



Hybridisations Based On Visual Information For The Localisation Of Self-Driving Cars

Jean-Luc Demange, Valentin Demange, Maxime Ferreira, Kevin Honoré, T. Maziol, V. Ollier, Emmanuel Robert

► To cite this version:

Jean-Luc Demange, Valentin Demange, Maxime Ferreira, Kevin Honoré, T. Maziol, et al.. Hybridisations Based On Visual Information For The Localisation Of Self-Driving Cars. ITSNT 2018, International Technical Symposium on Navigation and Timing, Oct 2018, Toulouse, France. 10.31701/itsnt2018.09 . hal-01942266

HAL Id: hal-01942266

<https://enac.hal.science/hal-01942266>

Submitted on 12 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Hybridisations based on visual information for the localisation of self-driving cars

J.L Demange, V. Demange, M. Ferreira, K. Honore, T. Maziol, V. Ollier, E. Robert, *Safran Electronics & Defense*

Emails: kevin.honore@safrangroup.com, valentin.demange@safrangroup.com, maxime.ferreira@safrangroup.com, jean-luc.demange@safrangroup.com, emmanuel.robert2@safrangroup.com

BIOGRAPHIES

All authors are system engineers at Safran Electronics & Defense's R&D centre near Paris. The works presented in this paper are hosted in the R&T team of the Centre of Excellence for Navigation Systems.

ABSTRACT

Safran has been working for several years on autonomy of vehicles. Whether it is airborne with the UAV Patroller, or on the ground with the military vehicle eRider and with the civilian autonomous car in cooperation with Valeo. This paper focuses on the use of visual information to improve the localisation of the car. More precisely, it presents, from a theoretical point of view, the different kind of visual information that can be used in the navigation filter to improve the localisation of the car and the corresponding hybridisation. The efficiency of these hybridisation are evaluated one by one on simulated and/or on real data.

1 INTRODUCTION

Well known as the European leader in inertial navigation and for its in-house inertial technologies, RLG, FOG, HRG... Safran is also a leading player in visual inertial hybridisation techniques whose use is spreading in aerospace, defence and industrial applications.

One of the major challenges to be achieved in next years is to mature positioning solutions for autonomous vehicles such as the UAV Patroller in the air, or the self-driving cars on the ground whether they are designed for military missions like Safran's eRider or for civilian applications like the autonomous car in development at Valeo for instance.



Figure 1. Safran's Patroller UAV



Figure 2. Safran's eRider



Figure 3. Valeo's Drive4U

Localisation is a mandatory function of the self-driving car. The car needs to locate itself relatively to its close environment in order to ensure basic driving: staying in the middle of its lane, stopping at the traffic light, overtaking another car ... And it also needs more global localisation in order to choose the right directions and roads to follow to reach the destination wished by the user.

The highest precision needed for the localisation system of an autonomous car is about a dozen of centimetres. This requirement must be fulfilled all the time, in all weather conditions, on all the roads (including urban canyons where GNSS is neither accurate, nor reliable). This level of accuracy is not reachable with state-of-the-art INS-GNSS localisation, and extra information is needed. For anti-collision purposes, autonomous cars will be equipped with several cameras and LIDARs which can bring information on the movement, and hence the localisation of the car.

Safran is developing four different hybridisations based on visual observation for the autonomous car in a cooperative project with Valeo:

- Based directly on extracted characteristic features, as in classic VISUAL SLAM applications
- Based on visual odometry (inferred from extracted characteristics features thanks to a Bundle Adjustment algorithm for example)
- Based on visual beacon recognition
- Based on visual map matching with a local map built from LIDARs and cameras and precise 3D a priori map

These observations can be combined in a unique data fusion algorithm (an Extended Kalman Filter for Safran) but the first two cannot be used at the same time (as they come from the same basic information).

The theoretical models of these four hybridisations based on visual information are presented and the projections of their performance for the self-driving cars localisation are illustrated with simulations results.

2 PROBLEM DESCRIPTION AND MODELISATION

2.1 Inertial Navigation System

The Inertial Navigation Systems (INS) are commonly composed of three accelerometers and three gyrometers. These latter Inertial Measurement Units (IMUs) use the terrestrial rotation, the gravity and their internal physics to determine the useful parameter needed for the navigation. The self-sufficiency of this system makes it an interesting candidate for the autonomous car self-driving.

The independence of the device is however not enough to counteract the inherent errors that affect directly the IMUs. The accelerometers and gyrometers measurements are indeed not perfect due to errors that are generally modelled as:

- Biases that affect the accelerometers measurements by adding a term, b , to the true specific force, f_{s_v} , measured by the accelerometers :

$$f_s = f_{s_v} + b$$

- Drifts that affect the gyrometers measurements by adding a term, d , to the true angular velocity, $\frac{d\theta}{dt_v}$, measured by the gyrometers:

$$\frac{d\theta}{dt} = \frac{d\theta}{dt_v} + d$$

- Scale factors, S_{f_a} and S_{f_g} , that affect both the gyrometers and accelerometers measurements:

$$f_s = f_s * (1 + S_{f_a})$$

$$\frac{d\theta}{dt} = \frac{d\theta}{dt_v} * (1 + S_{f_g})$$

In order to take into account these errors the INS used for navigation must be headed by a so called alignment

phase. By using a Zero velocity UPdaTe (ZUPT) information, this preliminary step permits to estimate the attitude and heading as well as the intrinsic errors presented above. However, once the vehicle is moving the alignment phase ends. The integration of the IMUs increments, corrected of the errors estimated, does not permit to meet the navigation precision required for the self-driving localisation problem.

2.2 Hybridisation with camera using EKF

The camera boarded in the vehicle can collect interesting data coming from the user's direct environment. Image processing techniques, not addressed in this paper, make it indeed possible to extract from the consecutive images, points or lines whose positions and movements in the image can be used in complementarity with the INS (see 3 and 4). These same image processing techniques enable also to recognize beacons, whose positions are known, or routes from a map. The latter information can also be processed by a hybridisation algorithm (see 5 and 6).

The Extended Kalman Filter (EKF) is the Bayesian estimator preferred by Safran for both the alignment phase and the vision-aided IMU navigation. With the possibility to rely on years of knowledge coming from both the experience and the scientific community, this estimator permits to combine effectively the IMUs increments integration with plenty of other observation. This paper focuses on the clarification of the EKF observation equations by using the following EKF steps and notations:

Prediction:

$$\hat{X}_{(k+1|k)} = \Phi_k \cdot \hat{X}_{(k|k)} + G_k \cdot U_k$$

$$P_{k+1|k} = \Phi_k \cdot P_{k|k} \cdot \Phi_k^T + Q_k$$

Kalman Gain:

$$K_{k+1} = P_{k+1|k} \cdot H_{k+1}^T * (H_{k+1} \cdot P_{k+1|k} \cdot H_{k+1}^T + R_{k+1})^{-1}$$

Update:

$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1} * (Y_{k+1} - H_{k+1} \cdot \hat{X}_{k+1|k})$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1} * (H_{k+1} \cdot P_{k+1|k})$$

By considering:

- $\hat{X}_k$: The state vector
- Y_k : The observation vector
- Φ_k : The propagation matrix
- U_k : The command vector
- G_k : The command matrix
- H_k : The observation matrix
- W_k : The state noise vector (with $Q_k = \text{Cov}(W_k)$)
- V_k : The observation noise vector (with $R_k = \text{Cov}(V_k)$)

Notes:

The four hybridisations techniques presented assume the alignment phase done.

The integration of the increments coming from the IMUs is not detailed to focus on the vision contribution.

The camera is modelled as a pinhole.

2.3 Notations

Hereafter are the definitions of the frames used in this paper.

2.3.1 Earth frame [e]

[e] is an orthonormal coordinate system fixed within the earth. Its characteristics are:

- Origin at the centre of the WGS84 ellipsoid.
- Z_e directed towards the north celestial pole.
- X_e directed towards the intersection of the equatorial plane with the reference meridian.
- Y_e in the equatorial plane completing the direct frame.

2.3.2 Local geodetic frame [g]

[g] is an orthonormal coordinate system which is rigidly attached to and defined within the vehicle carrying the navigation system. Its characteristics are:

- Origin at the vehicle gravity centre.
- X_g directed towards geographical north and belongs to local horizontal plane.
- Y_g directed towards west and belongs to local horizontal plane.
- Z_g along the plane (X_g, Y_g) normal direction.

2.3.3 Navigation platform frame [v]

[v] is an orthonormal coordinate system which is rigidly attached to and defined within the vehicle carrying the navigation system. This frame is the one used for the navigation integration and has the following characteristics:

- Origin at the vehicle gravity centre.
- [v] is derived from [g] frame by a wander azimuth α rotation along Z_g axis.

