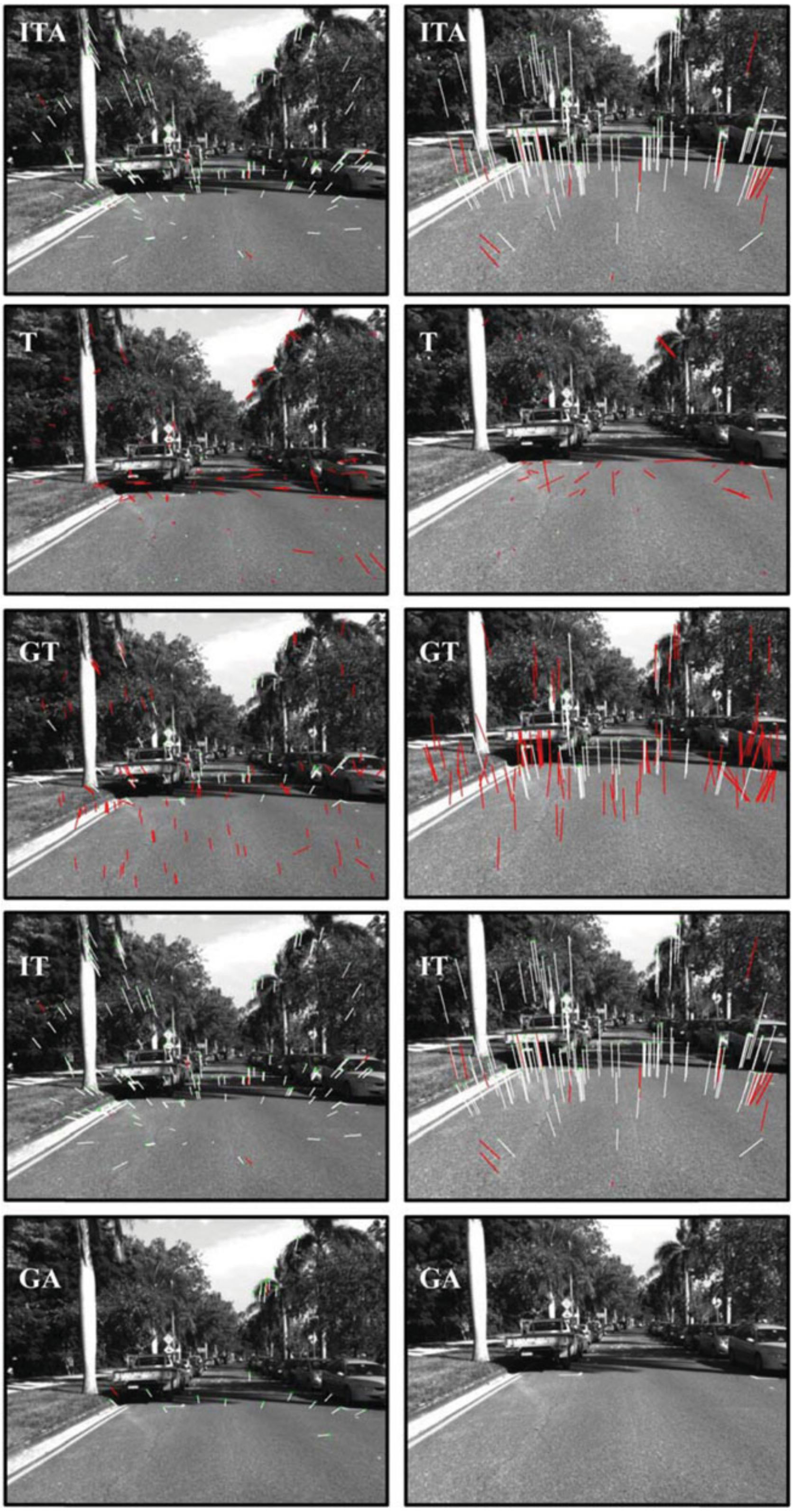

Fig. 10. 2nd Humped crossing case: white optical flow-inliers; red optical flow-outliers; green current position of the tracked feature.

Table III. KLT techniques: tracking performances, processing time, and ratio at different frequencies (for three challenging areas) on dataset[35].

