Research Notes

Mohsin Dalvi – DT17MEC050

May 5, 2021

Hello World! [1]

1 Summary of Jeroen Hol

Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors
20088
Division of Automatic Control
Department of Electrical Engineering
Linköping University, Sweden

Augmented reality (AR) involves overlaying a real scene with computer generated graphics in real-time by showing the virtual objects on see-through head-mounted displays or superimposing them on live video imagery. In order to have realistic augmentation, i.e., positioning and aligning the virtual objects correctly with the real world, it is essential to know the position and orientation (pose) of the camera with high accuracy and low latency. A ?vision only approach? gives good absolute accuracy, but is difficult to run at high frame rate and is not robust during fast motions. An inertial measurement unit (IMU) is accurate for a short period and is drift-prone for longer time scales.

In this thesis, pose estimation is approached using a combination of a camera and an IMU. Using a 3D scene model containing natural landmarks removes the need for costly and time consuming procedure of preparing the environment, and allows for AR applications outside dedicated studios. The IMU unit provides rapid measurements of acceleration and angular velocity. The computer vision system generates correspondences between the camera image and a 3D scene model that contains positions of various natural markers and is generated offline using images and/or drawings of the scene. The camera pose is estimated by combining inertial and visual measurements in a sensor fusion model that solves a non-linear state estimation problem in real-time.

Some aspects have been previously published in
F. Gustafsson, T. B. Schön, and J. D. Hol. Sensor fusion for augmented reality. In Proceedings of 17th International Federation of Automatic Control World Congress, Seoul, South Korea, July 2008. Accepted for publication.
J. D. Hol, T. B. Schön, H. Luinge, P. J. Slycke, and F. Gustafsson. Robust real-time tracking by fusing measurements from inertial and vision sensors. Journal of Real-Time Image Processing, 2(2):149?160, Nov. 2007. doi:10.1007/s11554-007-0040-2.
J. D. Hol, T. B. Schön, F. Gustafsson, and P. J. Slycke. Sensor fusion for augmented reality. In Proceedings of 9th International Conference on Information Fusion, Florence, Italy, July 2006b.

doi:10.1109/ICIF.2006.301604.

J. D. Hol, T. B. Schön, and F. Gustafsson. On resampling algorithms for particle fil- ters. InProceedings of Nonlinear Statistical Signal Processing Workshop, Cambridge, UK, Sept. 2006a. doi:10.1109/NSSPW.2006.4378824.

G. Hendeby, J. D. Hol, R. Karlsson, and F. Gustafsson. A graphics processing unit implementation of the particle filter. In Proceedings of European Signal Processing Conference, Poznán, Poland, Sept. 2007.

1.1 Robust real-time tracking by fusing measurements from inertial and vision sensors

The pose of a camera is estimated in real-time by fusing measurements from inertial sensors (accelerometers and rate gyroscopes) and a camera. The non-linear state estimation is based on a multi-rate extended Kalman filter using a dynamic model with 22 states, where 100 Hz inertial measurements and 12.5 Hz correspondences from vision are processed.Vision measurements are based on natural landmarks, which are detected guided by pose predictions. The measurements from the sensors are used directly rather than being processed to a vision based pose or an inertial based pose. Accurate short time pose estimates are available using information from IMU hereby reducing the need for fast vision updates, which would otherwise require high frame rates and compute.

Coordinate systems used:

- Earth e: It is fixed to earth and generally vertically aligned. The camera pose is estimated w.r.t. it and features of the scene are modeled in it.

- Camera c: It is attached to the moving camera. Its origin is located in the optical center of the camera, with the z-axis pointing along the optical axis. The camera acquires its images in the image plane (i). This plane is perpendicular to the optical axis.

- Body b: It is the coordinate system of the IMU attached by a constant translation and rotation w.r.t. the camera.

These coordinate systems are used to denote geometric quantities, for instance, c^e is the position of the camera coordinate system expressed in the earth system and R^{cb} is the rotation matrix from the body system to the camera system.