2.3.4 Body frame [b]

[b] is an orthonormal coordinate system which is rigidly attached to and defined within the vehicle carrying the navigation system. Its characteristics are:

- Origin at the vehicle gravity centre.
- X_b is collinear with longitudinal axis of vehicle, positive towards the nose.
- Y_b is collinear with lateral axis of carrier, positive towards the right wing.
- Z_b axis completes the right-hand system.

2.3.5 INS measurement frame [m]

[m] is an orthonormal coordinate system attached to the INS. It corresponds to the frame in which the increments are calculated. Its characteristics are:

- Origin at the INS gravity centre.
- X_m is collinear with longitudinal axis of the device, positive towards the front.
- Y_m is with the lateral axis of the device, positive towards the right wing.
- Z_m completes the right-hand system.

2.3.6 Camera frame [c]

[c] is an orthonormal coordinate system attached to the camera. Its characteristics are:

- Origin at the Camera optical centre.
- X_c is collinear with longitudinal axis of the device, positive towards the front.
- Y_c is with the lateral axis of the device, positive towards the right wing.
- Z_c completes the right-hand system.

2.3.7 Pixel frame [uv]

[uv] is an orthonormal coordinate system attached to the camera. Its characteristics are:

- Origin at the image centre.
- U is collinear with Y_c .
- V is collinear with Z_c .

2.3.8 Vectors and Matrices

- In the whole paper the rotation matrices are noted with T such as T_{cb} represents the camera harmonisation matrix to transform a vector expressed in body frame [b] to a vector expressed in camera frame [c].
- Considering U a vector of three elements, $A(U)$ corresponds to the antisymmetric matrix, such

as, with $U = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix}$:

$$A(U) = \begin{pmatrix} 0 & U_2 & -U_3 \\ -U_2 & 0 & U_1 \\ U_3 & -U_1 & 0 \end{pmatrix}$$

- Considering V a vector, the whole paper notes this vector and components expressed in the camera frame [c] as:

$$V^{[c]} = \begin{pmatrix} V_x^{[c]} \\ V_y^{[c]} \\ V_z^{[c]} \end{pmatrix}$$

3 TIGHTLY-COUPLED HYBRIDISATION

INS-VISION

Inspired from the Simultaneous Localisation And Mapping (SLAM) algorithm, this hybridisation permits an increased calculation efficiency by considering a limited amount of data coming from the camera.

3.1 Theory

3.1.1 Motivation

The camera can provide information about the movements of the user surrounding objects through what can be called Feature Points (FP). These FPs are pixels of the camera image and correspond directly to points, in the terrestrial frame [e], projected in the pixel frame [uv]. These pixels are trackable in time thanks to the image processing algorithms. The processing of this information,

alone or with an INS can lead to a positioning estimation technique.

The SLAM algorithms are one of the methods that use the FPs to locate the user in a map. By processing a huge amount of information this well-known localization technique permits to estimate the user position while creating a map of the visited place. As mentioned in (Bailey, Nieto, Guivant, Stevens, & Nebot, 2006), although this can lead to an interesting positioning precision in certain case, the amount of data that is to be processed makes the algorithms hard to be implemented in real time.

The combination of the INS with the camera reduces the need of an important amount of FPs. By using the INS continuity advantage, the EKF presented in this paragraph is indeed behaving as a SLAM algorithm by taking advantage of only 15 camera FPs.

3.1.2 The inverse depth parameterisation

Once known the general principle of this hybridisation, one must decide the composition of the EKF state vector. The camera part of this vector depends on the camera information that is to be used. In the case of the tightly-coupled algorithm the state of the art provides plenty of possibilities. The most intuitive one, presented in (Mourikis & Roumeliotis, 2007), consists on filling the state vector directly with the position of the FPs in the terrestrial frame [e]. Another solution, presented in (Veth, 2009), suggest completing the state vector with the position of the FPs in the camera pixel frame [uv].

These two latter methods are not the one discussed in this paper whether because of a low stability of the solution or because of a complex EKF state prediction equation. Safran has preferred the 6 states parameterisation method consisting of the consideration of 6 states for each FPs considered:

- $p_{user_ini/k}^{[e]}$: (3x1) the initial user position, in the terrestrial frame. It is the user position at the first time of the FP view.
- $\theta_{user_ini/k}^{[e]}$: (2x1) the initial azimuth and elevation angles in the terrestrial frame. These angles permit to determine the direction of the vector with origin $p_{user_ini/k}^{[e]}$ and pointing at the FP.
- $\rho_{ini/k}$: The initial inverse of the distance between the user initial position and the FP.

This solution, called the inverse depth parameterisation, may not be intuitive. The inverse depth is in fact used in order for this estimated quantity to be Gaussian. Part IV of paper (Civera, Davidson, & Martinez Montiel, 2008) gives the mathematical theory lying behind this representation. It also provides mathematical tools to prove the bigger efficiency of this method compared to the other presented in (Mourikis & Roumeliotis, 2007).

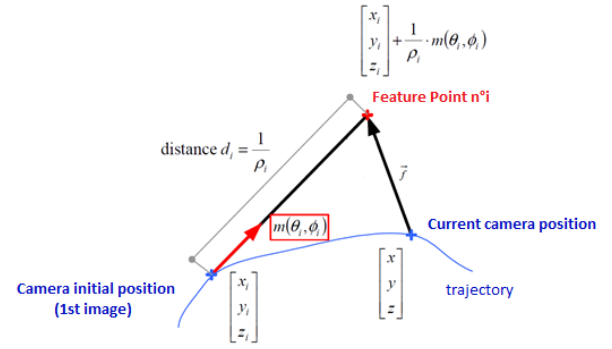


Figure 4: Tightly-Coupled hybridisation parameterisation illustration

3.1.3 EKF equations

Specific notations

For this part only, in order for the equations to be clear, the EKF hybridisation uses the state noise vector presented below, considering the states instead of the error states:

$$X_k = \begin{bmatrix} P_{user/k} \\ V_{user/k} \\ \theta_{user/k} \\ D_{gyr/k} \\ B_{acc/k} \\ S_{facc/k} \\ Camera_k \end{bmatrix}$$

With:

- $P_{user/k}$: (3x1) User position vector.
- $V_{user/k}$: (3x1) User velocity vector.
- $\theta_{user/k}$: (3x1) User attitude and heading vector.
- $D_{gyr/k}$: (3x1) Drifts vector for the gyrometers.
- $B_{acc/k}$: (3x1) Biases vector for the accelerometers.
- $S_{facc/k}$: (3x1) Accelerometer scale factor.
- $Camera_k$: (6x1) inverse depth parameterisation (see 3.1.2).

State equation

Considering FPs that are immobile in the scene, the inverse depth parameterisation selected for this hybridisation lies on states that are constant over time. This important hypothesis - that constraints the image processing in the choice of the FPs - makes the state transition matrix equal to identity for the FPs related states.

Observation equation

Assuming the notation presented in 2, the EKF observation equation can be considered as follow:

$$Y_k = h(X_k) = g_1(g_2(X_k))$$

With:

$$g_2(X_k) = h^{[c]} = T_{ce} * \left(p_{user_ini/k}^{[e]} - p_{user/k}^{[e]} + \frac{1}{\rho_{ini/k}} * m(\theta_{user_ini/k}^{[e]}) \right)$$

$$g_1(h^{[c]}) = \begin{pmatrix} -f_u * \frac{h_y^{[c]}}{h_x^{[c]}} \\ -f_v * \frac{h_z^{[c]}}{h_x^{[c]}} \end{pmatrix}$$

Such as:

- $m(\theta_{user_ini/k}^{[e]})$: (3x1) Directional vector between the FP and the initial user position in the terrestrial frame.
- $h^{[c]}$: (3x1) FP position expressed in the camera frame.

$$h^{[c]} = \begin{bmatrix} h_x^{[c]} \\ h_y^{[c]} \\ h_z^{[c]} \end{bmatrix}$$

- f_u, f_v : Camera focal lengths.
- T_{ce} : Transfer matrix.