Tracking performances (%)		ITA	T	GT	IT	GA
Bump in a roundabout	10 Hz	**97.2**	16.3	62.1	89.5	81
	5 Hz	**89.9**	11.6	28	63.5	78.9
	3 Hz	**79.2**	11.6	25.1	49	33.9
1st Humped crossing	10 Hz	**95.9**	21.3	57.2	89.6	66.3
	5 Hz	**86.9**	11	24.3	63.7	61.2
	3 Hz	**88.1**	18.2	17.7	51.1	46.9
2nd Humped crossing	10 Hz	**96.9**	13.6	60.4	87.7	87
	5 Hz	**88**	14.8	26.5	76.4	63.5
	3 Hz	**91.4**	13.2	18.1	69.1	34.9

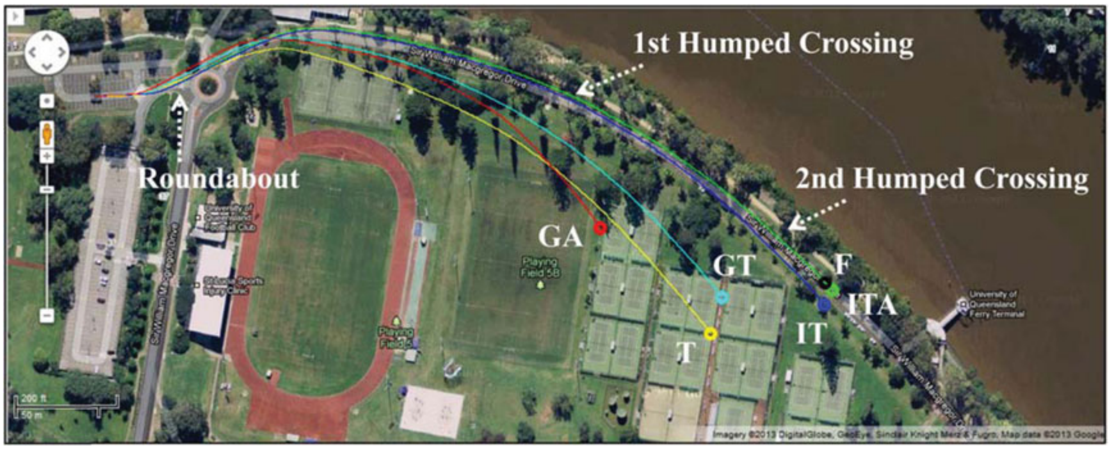

Fig. 11. Trajectories generated with KLT Techniques combined with double dogleg optimization method at 10 Hz, circle represent the end of each trajectory: green ITA; blue IT; cyan GT; yellow T; red GA; black F ground truth reference for final position (defined in Section 5.3 and Fig. 12).

First, T, GT, and GA have a lower rate of tracked features compared to ITA and IT (see Section 5.1). Furthermore, the majority of these features belong to far objects which are in fact coarse and very inaccurate data. Thus, estimation of the motion gives an under estimated travelled distance when it does not fail. Errors accumulation logically leads to large drifts (Fig. 11).

5.3. *Motion estimation performances*

This sub-section shows the performance of motion estimation regardless of the feature tracking method. The optimisation methods for motion estimation consist of DDL, DL, and SBA implementation of LM.[36] ITA algorithm will be used for those three methods as the feature tracking technique. Additionally, a filtered INS/GPS trajectory is included in the comparison test. Notably, we found after having aligned INS/GPS trajectory on a satellite image map, it slightly deviates from the road path.

Thus, we defined an as accurate as possible final ground truth position F according to the satellite map and the final images in the dataset[35] as illustrated in Fig. 13. Figure 12 shows the trajectories generated with the three techniques at 10 Hz. The trajectory generated using DDL is the one that remains the closest to the INS/GPS. The trajectories generated using DL and LM start slightly drifting from INS/GPS trajectory approximately at half way. However, the drift is not penalising much their respective final positions.

Table IV gives the 2D squared root error of DDL, DL, and LM using ITA as feature tracking approach for all the image sequence frequencies. At 10 Hz, DDL combined with ITA is closer to F than the INS/GPS with a 2D error of about 1% of the travelled distance.

Table IV. Final relative 2D position error against final position F for the different optimization methods using ITA at each frequency on dataset[35] full sequence.