The IMU consists of 1200 deg /s gyroscopes and 5g accelerometers. The signals from the inertial components of the IMU are synchronously measured at 100 |Hz using a 16 bit A/D converter. A temperature sensor is added to compensate for the temperature dependency of the sensing components. The IMU sensor is subjected to a calibration procedure to calibrate for the exact physical alignment of each component, the gains, the offsets, and the temperature relations of the gains and offsets. With these, a 3D angular velocity vector and a 3D accelerometer vector, both resolved in the body coordinate system, are computed. The calibrated gyroscope signal $y_{\omega,t}$ contains measurements of the angular velocity $\omega_{eb,t}^b$ from body to earth ($_{eb}$) expressed in the body coordinate system (^{b}):

$$y_{\omega,t} = \omega_{eb,t}^b + \delta_{\omega,t}^b + e_{\omega,t}^b \tag{1}$$

Here, $\delta_{\omega,t}^b$ is the low-frequency offset fluctuations due toe unmodeled acceleration dependency and $e_{\omega,t}^b$ is zero mean white noise. A change in orientation is obtained by integration of the gyroscope signal without relying on any infrastructure. However, accuracy in orientation deteriorates for periods longer than a few seconds.

The calibrated accelerometer signal $\boldsymbol{y}_{a,t}$ contains measurements of the combination of the body acceleration vector $\ddot{\boldsymbol{b}}_t$ and the gravity vector $\boldsymbol{g}$ both expressed in the body coordinate system (^{b}):

$$\boldsymbol{y}_{a,t} = \ddot{\boldsymbol{b}}_t^b - \boldsymbol{g}_t^b + \boldsymbol{\delta}_{a,t}^b + \boldsymbol{e}_{a,t}^b \tag{2}$$

Here, $\boldsymbol{\delta}_{a,t}^b$ is the low-frequency offset and $\boldsymbol{e}_{a,t}^b$ is zero mean white noise. Gravity expressed in body coordinates depends on the orientation of the sensor unit. Once the orientation is known, the accelerometer signal is used to estimate the acceleration, or alternatively, once the acc^n is known, the direction of the vertical is estimated. Acceleration is integrated twice to obtain change in position, as long as accurate orientation estimate is available from the gyroscopes. However, accuracy of position change deteriorates quickly as a result of double integration and orientation errors.

The camera has a perspective lens with focal length of $3.2\,\mathrm{mm}$ and captures images with a resolution of 320×240 pixels at a frame-rate of $12.5\,\mathrm{Hz}$. The camera is triggered by the IMU clock allowing for synchronized measurements. Extracting camera position and orientation from images is a known and well studied problem in computer vision. The key ingredient is to find correspondences, i.e., relations between features in the image which correspond to an element in the scene model. Point correspondences $\boldsymbol{z}^c \leftrightarrow \boldsymbol{z}^i$ are the relation between 3D points $\boldsymbol{z}^c$ and 2D image points $\boldsymbol{z}^i$. For a perspective lens and a pinhole camera, the correspondence relation is

$$\boldsymbol{z}^i = \begin{bmatrix} f z_c^x / z_c^z \\ f z_c^y / z_c^z \end{bmatrix} + \boldsymbol{e}^i \tag{3a}$$

or equivalently,

$$\boldsymbol{0} \approx \begin{bmatrix} -f\boldsymbol{I}_2 & \boldsymbol{z}_t^i \end{bmatrix} \boldsymbol{z}_t^c = \begin{bmatrix} -f\boldsymbol{I}_2 & \boldsymbol{z}_t^i \end{bmatrix} \boldsymbol{R}_t^{cc} (\boldsymbol{z}^c - \boldsymbol{z}_t^c) \tag{3b}$$

where f is the focal length, $\boldsymbol{I}_2$ is 2×2 identity matrix, $\boldsymbol{e}_t^i$ is zero-mean white noise. The camera pose depends on the rotation matrix R^{ce} and the position $\boldsymbol{c}^c$. Hence, given sufficient correspondences and a calibrated camera, the camera pose can be found.

The computer vision implementation is based on a sum of absolute difference (SAD) block matcher in combination with a planar patch or free-form surface model of the scene. Both pixel data and 3D positions are stored for each feature. While tracking, search templates are generated by warping the patches in the model according to homographies calculated from the latest prediction of the camera pose. These templates are then matched with the current calibrated camera image using the block matcher. In this way correspondences are generated.

1.1.1 Sensor fusion

Vision in combination with the map gives accurate absolute pose information at a low rate, but experiences problems during moderately fast motions. The IMU provides high rate relative pose information regardless of the motion speed, but becomes inaccurate after a short period of time. By fusing information from both sources, it is possible to obtain robust camera pose estimates.