This equation leads to the calculation of the observation matrix that comes from the Jacobian matrix calculation of $h(X)$. Thus:

$$H = \frac{\partial h}{\partial X} = \frac{\partial g_1}{\partial h^{[c]}} * \frac{\partial g_2}{\partial X}$$

Such as

$$\frac{\partial g_2}{\partial X} = (-T_{ce} \quad 0_{3 \times 3} \quad B \quad 0_{3 \times 3} \quad 0_{3 \times 3} \quad 0_{3 \times 2} \quad T_{ce} \quad C \quad D)$$

$$\frac{\partial g_1}{\partial A} = \begin{pmatrix} f_u * \frac{h_y^{[c]}}{h_x^{[c]2}} & -f_u * \frac{1}{h_x^{[c]}} & 0 \\ f_v * \frac{h_z^{[c]}}{h_x^{[c]2}} & 0 & -f_v * \frac{1}{h_x^{[c]}} \end{pmatrix}$$

With

$$B = \frac{\partial T_{ce} * \left(p_{user_ini/k}^{[e]} - p_{user/k}^{[e]} + \frac{1}{\rho_{ini/k}} * m(\theta_{user_ini/k}^{[e]}) \right)}{\partial \theta}$$

$$B = \frac{\partial T_{ce}}{\partial \theta} * \left(p_{user_ini/k}^{[e]} - p_{user/k}^{[e]} + \frac{1}{\rho_{ini/k}} * m(\theta_{user_ini/k}^{[e]}) \right)$$

$$C = \frac{\partial T_{ce} * \left(p_{user_ini/k}^{[e]} - p_{user/k}^{[e]} + \frac{1}{\rho_{ini/k}} * m(\theta_{user_ini/k}^{[e]}) \right)}{\partial \theta_{user_ini/k}}$$

$$C = T_{ce} * \frac{1}{\rho_{ini/k}} * \frac{\partial m(\theta_{user_ini/k}^{[e]})}{\partial \theta_{user_ini/k}}$$

$$D = \frac{\partial T_{ce} * \left(p_{user_ini/k}^{[e]} - p_{user/k}^{[e]} + \frac{1}{\rho_{ini/k}} * m(\theta_{user_ini/k}^{[e]}) \right)}{\partial \rho_{ini/k}}$$

$$D = T_{ce} * \frac{\partial 1/\rho_{ini/k}}{\partial \rho_{ini/k}} * m(\theta_{user_ini/k}^{[e]})$$

Re-initialization of the covariance matrix

One of the specificity of this hybridisation lies in the FPs states ephemerality. While moving, the vehicle may indeed consider each FPs for a short period of time before having to consider a new one. Consequently, the EKF shall adapt its estimations and especially the covariance matrix parts referring to the disappeared/appeared FPs at a possibly high frequency. This uninterrupted intervention in P leads inevitably to break the positive definite

characteristic of the covariance matrix necessary for a healthy behaviour of the estimator.

To counteract this point Safran has implemented the tight coupled hybridisation using a Kalman-Bierman filter. The latter filter basic principle lies into the propagation of a UDUt covariance factorization of the P covariance matrix. By using U and D, upper triangular matrix and diagonal matrix respectively, such as $P = U \cdot D \cdot U^T$, and by reinitialising the U and D decomposition of P at each disappearance of a FP, the filter is assured not to lose the positive definite characteristic of P.

3.2 Simulation results

3.2.1 Simulation tool

The EKF tight vision-aided IMU hybridisation has been implemented on Matlab. The simulator coded and owned by Safran permits to simulate:

- A typical Safran INS. This is done by simulating the accelerometers and gyrometers increments affected to the precisely modelled errors. The main characteristics are:
 - Accelerometer biases: 50 micro g
 - Accelerometer Scale Factor: 4 ppm
 - Gyrometer drift: 0.006 °/h
 - Gyrometer Scale Factor: 60 ppm
- A basic pinhole camera with the following characteristics:
 - Field of vision of 300 meters
 - Field of vision opening width of 30 degrees from the left to the right
 - Camera focal centre located at the gravity centre of the INS (no lever arms are modelled).
 - Camera noise is 1 pixel.
 - Direction of the camera is configurable. The camera can look straight forward, along the vehicle front, or on the wayside of the latter.
- Feature points that appear directly in the field of vision of the camera for use by the EKF.
- The EKF algorithm presented previously.

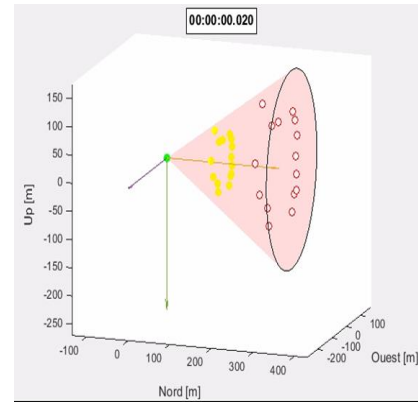


Figure 5: Pinhole camera, FPs and vehicle simulation illustration.

Figure 5 illustrates the basic environment of the camera that is simulated. The green point represents the true

position of the vehicle whose frame [b] attached to is represented with coloured arrows. In pink, the camera field of view is represented covering FPs in white and red. The yellow points are the FPs position estimated by the EKF.

3.2.2 Simulation context

Using a typical Safran INS and a pinhole camera presented before, two times 100 Monte Carlo campaigns have been launched for a vehicle navigating on a synthetic, city-like trajectory. The first campaign assumes a camera looking in the direction of the vehicle front (case Front) whereas the second assumes a camera pointing in the wayside of this vehicle (case Wayside). Table 1 summarises the way the FPs are simulated for both the Monte Carlo campaigns.

Parameters	Case	Value	Units
Initial distance between user and FP	Front	[100;300]	meters
	Wayside	[200;300]	meters
Initial Azimuth angle	Front	Anywhere	-
	Wayside	$[-Azimut_{max};0]$	degrees
Initial Elevation angle	Front	$[-5;5]$	degrees
	Wayside	$[-5;5]$	degrees

Table 1: Initial parameters for the FPs appearance

3.2.3 Results

The horizontal position accuracy is at best 6 meters RMS. The different configurations of the camera lead to two different estimation precisions. When the camera looks straight forward the EKF is less able to estimate the FP position than when it looks in the wayside. The inverse depth is indeed less observable in time by the EKF when the FPs moves in the image are not significant. Thus, as illustrate in Figure 6, the camera pointing on the wayside is more advantageous for this hybridisation than in the other case.

Parameters	Case	Value	Units
Position RMS	Front	20,42	meters
	Wayside	6,06	meters
Velocity RMS	Front	0,15	meters/second
	Wayside	0,07	meters/second
Heading RMS	Front	3,4	mrاد
	Wayside	1,88	mrاد
Pitch RMS	Front	0,21	mrاد
	Wayside	0,14	mrاد
Roll RMS	Front	0,18	mrاد
	Wayside	0,12	mrاد
FP Position RMS	Front	23,35	meters
	Wayside	1,4	meters

Table 2: Results of the Monte Carlo

3.2.4 Conclusion

This hybridisation simulated as presented is a priori not efficient enough to be implemented alone in the self-driving car. Further studies are being led in Safran to improve this hybridisation as well as to combine it with several others like the one in the following chapters of this article.

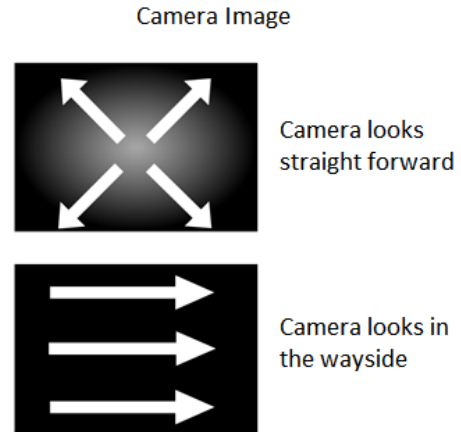


Figure 6: Illustration of the moves in the image of the FPs for the two cases studied

4 LOOSLY-COUPLED INS-VISION HYBRIDISATION

Another approach for fusing visual odometry information with localisation is to compute VSLAM on camera video streams and use the outputs (rotation and displacement) for data fusion. This is the “loosely coupled” version of hybridisation presented in chapter 3.

4.1 Theory

We suppose that Bundle Adjustment provides 2 measurements that are:

- the normalised translation vector named “u”, a unit vector supported by the translation vector T of the camera reference point between the initial and final instants projected in the camera frame, dates from the initial moment,
- the rotation matrix named R which transforms the reference frame of the data camera at initial moment to the reference frame of the data camera at final moment,
- Initial instant t_i and final instant t_f are supposed synchronous with Kalman filter tasks: $t_i = t_{k-1}$ and $t_f = t_k$. To ensure of that, it is necessary that the INS sends to the camera a signal to synchronise frame acquisition with Kalman filter tasks.