		INS/GPS	Double dogleg (ITA)	Dogleg (ITA)	Levenberg–Marquardt (ITA)
10 Hz	2D RMS (m)	9.78 m	5.03 m	10.33 m	11.74 m
	2D RMS (%)	1.99 %	1.02 %	2.10 %	2.39 %
5 Hz	2D RMS (m)	9.78 m	24.93 m	32.01 m	55.35 m
	2D RMS (%)	1.99 %	5.07 %	6.52 %	11.27 %
3 Hz	2D RMS (m)	9.78 m	36.96 m	44.01 m	146.5 m
	2D RMS (%)	1.99 %	7.53 %	8.98 %	29.85 %

Fig. 12. Trajectories generated with ITA combined with different optimization methods at 10 Hz: blue INS/ INS/GPS; green DDL; cyan DL; yellow LM; black circle final position F. Left full map/right zoom on the final position.

Fig. 13. Left—Zoom on the final position (white dot) in front of the "Head Hump" road marking highlighted with the red dashed line; Right—Related image from the dataset of the vehicle final position in front of the "Head Hump" road marking highlighted with the red dashed line.

When the image sequence frequency gets lower, motion estimation accuracy logically decreases for all the techniques. However, trajectories generated using DDL, and DL using ITA remains under 10% error of the travelled distance. This demonstrates the robustness of our technique to severe scaling in feature tracking but also in the motion estimation. Table V confirms that DDL is the technique that requires the least iterations to converge to a solution. In Fig. 12 and Table IV, we demonstrated that INS/GPS can be challenged by our method in terms of accuracy in a certain context.

To complement this study and also to show the robustness of our method, we tested our algorithm on KITTI Vision Benshmark[37] raw data sequences (http://www.cvlibs.net/datasets/kitti/raw_data.php). This dataset consists of 1242×375 rectified stereo images as well as synchronised GPS and IMU data (accelerometer and gyroscope) acquired at 10 Hz. Contrary to the first dataset[35] the IMU are non-calibrated and the update rate is the same as the visual information. Consequently, it affects

Table V. Average number of iterations for all frequencies on dataset[35] full sequence with the different optimisation methods.

	Double dogleg (ITA)	Dogleg (ITA)	Levenberg–Marquardt (ITA)
Iterations	7.1	9.8	12.6

Table VI. Final relative 2D and 3D position error for ITA using double dogleg and LiViso2 methods on KITTI vision Benchmark raw data sequences (http://www.cvlibs.net/datasets/kitti/raw_data.php) and on dataset[35].

				ITA with DDL		LibViso2[38]	
		Distance (m)	# of Frames	2D RMS %	3D RMS %	2D RMS %	3D RMS %
KITTI	2011_09_26_0001	106.75	108	**0.50**	**0.53**	0.95	0.96
	2011_09_26_0002	80.64	77	**0.34**	**0.34**	1.40	1.47
	2011_09_26_0005	69.10	154	**6.28**	**6.28**	7.37	7.38
	2011_09_26_0009	333.36	447	2.71	2.71	**1.42**	**1.43**
	2011_09_26_0014	164.2	314	**1.62**	**1.99**	2.34	2.54
	2011_09_26_0027	344.64	188	**0.82**	**0.82**	1.47	1.47
	2011_09_26_0051	253.08	438	**0.24**	**0.26**	0.92	0.94
	2011_09_26_0084	238.96	383	**0.36**	**0.37**	1.01	1.08
	2011_09_26_0091	198.09	340	**0.20**	**0.24**	0.52	0.54
	2011_09_260.5018_0117	323.14	660	2.57	2.59	**2.04**	**2.06**
QUT[35]	10 Hz	501.42	600	**1.02**	–	12.36	–

the integration of these measurements over the time. It is especially true for the accelerations. This, results into inaccurate inertial angular/position data at the available sampling rate. For this reason, the use of our algorithm using double dogleg with ITA could be only run at 10 Hz and could not be re-sampled at lower frequency as for dataset.[35]

Table VI shows the results on 10 sequences from KITTI Vision Benshmark[37] of our algorithm and libViso2 method (http://www.cvlibs.net/datasets/kitti/raw_data.php) against the provided GPS taken as reference here.

On overall our algorithm and LibViso2[38] method give accurate results achieving a position error of <1% of the travelled distance on certain cases, and without exceeding 3% position error of the travelled distance (expect for sequence 2011_09_26_0005, but GPS reference might be biased).

Except for the sequences 2011_09_26_0009 and 2011_09_26_00117 in KITTI Vision Benshmark[37] raw data sequences (http://www.cvlibs.net/datasets/kitti/raw_data.php), our algorithm is more accurate than LibViso2[38] method.