One way is to extend vision-based methods by using pose predictions from the IMU to determine where in the image the features are to be expected. Once detected, the features are used to calculate the pose, and this pose is then used as a starting point for the next pose prediction by the IMU. Another way is to consider IMU to be the main sensor, which is quite common in the navigation industry. Then vision is used for error correction, similar to how radio beacons or global positioning system (GPS) are used to correct the drift in an inertial navigation system (INS).

The objective in filtering is to extract as much information as possible from the measurements. More specifically, this amounts to finding the best possible estimate of the filtering probability density function (pdf) $p(x_t|y_{1:t})$ where $y_{1:t} \triangleq \{y_1, \ldots, y_t\}$. The objective in sensor fusion is to recursively in time estimate the state in the dynamic model,

$$x_{t+1} = f_t(x_t, u_t, v_t) \tag{4a}$$

$$y_t = h_t(x_t, u_t, e_t) \tag{4b}$$

where $x_t \in \mathbb{R}^{n_x}$ denotes state estimate, $y_t \in \mathbb{R}^{n_y}$ denotes measurements from sensors, v_t and e_t denote the stochastic process and measurement noise respectively, $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_x}$ is process model that describes evolution of state over time, and $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_e} \to \mathbb{R}^{n_y}$ is measurement model that describes how measurements from IMU and camera relate to the state.

The goal is to infer information from measurements y_t onto the state x_t by computing the filtering pdf $p(x_t|y_{1:t})$. The filtering pdf contains everything there is to know about the state at time t, given the information in all the past measurements $y_{1:t}$. The approximation of $p(x_t|y_{1:t})$ is used to form different (point) estimates, including maximum likelihood estimates, confidence intervals and expectation estimate $\hat{x}_t = E(x_t|y_{1:t})$. The key element in solving the nonlinear state estimation problem in real-time is the propagation of $p(x_t|y_{1:t})$ over time.

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1}) \, \mathrm{d}x_t} \tag{5a}$$

$$p(x_{t+1}|y_{1:t}) = \int p(x_{t+1}|x_t)p(x_t|y_{1:t}) \, \mathrm{d}x_t \tag{5b}$$

Eqs. 5a and 5b are called measurement update and time update, respectively. The sensor fusion problem is reduced to propagating eq. 5 over time as new measurements arrive. When eq. 4 is linear and noise is Gaussian, then involved densities become Gaussian and updating recursions are given by the Kalman Filter (KF). It is sufficient to assume that noise enters additively, according to

$$x_{t+1} = f_t(x_t) + v_t \tag{6a}$$

$$y_t = h_t(x_t) + e_t \tag{6b}$$

Additive noise modifies Eq. 5 to

$$p(x_{t+1}|x_t) = p_{v_t}(x_{t+1} - f_t(x_t)) \tag{7a}$$

$$p(y_t|x_t) = p_{e_t}(y_t - h_t(x_t)) \tag{7b}$$

where $p_{v_t}(\cdot)$ and $p_{e_t}(\cdot)$ denote the pdf's for the noise v_t and e_t, respectively.

The state vector consists of position of the IMU (the body coordinate system) expressed in the earth system b^e, its velocity $\dot{\boldsymbol{b}}^e$ and acceleration $\ddot{\boldsymbol{b}}^e$, the orientation (quaternion) of the body wrt earth q^{be}, its angular velocity ω_{eb}^b, the gyroscope bias δ_ω^b and the accelerometer bias δ_a^b. The state vector has dimension 22 and is given by

$$x_t = \begin{bmatrix} \boldsymbol{b}_t^e \\ \dot{\boldsymbol{b}}_t^e \\ \ddot{\boldsymbol{b}}_t^e \\ q_t^{be} \\ \boldsymbol{\omega}_{eb,t}^b \\ \boldsymbol{\delta}_{\omega,t}^b \\ \boldsymbol{\delta}_{a,t}^b \end{bmatrix} \tag{8}$$