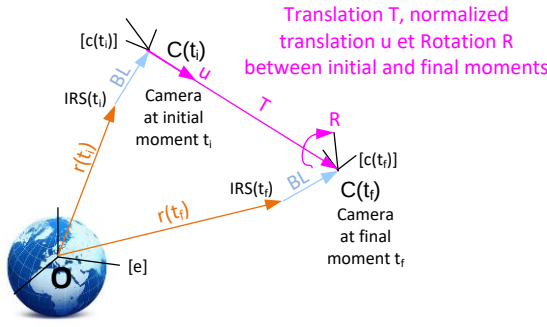


Figure 7: Loosely-Coupled hybridisation parameterisation illustration

Assuming the notations presented in 2, let us consider:

- BL lever arm from navigation point of IRS to C, reference point of the camera expressed in [b]. This lever arm is supposed to be constant and perfectly known
- T_{cb} is assumed to be constant
- T_{mb} is assumed to be constant and perfectly known
- $Id_{3 \times 3}$ Identity matrix 3x3

The camera reference point geocentric position at the initial instant t_i expressed in the reference [c] dated from the initial instant t_i is written:

$$PosECEF Cam(t_i)^{[c]t_i} = T_{cb} \cdot [T_{mb}^T \cdot [T_{me}(t_i) \cdot r^{[e]}(t_i)] + BL^{[b]}]$$

Similarly, the camera reference point geocentric position at the final instant t_f expressed in the reference [c] dated from the initial instant t_i is written:

$$PosECEF Cam(t_f)^{[c]t_i} = T_{cb} \cdot [T_{mb}^T \cdot [T_{me}(t_i) \cdot r^{[e]}(t_f)] + BL^{[b]}]$$

The observation model of translation T is obtained by difference between the true translation and the estimated translation by the inertial navigation:

$$\delta Y_T(t_f) = Translation^{[c]t_i}(t_i, t_f) - \hat{T} Translation^{[c]t_i}(t_i, t_f)$$

With following navigation errors:

- Position errors named δr and defined by $\delta r = \hat{T}_{ve} \cdot (r^{[e]} - \hat{r}^{[e]})$,
- Velocity errors named δv and defined by $\delta v = \hat{T}_{ve} \cdot (V^{[e]} - \hat{V}^{[e]})$,
- Attitude errors named θ and defined by $A(\hat{T}_{ev} \theta) = T_{em} \hat{T}_{em}^T - Id_{3 \times 3}$,

we can rewrite the model as follows:

$$\delta Y_T(t_f) = T_{cb} \cdot T_{mb}^T \cdot \hat{T}_{me}(t_i) \cdot \left\{ \begin{array}{l} [\hat{T}_{ev}(t_f) \cdot \delta r(t_f) - \hat{T}_{ev}(t_i) \cdot \delta r(t_i)] \\ -A[\hat{T}_{ev}(t_i) \cdot \theta(t_i)] \cdot [\hat{r}^{[e]}(t_f) - \hat{r}^{[e]}(t_i)] \end{array} \right\} + NL$$

The first term $T_{cb} \cdot T_{mb}^T \cdot \hat{T}_{me}(t_i)$ is the projection matrix in the initial camera frame.

The second term $[\hat{T}_{ev}(t_f) \cdot \delta r(t_f) - \hat{T}_{ev}(t_i) \cdot \delta r(t_i)]$ is the variation of position error.

The third term $A[\hat{T}_{ev}(t_i) \cdot \theta(t_i)] \cdot [\hat{r}^{[e]}(t_f) - \hat{r}^{[e]}(t_i)]$ is the product Translation by Variation of attitude error. NL represents a nonlinear term.

To link the errors of the instant t_i (supposed equal to previous Kalman filter task t_{k-1}) to the current instant t_f (supposed equal to actual Kalman filter task t_k), an augmented state is introduced:

$$\begin{bmatrix} \delta X(t_k) \\ \delta r(t_{k-1}) \\ \theta(t_{k-1}) \end{bmatrix} = \begin{bmatrix} \Phi(t_k, t_{k-1}) & 0 & 0 \\ T_r & 0 & 0 \\ T_\theta & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta X(t_{k-1}) \\ \delta r(t_{k-2}) \\ \theta(t_{k-2}) \end{bmatrix} + \begin{bmatrix} v(t_{k-1}) \\ 0 \\ 0 \end{bmatrix}$$

with following notations:

- $\Phi(t_k, t_{k-1})$ is transition matrix defined by $\delta X(t_k) = \Phi(t_k, t_{k-1}) \cdot \delta X(t_{k-1})$
- T_r makes it possible to extract the position error states δr from δX state vector: $\delta r(t_k) = T_r \cdot \delta X(t_k)$

For example, if δr states are placed at the beginning of the state vector, T_r is defined by

$$T_r = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix}$$

- T_θ makes it possible to extract the position error states θ from δX state vector: $\delta r(t_k) = T_r \cdot \delta X(t_k)$

For example, if θ states are placed after δr and δv errors, T_θ is defined by

$$T_\theta = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 \end{bmatrix}$$

The observation model of translation T can be rewritten with this augmented state;

$$\begin{aligned} \delta Y_T(t_f) = & H_T(t_k, \delta X(t_k)) \cdot T_r \cdot \delta X(t_k) \\ & + H_T(t_k, \delta r(t_{k-1})) \cdot \delta r(t_{k-1}) \\ & + H_T(t_k, \theta(t_{k-1})) \cdot \theta(t_{k-1}) + NL \end{aligned}$$

$$H_T(t_k, \delta X(t_k)) = T_{cb} \cdot T_{mb}^T \cdot \hat{T}_{me}(t_{k-1}) \cdot \hat{T}_{ev}(t_k) \cdot T_r$$

$$H_T(t_k, \delta r(t_{k-1})) = -T_{cb} \cdot T_{mb}^T \cdot \hat{T}_{me}(t_{k-1}) \cdot \hat{T}_{ev}(t_{k-1})$$

$$H_T(t_k, \theta(t_{k-1})) = T_{cb} \cdot T_{mb}^T \cdot \hat{T}_{me}(t_{k-1}) \cdot A[\hat{r}^{[e]}(t_k) - \hat{r}^{[e]}(t_{k-1})] \cdot \hat{T}_{ev}(t_{k-1})$$

The observation model of rotation R is obtained by written the micro-rotator between the true rotation and estimated rotation by inertial navigation:

$$T_{c_{t_f}c_{t_i}} = \exp_{SO3} \left(\text{Rotation}^{[c]_{t_i}}(t_i, t_f) \right) = T_{cb} \cdot T_{mb}^T \cdot T_{me}(t_f) \cdot T_{me}^T(t_i) \cdot T_{mb} \cdot T_{cb}^T$$

$$\hat{T}_{c_{t_f}c_{t_i}} = \exp_{SO3} \left(\hat{\text{Rotation}}^{[c]_{t_i}}(t_i, t_f) \right) = T_{cb} \cdot T_{mb}^T \cdot \hat{T}_{me}(t_f) \cdot \hat{T}_{me}^T(t_i) \cdot T_{mb} \cdot T_{cb}^T$$

$$\delta Y_R(t_f) = T_{c_{t_f}c_{t_i}} \cdot \hat{T}_{c_{t_f}c_{t_i}}^T - Id_{3 \times 3}$$

$$\begin{aligned} \delta Y_R(t_f) = & -T_{cb} \cdot T_{mb}^T \cdot \hat{T}_{me}(t_f) \\ & \cdot A \left[\hat{T}_{ev}(t_f) \cdot \theta(t_f) - \hat{T}_{ev}(t_i) \cdot \theta(t_i) \right] \\ & \cdot \hat{T}_{me}^T(t_f) \cdot T_{mb} \cdot T_{cb}^T + NL_R \end{aligned}$$

The term $\hat{T}_{ev}(t_f) \cdot \theta(t_f) - \hat{T}_{ev}(t_i) \cdot \theta(t_i)$ represents the variation of attitude error. NL_R represents a nonlinear term.