However, LibViso2[38] showed more difficulty on dataset[35] to keep the same accuracy than for KITTI Vision Benshmark raw data sequences (http://www.cvlibs.net/datasets/kitti/raw_data.php). As mentioned in Sections 5.1 and 5.2, this ability of ITA to keep tracking features close to the camera presenting large optical flows make a real difference comparing to LibViso2[38] on dataset.[35]

Therefore, we demonstrated that with an efficient and clever association between inertial data and visual information we are able to enhance feature tracking of the conventional KLT.

This solution is also independent from external sources of information (e.g., satellites) and can thus be employed as an interesting alternative to GPS or INS/GPS. Additionally, by offering the ability to process low frame rate sequences while keeping good performance and low computation meaning that the IMU-assisted KLT can be easily implemented for real-time applications.

6. Conclusion and Future Work

In this paper, we have presented a new feature tracking technique called IMU-assisted KLT greatly improving the performance of the conventional KLT while reducing its processing time by a factor 2. Initial detected features are accurately projected into the subsequent stereo images by using 3D geometry and stereoscopic properties to combine visual information with the inter-frame full inertial data. This significantly narrows the matching search region and consequently reduces ambiguity. Resulting adaptive local tracking windows enable us to precisely bounder areas of the inertial predictive features. This gives the ability to the IMU-assisted KLT to maintain a high tracking rate and to remain robust against scaling by handling very large motions while keeping a low computational cost. This also guarantees accurate tracking of close features which are better quality. Results obtained with the proposed feature tracker demonstrate higher performance than gyro-aided affine model, gyro-only translational model. Even in the worst cases of severe scale change, the rate of successful tracks does not go below 80% while the feature tracking performance of other techniques clearly collapse. It has been shown that tracked features with the IMU-assisted KLT enhances the estimated visual-based motion. Additionally, double dogleg optimisation technique was adopted in the motion estimation scheme. Its association to our IMU-assisted KLT method offers a better alternative than LM and GN on different datasets. VO performances lie below or around 1% error of the travelled distance and it also demonstrates higher accuracy than INS/GPS in certain conditions. Future work, aims to develop an embedded navigation technology based on this research including a small IMU-stereo camera set up capable of reproducing same accurate results.

Acknowledgements
The research leading to these results has received funding from Thales and E.S.A (European Space Agency). We would like to thank them for their help and support.

References

1. D. Nister, O. Naroditsky and J. Bergen, “Visual Odometry,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Washington, D.C., USA, vol. 1, (Jun. 2004) pp. 652–659.
2. M. Boulekchour, N. Aouf and M. Richardson, “Robust L∞ convex pose-graph optimisation for monocular localisation solution for unmanned aerial vehicles,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 229(10), 1903–1918 (2014).
3. T. Mouats, N. Aouf, L. Chermak and M. Richardson, “Thermal stereo odometry for UAVs,” *IEEE Sensors J.* **15**(11), 6335–6347 (2015).
4. A. Howard, “Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles,” *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Nice, France (Sep. 2008) pp. 3946–3952.
5. N. Sünderhauf, P. Protzel, “Stereo odometry — a review of approaches”, Chemnitz University of Technology Technical Report, March, 2007.
6. B. D. Lucas and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision,” *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, vol. 2, (1981) pp. 674–679.
7. J. Shi and C. Tomasi, “Good Features to Track,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, W.A., USA, (Jun. 1994) pp. 593–600.
8. S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework,” *Int. J. Comput. Vis.* **56**(3), 221–255 (2004).
9. J.-Y. Bouguet, “Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker Description of the Algorithm,” *OpenCV documentation, Intel Corporation, Microprocessor Research Labs*, (1999).
10. T. Zinßer, C. Gräßl and H. Niemann, “Efficient feature tracking for long video sequences,” **In**: *Joint Pattern Recognition Symposium*, (Springer Berlin Heidelberg, 2004) pp. 326–333.
11. S. N. Sinha, J.-M. Frahm, M. Pollefeys and Y. Genc, “Feature tracking and matching in video using programmable graphics hardware,” *Mach. Vis. Appl.* **22**(1), 207–217 (2007).
12. H.-K. Lee, K.-W. Choi, D. Kong, and J. Won, “Improved Kanade-Lucas-Tomasi Tracker for Images with Scale Changes,” *Proceedings of the IEEE International Conference on Consumer Electronics*, Berlin, Germany, (Sep. 2013) pp. 33–34.
13. M. Hwangbo, J.-S. Kim and T. Kanade, “Gyro-aided feature tracking for a moving camera: Fusion, auto-calibration and GPU implementation,” *Int. J. Robot. Res.* **30**(14), 1755–1774 (2011).
14. Y.-G. Ryu, H.-C. Roh and M.-Y. Chung, “Video Stabilization for Robot Eye Using IMU-Aided Feature Tracker,” *Proceedings of the IEEE International Conference on Control Automation and Systems*, Gyeonggi-do, Korea, (Oct. 2010) pp. 1875–1878.
15. S. Tanathong and I. Lee, “Translation-based KLT tracker under severe camera rotation using GPS/INS data,” *IEEE Geoscience Remote Sensing Lett.* **11**(1), 64–68 (2013).
16. D. Scaramuzza and F. Fraundorfer, “Visual odometry [tutorial],” *IEEE Robot. Autom. Mag.* **18**(4), pp. 80–92 (2011).
17. J. Goslinski, M. Nowicki and P. Skrzypczynski, “Performance comparison of EKF-based algorithms for orientation estimation on android platform,” *IEEE Sensors J.* **15**(7), pp. 3781–3792 (2015).