The process model is given by

$$\boldsymbol{b}_{t+1}^e = \boldsymbol{b}_t^e + T\dot{\boldsymbol{b}}_t^e + \tfrac{T^2}{2}\ddot{\boldsymbol{b}}_t^e \tag{9a}$$

$$\dot{\boldsymbol{b}}_{t+1}^e = \dot{\boldsymbol{b}}_t^e + T\ddot{\boldsymbol{b}}_t^e \tag{9b}$$

$$\ddot{\boldsymbol{b}}_{t+1}^e = \ddot{\boldsymbol{b}}_t^e + \boldsymbol{v}_{b,t}^e \tag{9c}$$

$$q_{t+1}^{be} = \exp\!\left(-\tfrac{T}{2}\boldsymbol{\omega}_{eb,t}^b\right) \odot q_t^{be} \tag{9d}$$

$$\boldsymbol{\omega}_{eb,t+1}^b = \boldsymbol{\omega}_{eb,t}^b + \boldsymbol{v}_{\omega,t}^b \tag{9e}$$

$$\boldsymbol{\delta}_{\omega,t+1}^b = \boldsymbol{\delta}_{\omega,t}^b + \boldsymbol{v}_{\delta_\omega,t}^b \tag{9f}$$

$$\boldsymbol{\delta}_{a,t+1}^b = \boldsymbol{\delta}_{a,t}^b + \boldsymbol{v}_{\delta_a,t}^b \tag{9g}$$

where the quaternion multiplication and exponential are defined according to

$$\begin{bmatrix} p_0 \\ \boldsymbol{p} \end{bmatrix} \odot \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix} \triangleq \begin{bmatrix} q_0 q_0 - \boldsymbol{p}\cdot\boldsymbol{q} \\ q_0\boldsymbol{q} + q_0\boldsymbol{p} + \boldsymbol{p}\times\boldsymbol{q} \end{bmatrix} \tag{10a}$$

$$\exp(\boldsymbol{v}) \triangleq \begin{bmatrix} \cos\|\boldsymbol{v}\| \\ \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\sin\|\boldsymbol{v}\| \end{bmatrix} \tag{10b}$$

The measurement models are given by

$$\boldsymbol{y}_{a,t} = R_t^{be}(\ddot{\boldsymbol{b}}_t^e - \boldsymbol{g}^e) + \boldsymbol{\delta}_{a,t}^b + \boldsymbol{e}_{a,t}^b \tag{11a}$$

$$\boldsymbol{y}_{\omega,t} = \boldsymbol{\omega}_{eb,t}^b + \boldsymbol{\delta}_{\omega,t}^b + \boldsymbol{e}_{\omega,t}^b \tag{11b}$$

$$\boldsymbol{y}_{c,t} = \begin{bmatrix} -f\boldsymbol{I}_2 & \boldsymbol{z}_t^i \end{bmatrix} \boldsymbol{R}^{cb}(\boldsymbol{R}_t^{be}(\boldsymbol{z}_t^e - \boldsymbol{b}_t^e) - \boldsymbol{c}^b) + \boldsymbol{e}_{a,t}^b \tag{11c}$$

The rotation matrix $\boldsymbol{R}_t^{be}$ is constructed from q_t^{be}. The transformation from body to camera coordinate system is included in vision measurement.

Algorithm 1 uses models in Eqs. 9 and 11 to perform updates given in Eq. 5. The algorithm runs in real time with inertial measurements at $100\,\text{Hz}$ and frame rates up to $25\,\text{Hz}$ and uses EKF to compute pose estimates.

Algorithm 1: Recursive camera pose calculation

Input : Inertial data $\{\boldsymbol{y}_{a,t}, \boldsymbol{y}_{\omega,t}\}_{t=0}^N$
         magnetometer data $\{\boldsymbol{y}_{m,t}\}_{t=0}^N$
Output: Pose estimate $\boldsymbol{x}_t$ for $t = 1, \cdots, N$

1 **for** $t = 1$ **to** N **do**
2   Time update. Calculate $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ by propagating $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ through process model Eq. 9.
3   Accelerometer and gyroscope measurement update using model Eqs. 11a and 11b.
4         $\boldsymbol{x}_t \sim p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$
5   **if** *there is a new image from the camera* **then**
6   Predict feature positions from the scene model using $\hat{\boldsymbol{x}}_t = E(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$.
7   Detect features in the image.
8   Measurement update with the found point correspondences using model Eq. 11c.
9           $\boldsymbol{x}_t \sim p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$
10   **end if**
11 **end for**

The camera and the IMU both deliver measurements which are resolved respectively in the camera and the body coordinate system .As the sensors are physically translated and rotated with respect to each other, this rigid transformation should be taken into account while fusing the measurements.