The observation model of rotation R can be rewritten with this augmented state;

$$\delta Y_R(t_f) = H_R(t_k, \delta X(t_k)) \cdot \delta X(t_k) + H_R(t_k, \theta(t_{k-1})) \cdot \theta(t_{k-1}) + NL_R$$

$$H_R(t_k, \delta X(t_k)) = T_{cb} \cdot T_{mb}^T \cdot \hat{T}_{me}(t_k) \cdot \hat{T}_{ev}(t_k) \cdot T_\theta$$

$$H_R(t_k, \theta(t_{k-1})) = -T_{cb} \cdot T_{mb}^T \cdot \hat{T}_{me}(t_k) \cdot \hat{T}_{ev}(t_{k-1})$$

The true normalized translation is written as follows:

$$u = \frac{\text{Translation}^{[c]_{t_i}}(t_i, t_f)}{\|\text{Translation}^{[c]_{t_i}}(t_i, t_f)\|} = \frac{T}{\|T\|}$$

Moreover, the differential of u is:

$$\delta \hat{u} = \frac{\delta \hat{T} \cdot \|\hat{T}\| - \hat{T} \cdot \delta \|\hat{T}\|}{\|\hat{T}\|^2} = \frac{\delta \hat{T}}{\|\hat{T}\|} - \frac{\hat{T} \cdot \delta \|\hat{T}\|}{\|\hat{T}\|^2}$$

We can write the innovation of normalised translation as follows:

$$\delta Y_u(t_f) = u - \hat{u} = H_u(t_f) \cdot \delta X(t_f) + NL_u$$

$$H_u(t_f) = \frac{1}{\|\hat{T}\|} \cdot [H_T(t_f) - \hat{u} \cdot \hat{u}^T \cdot H_T(t_f)] = \frac{1}{\|\hat{T}\|} \cdot [Id_{3 \times 3} - \hat{u} \cdot \hat{u}^T] \cdot H_T(t_f)$$

$$\begin{aligned} \delta Y_u(t_f) = & \frac{1}{\|\hat{T}\|} \cdot [Id_{3 \times 3} - \hat{u} \cdot \hat{u}^T] \\ & \cdot \{H_R(t_k, \delta X(t_k)) \cdot \delta X(t_k) + H_R(t_k, \theta(t_{k-1})) \cdot \theta(t_{k-1})\} + NL_u \end{aligned}$$

4.2 Simulation results

This hybridisation has been tested on the Kitti dataset and shows enhanced results on the estimated position.

As illustrated on Figure 8, the position error at the end of the simulation is 20 m on a 320 m track, which is 50% better than the pure inertia localisation.

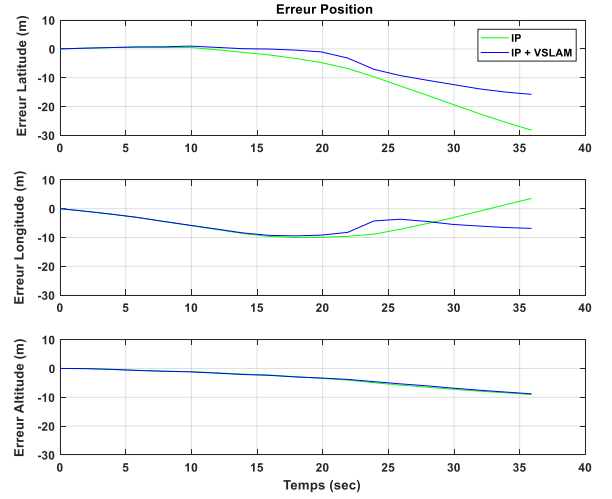


Figure 8: Results of loosely-coupled INS-VISION hybridisation on Kitti dataset

5 VISUAL BEACON HYBRIDISATION

On autonomous vehicle, vision sensors (camera, Lidar) coupled with high accuracy cartography could be used as a very precise absolute positioning. In this chapter, we experiment using mapped visual beacon, such as road signs, for navigation aiding. Beacons are considered punctual, this does not rely on continuous structure such as lane markings.

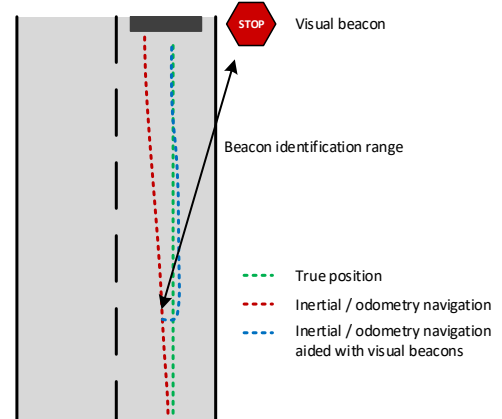


Figure 9: Mitigation of dead-reckoning drift with visual beacon aiding

The overall process can be decomposed in three steps:

- 1. Detection:** vision algorithms - based on neural networks - can recognise a specific beacon through a video stream. Its direction in the camera frame is characterised by two angles, obtained with pixel coordinates.
- 2. Identification:** The detected beacon is compared with the local map. If a beacon is found in the direction of detection, association is made to get the beacon geographical coordinates.
- 3. Data fusion:** Beacon direction in camera frame (identification) and absolute position (identification) are sent to the navigation system for data fusion.

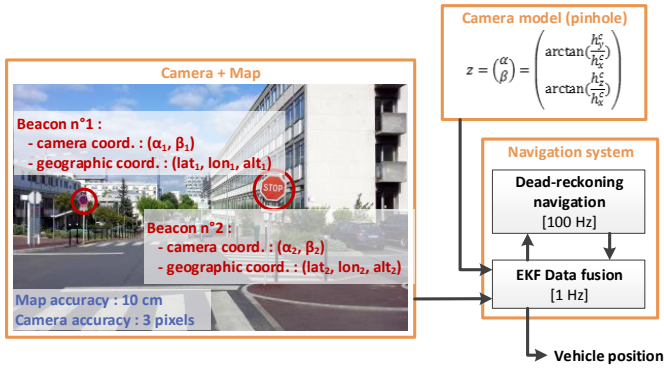


Figure 10: Data fusion diagram (EKF)

5.1 Kalman Model

5.1.1 State vector and measurement matrix

The information of beacon detection and identification is fused with dead-reckoning navigation through an Extended Kalman Filter (EKF).

The EKF state vector contains error vectors of position and attitude (in [g] frame):

$$X = \begin{bmatrix} \delta p_x \\ \delta p_y \\ \delta p_z \\ \delta \varphi_x \\ \delta \varphi_y \\ \delta \varphi_z \end{bmatrix} \quad (1)$$

The camera sensor is modeled with the pinhole equation:

$$z = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \arctan(\frac{h_y^c}{h_x^c}) \\ \arctan(\frac{h_z^c}{h_x^c}) \end{pmatrix} \quad (2)$$

With the following notations:

- $h^c = \begin{pmatrix} h_x^c \\ h_y^c \\ h_z^c \end{pmatrix} = T_{c/g} \cdot h^g$: vector between beacon and camera in the camera frame
- h^g : vector between beacon and camera in the geographic frame
- $T_{c/g}$: transition matrix between geographic frame [g] and camera frame [c]

For EKF implementation, equation (2) has to be derivated. The measurement matrix H is obtained with the calculation of the Jacobian:

$$H = \frac{\partial z}{\partial x} = \frac{\partial z}{\partial h^c} \cdot \frac{\partial h^c}{\partial x} = \frac{\partial z}{\partial h^c} \cdot [H_p \quad H_\varphi] \quad (3)$$

Terms $\frac{\partial z}{\partial h^c}$, H_p and H_φ are given by the following equations:

$$\frac{\partial z}{\partial h^c} = \begin{bmatrix} -\frac{h_y^c}{h_x^c + h_y^c} & \frac{h_x^c}{h_x^c + h_y^c} & 0 \\ -\frac{h_z^c}{h_x^c + h_z^c} & 0 & \frac{h_x^c}{h_x^c + h_z^c} \end{bmatrix} \quad (4)$$

$$H_p = \frac{\partial h^c}{\partial x_p} = -T_{c/g} \quad (5)$$

$$H_\varphi = \frac{\partial h^c}{\partial x_\varphi} = T_{c/g} \cdot \begin{bmatrix} 0 & h_x^g & -h_y^g \\ -h_x^g & 0 & h_z^g \\ h_y^g & -h_z^g & 0 \end{bmatrix} \quad (6)$$

5.1.2 Management of nonlinearities

The linearisation approximation is a strong hypothesis that is not satisfied for every geometric configuration. In order to mitigate the effect of nonlinearities, an iterative observation has been implemented (Mirzaei & Roumeliotis, 2008).