18. K. Konolige, M. Agrawal and J. Sola, "Large-Scale Visual Odometry for Rough Terrain," **In**: *Robotics Research*, (Springer Berlin Heidelberg, 2011) pp. 201–212.
19. J.-P. Tardif, M. George, M. Laverne, A. Kelly and A. Stentz, "A New Approach to Vision-Aided Inertial Navigation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, (Oct. 2010) pp. 4161–4168.
20. A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*, Reston, V.A., USA, vol. 174. (AIAA, Sep. 1997).
21. A. Makadia and K. Daniilidis, "Correspondenceless Ego-Motion Estimation Using an IMU," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (2005) pp. 3534–3539.
22. F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Trans. Robot.* **24**(5), pp. 1143–1156 (2008).
23. A. Martinelli, "Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Trans. Robot.* **28**(1), pp. 44–60 (2012).
24. S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.* **34**(3), pp. 314–334 (2015).
25. R. I. Hartley and P. Sturm, "Triangulation," *Comput. Vis. Image Understanding* **68**(2), pp. 146–157 (1997).
26. A. Geiger, J. Ziegler and C. Stiller, "Stereoscan: Dense 3d Reconstruction in Real-Time," *Proceedings of the IEEE Symposium on Intelligent Vehicles*, Baden-Baden, Germany, (Jun. 2011) pp. 963–968.
27. N. Sünderhauf, K. Konolige, S. Lacroix and P. Protzel, "Visual Odometry using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle," **In**: *Autonome Mobile Systeme*, (Springer Berlin Heidelberg, 2006) pp. 157–163.
28. C. Wu, S. Agarwal, B. Curless and S. M. Seitz, "Multicore Bundle Adjustment," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, R.I., USA (2011) pp. 3057–3064.
29. K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly Appl. Math.* **2**, 164–168 (1944).
30. D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.* **11**(2), pp. 431–441 (1963).
31. J. E. Dennis, Jr and H. H. W. Mei, "Two new unconstrained optimization algorithms which use function and gradient values," *J. Optimization Theory Appl.* **28**(4), pp. 453–482 (1979).
32. M.J. Powell, "A new algorithm for unconstrained optimization", Nonlinear programming, *Proceedings of a Symposium Conducted by the Mathematics Research Center, the University of Wisconsin–Madison*, (May 1970) pp. 31–65
33. M. L. A. Lourakis and A. A. Argyros, "Is Levenberg-Marquardt the Most Efficient Optimization Algorithm for Implementing Bundle Adjustment?," *Proceedings of the IEEE International Conference on Computer Vision*, Beijing, China, vol. 2, (Oct. 2005) pp. 1526–1531.
34. W. Sun and Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, vol. 1. (Springer Berlin Heidelberg, 2006).
35. M. Warren, D. McKinnon, H. He and B. Upcroft, "Unaided Stereo Vision Based Pose Estimation," *Proceedings of the Australasian Conference on Robotics and Automation*, Brisbane, Australia, (Dec. 2010).
36. M. I. Lourakis and A. A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Trans. Math. Software* **36**(1), pp. 1–30 (2009).
37. A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.* (2013).
38. http://www.cvlibs.net/software/libviso/.