The high frequency inertial measurement updates provide a rather accurate state estimate when a new image is acquired. This implies that the feature positions can be predicted with an improved accuracy, which in turn makes it possible to use a guided search in the image using reduced search regions. The algorithm can calculate the expected covariance of a measurement.

The information from the IMU makes Algorithm 1 robust for temporary absence of vision due to motion blur, obstruction of the camera or unmodeled scene. Without vision measurements, the estimates will eventually drift away. The algorithm is flexible and can be rather straightforwardly extended to include other information sources. When the sensor unit is static during initialization, the IMU provides partial or full orientation estimates (using magnetometers). This information is used to constrain the search space when initializing from vision.

Vision-only methods suffer from scale ambiguity, since projections are used. Changing the scale of a scene model will give scaled, but indistinguishable poses. As sensor fusion utilizes position information both from the camera and the IMU, the scale is relevant and these quantities must have identical units. Introducing a metric scale into the scene model solves this issue. An existing scene model with arbitrary scale can be converted by comparing it with a CAD model or by measuring an object with known dimension.

To separate the contribution of gravity in acceleration measurement, the gravity vector is rotated from earth frame to body frame and subtracted. The scene model should be vertically aligned, or the gravity vector should be known in the scene model. But, this is not the case. The performance of the system is extremely sensitive to this alignment, since gravity is typically an order of magnitude larger than normal body accelerations. A misalignment of $1°$ introduces an artificial acceleration of $0.17\,\mathrm{m/s^2}$ which gives rise to a systematic position drift of $8.5\,\mathrm{cm}$ when integrated over $1\,\mathrm{s}$. Hence, even for small errors, a systematic drift is introduced which causes the system to lose track without continuous corrections from correspondences. Drifts followed by a correction gives rise to a saw tooth pattern in the estimates, which is visible as 'jitter'. The gravity vector can be determined by averaging the accelerometer readings over some time, while the camera is stationary in a known pose. However, a preferable method is to record accelerometer measurements while scanning the scene and include this data in the model building procedure to align the scene model vertically.

1.1.2   Implementation considerations

Sensor pose calibration: The problem of determining the relative position and orientation is a well studied problem in robotics where it is known as hand-eye calibration. However, most methods do not apply directly since the IMU does not provide an absolute position reference. Absolute orientation is available when the IMU is stationary, since the accelerometers measure only gravity. The orientation part of the calibration is determined using a slight modification of standard camera calibration procedures. As floors and desks in buildings are better horizontally aligned than walls are vertically aligned, it is recommended to use horizontal surfaces. The calibration pattern is placed on a horizontal surface and accelerometer readings are taken at various camera poses. The camera poses are determined in the camera calibration procedure, from which the vertical directions in the camera frame are determined. The combination of these and the vertical directions in the body frame measured by the accelerometers allows for calculation of the rotation between the frames. This method requires accurate positioning of the calibration pattern.

Time synchronization: It is very important to know exactly when the different measurements are taken. Time synchronization between clocks of multiple sensors is achieved by hardware synchronization, i.e., one central clock is used to trigger the other sensors. This solves the problem of simultaneously started clocks diverging after operating for a while.

Filter tuning: The models have stochastic components which are used to tune the EKF (Refer Table 2.1 in report page 21). For IMU, noise covariance is calculated either experimentally from measurements of a few seconds with a static pose, or directly from manufacturer specification. The noise in vision measurements is harder to determine as accuracy for each match is highly individual, and variations due to lighting conditions, local texture, viewing angle, distance and motion blur cannot be captured by a common noise setting. Hence, it would be beneficial to include accuracy estimation in the image processing algorithms. The current implementation uses a predefined noise covariance. Again, the process model is a random walk in acceleration and angular velocity, and though not informative, the model is useful for tracking uncontrolled motions such as those generated by a human. Also, the motion model is to be considered as a separate source of information, apart from the sensors. Including additional information (such as control signals) in the model improves filter performance. The covariances in the process model can be seen as tuning knobs, controlling the relative importance of the measurements and the process model and as such they are important parameters for stable tracking. The innovations, defined as the difference between a measurement and its expected value, $e_t = y_t - \hat{y}t$, asses whether the models are correctly tuned. The innovations should be normally distributed, and the squared normalized innovations $e_t^{\mathrm{T}} S_t^{-1} e^t$, where S_t is the predicted covariance of the measurement, should have a χ^2 distribution. These performance indicators should be monitored during both testing and normal operation.