Indeed, the iterated EKF (IEKF) enables to improve the Kalman Filter update, as illustrated in Figure 11: the estimated position progressively moves towards the true line of sight and the covariance is getting error

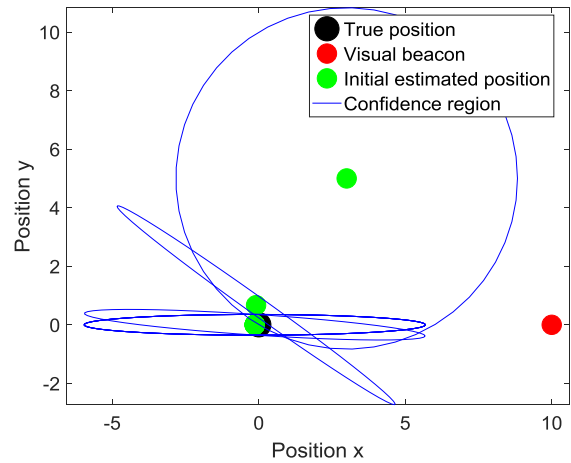


Figure 11: Evolution of the initial estimated position with the IEKF

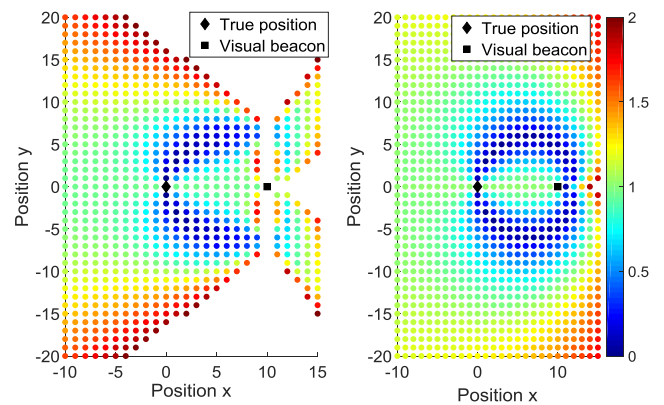


Figure 12 : Distribution of the ratio r for 1 (left) and 4 (right) iterations of the IEKF ($r > 2$ is not displayed)

Figure 12, in which the ratio $r = \frac{error}{std}$ after Kalman update is displayed with a colorbar. As expected, the region of validity widens and gets more accurate as more iterations are considered.

Nonetheless, on some degenerate cases in which camera line of sight direction is erroneous (see Figure 13), the iterative process can bring divergence. To address this problem, a condition can be added to execute the data fusion, based on the estimated beacon coordinates in camera frame:

$$\hat{\alpha} < \alpha_{thres.} \text{ AND } \hat{\beta} < \beta_{thres.} \quad (7)$$

Terms $\hat{\alpha}$ and $\hat{\beta}$ are, respectively, Kalman estimates of α and β , which depend on position and attitude estimates (see Figure 13). In this study, there is no analytical formulation proposal for thresholds $\alpha_{thres.}$ and $\beta_{thres.}$, they are set empirically. In

Figure 14, we show the evolution of the region of validity (and thus, $\alpha_{thres.}$ in Table 1) for different values of r , considering 4 IEKF iterations.

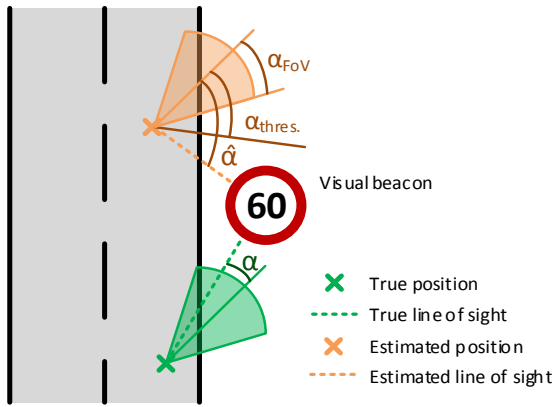


Figure 13: Illustration of a geometric configuration with nonlinearities

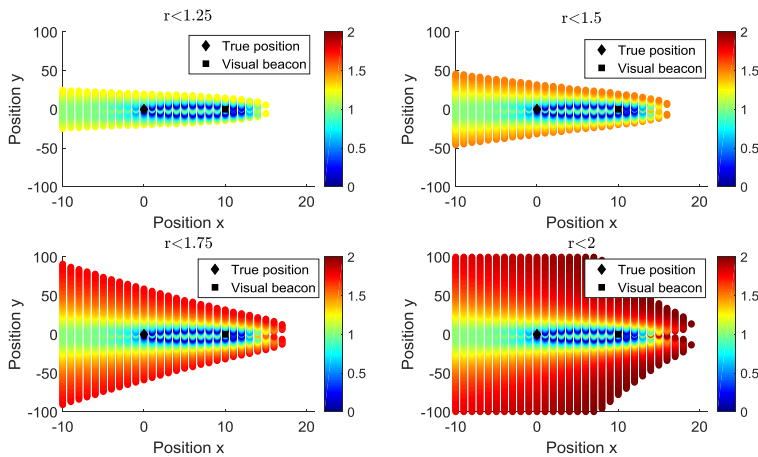


Figure 14: Evolution of the region of validity

Table 3: Evolution of the threshold angle

$r_{max.}$	1	1.25	1.5	1.75	2
$\alpha_{thres.}$ (degrees)	14.7	51.3	66.5	77.6	88.3

5.1.3 Measurement errors

The measurement error contributors can be classified into two categories:

- If error is Gaussian, its amplitude is modelled through the Observation noise covariance matrix R . This can be further classified:

- Homogeneous to angle errors in the camera frame

$$R = \begin{bmatrix} \sigma(\epsilon_\alpha)^2 & 0 \\ 0 & \sigma(\epsilon_\beta)^2 \end{bmatrix}$$

- Homogeneous to position errors in the geographic frame

$$R = K \cdot \begin{bmatrix} \sigma(\epsilon_x)^2 & 0 \\ 0 & \sigma(\epsilon_y)^2 \end{bmatrix} \cdot K^T$$

$$\text{with } K = \frac{\partial z}{\partial h^c} \cdot T_{c/g}$$

- If error is constant, there are two configurations:
 - Its amplitude is negligible toward other contributors. This does not need to be modelled
 - Its amplitude is not negligible toward other contributors. This must be modelled through a new state in X . In the measurement matrix H the new terms are inspired from $\delta p_x, \delta p_y, \delta p_z$ for position errors and from $\delta \varphi_x, \delta \varphi_y, \delta \varphi_z$ for angle errors.

Table 4: Measurement error contributors

Error contributor	Angle	Position
Gaussian	Camera measure	-
Constant	Harmonisation	Lever arm Map bias Datation ^(*)

(*) position error is function of the vehicle speed

5.2 Simulation results

This hybridisation has been tested on Monte-carlo simulations on a simple use-case: the vehicle is moving on a straight road with a road sign on its right. Observability is provided by the line of sight evolution in

geographic frame (see

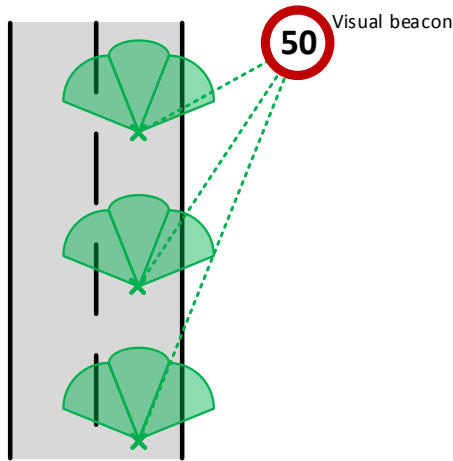


Figure 15).

Note: on these simulations, road sign is no longer detected after the vehicle have overtaken it.

Table 5: Parameters

Parameters	Value	Unit	Distribution
Number of simulations (Monte-carlo)	100	-	constant
Camera noise	1	mrاد	gaussian
Camera Horizontal FoV	45	°	constant
Camera Vertical FoV	45	°	constant
Camera acquisition rate	10	Hz	constant
Number of cameras	3	-	constant
Map bias (X, Y & Z)	0.1	m	gaussian
Initial position error (X, Y & Z)	1	m	gaussian
Initial azimuth error	3	mrاد	gaussian
Road sign distance to the road	2	m	constant

As illustrated in Table 5, errors between camera and IMU – such as lever arm, harmonization and datation - are neglected in this simulation. In future works (e.g real time trials), these terms must be considered by adding new states or new terms in the measurement matrix (see §5.1.3).

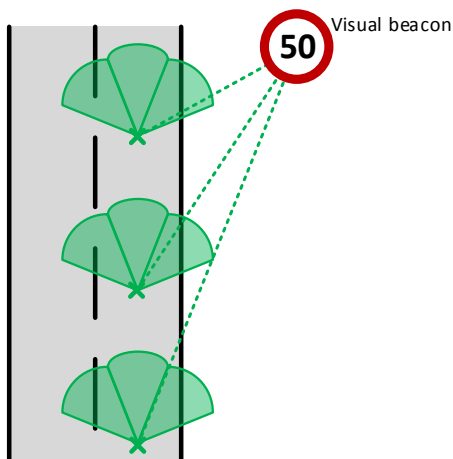


Figure 15: One beacon simulation with moving vehicle

The horizontal position accuracy is 16 cm error RMS, considering a $10 \cdot \sqrt{2}$ cm map bias (see Table 6, Figure 15). Thanks to this new observation, we are reaching the map accuracy on this simulation dataset.

Table 6: Hybridization accuracy

RMS position error	Unit	Longi.	Transv.	XY
Initial	[m]	1.0		1.4
End of trajectory	[m]	0.12	0.10	0.16

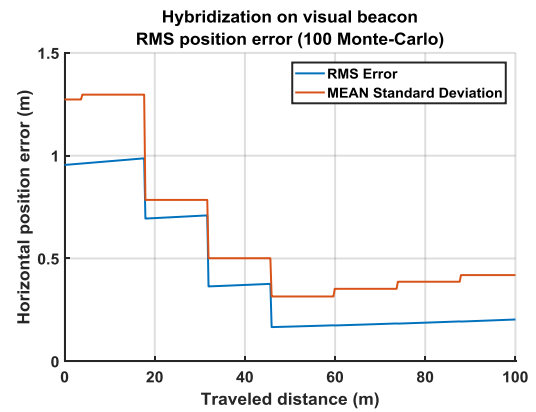


Figure 16: One beacon simulation, simulation results

As Kalman filter parameters have been set for robustness, this lead to a conservative covariance when observation is made many times in a row (see Table 7).

Table 7: Kalman filter covariance representativeness at end of trajectory

Covariance representativeness	Unit	Gaussian	Simulation
error < 1 x std	[%]	68	99
error < 2 x std	[%]	95	100
error < 3 x std	[%]	99.7	100

6 MAP MATCHING HYBRIDISATION

Map-matching localisation belongs to the absolute positioning algorithm family alongside with the visual-beacon localisation techniques. Unlike the latter, map-matching positioning methods mainly use continuous structures such as lanes, walls, road markings for instance and apply the following scheme: knowing a prior map of such structures and the predicted position and heading of the carrier – eg. a car or a plane–, one can predict a visual information and compare it to the actual visual sensors' stream. The resulting correlation is a rotation-translation pair $(R, \vec{t})$ that can be used as a loose observation within an Extended Kalman Filter (EKF).

Expected accuracies directly depend on the map precision, but are often at the decimeter-level.

6.1 Theory

The following parts describe what is encoded in the simulated maps and how. Then, the algorithm used is presented: it relies on a ICP technique called TRimmed Iterative Closest-Point (TRiCP). The associated covariance and observation matrix must be computed to be embedded in an EKF framework.

6.1.1 Map

The map used here are synthetics for simulation purposes.

The paper *Generation of High Precision Digital Maps using Circular Arc Splines* presented by (Andreas Schindler, 05 July 2012) explains a method to encode lane markings and most road markings with smoothed lines made of 2D line segments and circular lines called SMAPs, resulting in a 2D maps. The same geometric elements are generated and used here: they enable fast projections while reducing the map size. Since they are 2D maps – in UTM format –, they only enable $\begin{pmatrix} east \\ north \\ heading \end{pmatrix}$ computation which is enough for most applications.

6.1.2 TRiCP

Described in the (Besl & McKay, Feb. 1992) paper, the Iterative Closest Point algorithm (ICP) aims at computing the optimal transformation between a point cloud P and an entity M in the least-square sense in an iterative way: For each iteration, the steps are as follow:

- 1) Computation of the closest points of P , from M , resulting in Q .
- 2) $(R, \vec{t})$ pair computation for (P, Q)
- 3) $(R, \vec{t})$ application to P , resulting in a new P

In our case, the initial point cloud P is created from the visual sensors, processed and projected onto the map based on the state prediction. The 1) step is carried out with simple 2D projections on line segments and circular arcs. The 2) step formula is the “Unit Quaternion” (Horn, 1986) (Schindler, 15 October 2013), the latter being the main source of inspiration of this work. Another method of resolution exists nonetheless (S.Umeyama, 1991), both are inferred from the point-to-point cost function $J_{pt-pt}^{(i)}$ for the i -step:

$$J_{pt-pt}^{(i)} = \sum_{j=1}^n \|p_j^{(i)} - q_j^{(i)}\|_2^2$$

. The 3) step applies the following roto-translation T formula – for i step:

$$T_{\phi_i, s_i, \vec{t}_i} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$x \rightarrow s_i \begin{pmatrix} \cos(\phi_i) & -\sin(\phi_i) \\ \sin(\phi_i) & \cos(\phi_i) \end{pmatrix} x + \vec{t}_i$$

with s_i a scale factor, not used in this paper.

Given a threshold or a total number of iteration, the last step enables the computation of the total transformation parameters ϕ_{tot} and $\vec{t}_{tot}$ which can be used as innovation information by a filter.

To deal with outliers – always present in image processing techniques – the actual variant of ICP used is a tailor-made TRiCP (Chetverikov, Svirkov, Stepanov, & Krsek, 10 December 2002). It basically discards points too far from their respective counterparts at each step but reintegrates them at the beginning of each step though, ensuring the monotonic convergence of the algorithm.

One of the most problematic aspects of the ICP method is its risk of falling in local minima – not global ones. Given the lane and road markings patterns, this can happen in case of large initial heading error. A double consecutive TRiCP algorithm with a small and a larger radius is applied to reduce the risk caused by overlays.

6.1.3 ICP covariance and observation matrix

One of the main contributions of this section is the ICP covariance embedded within an EKF. Indeed, a particle filter is often preferred in map-matching methods because the covariance computation results from the particle distribution, at the cost of a heavier CPU usage though.

The paper of (Censi, 21 May 2007) provides an approached formula of $\text{cov}(Y_{ICP})$:

$$\text{cov}(Y_{ICP}) \sim \left(\frac{\partial^2 J}{\partial^2 Y_{ICP}} \right)^{-1} \frac{\partial^2 J}{\partial P \partial Y_{ICP}} \text{cov}(P) \frac{\partial^2 J}{\partial P \partial Y_{ICP}}^T \left(\frac{\partial^2 J}{\partial^2 Y_{ICP}} \right)^{-1}$$

with J the chosen cost function

However, here, the point-to-plane cost function is preferred in order to detect unobservable directions – straight or lightly curved roads– among others (Silvère Bonnabel, 16 Oct 2014):

$$J_{pt-plane}^{(i)} = \sum_{j=1}^n \left((p_j^{(i+1)} - q_j^{(i)}) | \vec{n}_j \right)^2$$

with $\langle . | . \rangle$ the scalar product operator

Also, the choice of the visual sensor’s point cloud – P – covariance $\text{cov}(P)$ is not diagonal since it goes to 0 as $\text{card}(P)$ increases. The “resolution error” (Martin Barczyk, October 2014) is used instead. Each point is associated with a road curve resulting in packets of point for every distinctive curve: within each packet, the points are correlated between each other.

Computing $\text{cov}(P)$ still requires the knowledge of the base standard deviation σ_p : it is larger to the sensors’ accuracy because the TRiCP’s result is less accurate due to TRiCP’s convergence thresholds. For conservative and dynamic purposes, it has been decided to choose $3\sigma_p$ as equal to the largest distance between the two point clouds resulting from the ICP algorithm:

$$\sigma_p = \frac{\text{argmax}_j \left(\|p_j^{(final)} - q_j^{(final)}\|_2 \right)}{3}$$

Such approximation works well with large point clouds P and small final errors.

The resulting covariance $\text{cov}(Y_{ICP})$ can be broken into 2 parts: a 2x2 matrix $\text{cov}(Y_{ICP(x,y)})$ corresponding to the

position accuracy $\begin{pmatrix} east \\ north \end{pmatrix}$ and a scalar $cov(Y_{ICP(\phi)})$ relative to the heading precision:

$$cov(Y_{ICP}) = \begin{pmatrix} cov(Y_{ICP(x,y)}) & 0 \\ 0 & 0 \end{pmatrix} \quad cov(Y_{ICP(\phi)})$$

$cov(Y_{ICP})$'s largest eigenvalue λ_{long} corresponds to the unobservable direction's eigenvector also called longitudinal vector ; the next largest eigenvalue λ_{lat} corresponds to the lateral vector and the smallest eigenvalue is easily associated with the heading uncertainty λ_{ϕ} . To deal with unobservable directions –in the road longitudinal axe –, the TRiCP observation vector is given in the road reference T_{road} giving the observation vector $\begin{pmatrix} longitudinal \\ lateral \\ orientation \end{pmatrix}$ – heading unchanged. The observation matrix H_{ICP} becomes T_{road} and $cov(Y_{ICP})$ becomes R_{VMM} :

$$R_{VMM} = \begin{pmatrix} \lambda_{long} & 0 & 0 \\ 0 & \lambda_{lat} & 0 \\ 0 & 0 & \lambda_{\phi} \end{pmatrix}$$

In case an unobservable direction is encountered the longitudinal observation is dropped. However, during the Kalman gain's computation, some residues in the unobservable direction can persist due to the state cross correlations $cov(east, north)$, $cov(east, \phi)$, $cov(north, \phi)$ and very low λ_{ϕ} notably. A directional Kalman-Schmidt is then applied to get rid of the Kalman gains aligned with the unobservable direction.