1.1.3   Experiments

The sensor unit is mounted onto a high precision 6 degrees of freedom (DoF) ABB IRB1440 industrial robot. The robot will allows repeatable 6 DoF motions andl provides true positions and orientations. The robot has an absolute accuracy of 2 mm and a repeatability of 0.2 mm. This enables systematic and rather objective performance evaluation of various algorithms, based on absolute pose errors instead of the commonly used feature reprojection errors. The sensor unit provides 100 Hz inertial measurements synchronized with 12.5 Hz images. The complete specification is listed in Table 2.1. The scene used for the experiments consists of two orthogonal planar surfaces as shown in Figure 2.3. Because of the simple geometry, the scene model could be constructed from a textured CAD model. Its coordinate system is such that the x-axis points upward and that the y and z-axis span the horizontal plane. Although the scene was carefully positioned, it had to be calibrated w.r.t. gravity as described in Section 2.4.2. It should be emphasized that the scene has been kept simple for experimentation purposes only. The system itself can handle very general scenes and these are modeled using the methods described in Koeser et al. (2007).

With the setup several trajectories have been tested. In this paper, an eight-shaped trajectory, shown in Figure 2.4, will be discussed in detail. The sensor unit traverses this 2.6 m eight-shaped trajectory in 5.4 s, keeping the scene in view at all times. The motion contains accelerations up to 4 m/s 2 and angular velocities up to 1 rad/s. Hence, the motion is quite aggressive and all six degrees of freedom are exited. As the displacement between images is limited to 15 pixels it is still possible to use vision-only tracking, which allows for a comparison between tracking with and without an IMU.

The experiment starts with a synchronization motion, which is used to synchronize the ground truth data from the industrial robot with the estimates from the system. Time synchronization is relevant, since a small time offset between the signals will result in a significant error. After the synchronization, the eight-shaped trajectory

2   Reviews

ID, Title, Author, Journal	Research areas, Tools	Objectives, Methodology, Discussion	Conclusions, My inference, Gaps
• **authyear** • *paper title* • firstname1 surname1 and firstname2 surname2 • journal	• keyword1 • keyword2	• objec • method • concl	• gap
• **aydin2006** • *Quaternion Based Inverse Kinematics for Industrial Robot Manipulators with Euler Wrist* • Yavuz Aydın and Serdar Kucuk • IEEE 3rd International Conference on Mechatronics (ICM 2006)	• wrist IK • quaternion-vector pair for 6-dof pose	• Double quaternion instead of dual quaternion • Approximate translation magnitue d as rotation by angle $\psi = \dfrac{d}{R}$ about normalised vector d.	• Results obtained by using double quaternions are coordinate frame invariant.
• **ge1998** • *Double quaternions for motion interpolation* • Ge, Q J and Varshney, Amitabh and Menon, Jai P and Chang, Chu-Fei • —	• dual quaternions	• Double quaternion instead of dual quaternion • Approximate translation magnitue d as rotation by angle $\psi = \dfrac{d}{R}$ about normalised vector d.	• Results obtained by using double quaternions are coordinate frame invariant.
• **authyear** • *paper title* • firstname1 surname1 and firstname2 surname2 • journal	• keyword1 • keyword2	• objec • method • concl	• gap

ID, Title, Author, Journal	Research areas, Tools	Objectives, Methodology, Discussion	Conclusions, My inference, Gaps
- **authyear** - *paper title* - firstname1 surname1 and firstname2 surname2 - journal	- keyword1 - keyword2	- objec - method - concl	- gap
- **ge1998** - *Double quaternions for motion interpolation* - Ge, Q J and Varshney, Amitabh and Menon, Jai P and Chang, Chu-Fei - —	- dual quaternions	- Double quaternion instead of dual quaternion - Approximate translation magnitue d as rotation by angle $\psi = \dfrac{d}{R}$ about normalised vector d.	- Results obtained by using double quaternions are coordinate frame invariant.
- **thomas** - *Approaching Dual Quaternions from Matrix Algebra* - Frederico Thomas - —	- double quaternion derivation		
End of Notes			

References

[1] Ramkumar Gandhinathan and Lentin Joseph. *ROS Robotics Projects: Build and control robots powered by the Robot Operating System, Machine Learning, and Virtual Reality.* Packt Publishing, 2 edition, 2019.