6.2 Simulation results

6.2.1 2D simulation

The simulations have been carried out in 2D so far –3D simulations are being implemented. The carrier has 2 accelerometers and 1 gyrometer of middle-range performance – compatible with Safran's Epsilon 10 INS datasheet. It is supposed the carrier has a visual system and an image/cloud processing algorithm outputting a noisy 2D-point cloud around the vehicle representing the lane and road markings extracted centre points. It is also supposed multiple consecutives 2D point clouds can be stored and kept coherent so that they only suffer from an error of translation and orientation related to the predicted state. Such large point clouds are only used to reduce or avoid unobservable direction periods: they require more time to process though.

6.2.2 Synthetics maps and increments

Maps are only composed of geometrical elements such as line segments and circular arcs: they encode the road themselves, some trajectories and most importantly the lane and road markings.

The trajectories are basically long smooths continuous lines made of line segments and circular arcs.

Accelerometers and gyrometers true increments are generated from these given a velocity trapezoidal profile.

Two maps have been generated:

- A short countryside track with a 2 lane-1 lane pattern whose ends are cut.
- A road network mimicking a small town, with many intersections

Each map has been “deteriorated” with multiple outliers taking the form of false lane/road markings.

Three trajectories are used:

- A countryside trajectory making a U-path following the associated road (Countryside)
- A town trajectory with few turns (Town A)
- A town “round” trajectory (Town B)

They last a couple of minutes each depending on the velocities used – 30km/h and 130km/h. The visual system range used is 50 meters and has a field of view of 360 degrees.

No map errors have been considered, but they are often inferior to the usual navigation errors – 5 to 10 cm.

6.2.3 Results

Being part of a larger sensitivity campaign, a batch of 30 Monte Carlo trials has been run for each trajectory and each velocity.

	Countryside	Town A	Town B
30 km/h	1.8m 0.8m 0.23°	1.2m 0.25m 0.12°	0.90m 0.20m 0.06°
130 km/h	1.0m 0.55m 0.25°	0.42m 0.18m 0.03°	0.80m 0.15m 0.06°

Table 8 Max errors (Longitudinal, Lateral, Heading)

	Countryside	Town A	Town B
30 km/h	0.40m 0.10m 0.04°	0.10m 0.03m 0.01°	0.18m 0.03m 0.01°
130 km/h	0.26m 0.10m 0.04°	0.03m 0.02m 0.01°	0.13m 0.03m 0.01°

Table 9 RMS errors (Longitudinal, Lateral, Heading)

The bulk of the errors occur when the accelerometers' bias and gyrometer's drift are not estimated yet, especially in a straight road at low speed. This leads to large errors at the beginning of trajectories or in long straight roads.

Previous static Monte Carlos trials have shown that in the map used, errors up to 1.5 meters don't lead to local minima. Only one scenario exceeds the limit, this occurs in the countryside simulation at low speed in a long lightly curvy road.

In these simulations, low speed has three disadvantages:

- Longer time spent in unobservable direction road portions, resulting in larger errors
- Longer periods in areas full of outliers, resulting in riskier situations

- Less acceleration to estimate bias, resulting in longer larger errors.

The town simulations show better accuracies thanks to the presence of intersections, enabling the fast estimation of bias and reducing the periods of unobservable directions.

6.3 Conclusion

The results obtained in these 2D simulations are almost compatible to the autonomous vehicle need in term of accuracy. However, there is a strong correlation between the reachable accuracy and the road pattern: long straight or lightly curved road are harmful in map-matching localisation, which could lead to larger errors or even divergences. Nonetheless, the hybridisation is light here: there is no odometer; no Visual SLAM etc. meaning unobservable directions should have less impact in a larger sensors fusion framework.

3D simulations are run to confirm the results obtained and an altitude measurement from the map is added. Next experiments aim at replaying recorded data to verify if the hypothesis formulated in simulations are correct –outlier rejecting algorithm, impact of obstacles, point cloud error interpretation, road precision among others.

7 GENERAL FUSION ALGORITHM

The several hybridisations presented in this paper can be used one by one or can be fused together in order to achieve the best theoretically estimation of the localisation of the bearer.

First of all, as mentioned in the introduction, two of them are not mixable: the loosely- and the tightly-coupled hybridisations cannot be combined as they are using the same basic information: feature points' position in the image.

Second of all, loosely-coupled is the only hybridisation giving no information on the environment. It is similar to a velocity observation (with an unknown scale factor though).

Third of all, the observations directly linked to information based on the environment (feature points, visual beacons, recognisable elements of a map) can be fused together in a fused map that can be matched with the a priori map as in the map matching hybridisation presented previously.

Last but not least, once the different kind of observations are selected, the corresponding hybridisations can be fused in a single localisation filter (i.e. with only one state vector) or they can be used sequentially, in different filters with different state vectors. For example, the visual beacons and map matching hybridisations can be performed on one state vector, including representation of the environment, before consolidated information built in this filter is sent to the main localisation filter estimating the localisation of the bearer thanks to the additional observation in a loosely-coupled scheme.

Although theory indicates a single filter with a full state vector should perform better than any other scheme, it is not clear it is the best solution. Being outside the theoretical background of the Kalman Filter (important non-linearities, non-gaussian noises ...) means no theory can assure the single filter will perform better. Furthermore, from an industrial point of view, it can also be easier to design and validate several filters instead of a complete single one.

The next step for Safran is to investigate on this topic in order to find the main design rules for Vision-aided localisation systems for its different applicative domains (earth, land, sea and space).

8 CONCLUSION

This article presented several hybridisations based on visual information for the localization of the autonomous cars (or any other bearer equipped with similar visual sensors).

The results on simulated data are satisfactory enough to imagine a fusion algorithm based on these different hybridisations that would be able to reach the performance needed for the localisation of the self-driving car.

While experiments on real data still need to be performed more intensively, the first results are really positive and Safran will keep on improving the hybridisations one by one and on how to combine them thanks to more and more data representative of different datasets brought by SafranTech Autonomous Vehicle Laboratory and by its partner Valeo.

9 REFERENCES

- Andreas Schindler, G. M. (05 July 2012). Generation of High Precision Digital Maps using Circular Arc Splines. *Intelligent Vehicles Symposium (IV), 2012 IEEE*. Alcalá de Henares, Spain: IEEE.
- Bailey, T., Nieto, J., Guivant, J., Stevens, M., & Nebot, E. (2006). Consistency of the EKF-SLAM Algorithm. *IEEE International Conference on Intelligent Robots and Systems*. Beijing, China.
- Besl, P., & McKay, N. D. (Feb. 1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 14, Issue: 2)*, 239 - 256.
- Censi, A. (21 May 2007). An accurate closed-form estimate of ICP'S. *Robotics and Automation, 2007 IEEE International Conference*. Roma, Italy: IEEE.
- Chetverikov, D., Svirko, D., Stepanov, D., & Krsek, P. (10 December 2002). The Trimmed Iterative Closest Point Algorithm. *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. 2.1047997: IEEE.
- Civera, J., Davidson, A. J., & Martinez Montiel, J. M. (2008). Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on robotics*.

- Horn, B. (1986, November 25). Closed-form solution of absolute orientation using unit Quaternion. *Journal of the Optical Society of America*.
- Martin Barczyk, S. B. (October 2014). Observability, Covariance and Uncertainty of ICP. *American Control Conference*. arXiv.
- Mirzaei, F., & Roumeliotis, S. (2008). A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE transactions on robotics*, 1143-1156.
- Mourikis, A. I., & Roumeliotis, S. I. (2007). A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. *IEEE International Conference on Robotics and Automation* . Rome, Italie.
- S.Umeyama. (1991, April). Least-square transformations between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 376 - 380.
- Schindler, A. (15 October 2013). Vehicle Self-localization with High-Precision Digital Maps. *Intelligent Vehicles Symposium (IV), 2013 IEEE*. Gold Coast, QLD, Australia: IEEE.
- Silvère Bonnabel, M. B. (16 Oct 2014). On the Covariance of ICP-based Scan-matching Techniques. *2016 American Control Conference*. Boston, United States: Proceedings 2016 American Control Conference.
- Veth, M. J. (2009). *Fusion of imaging and inertial sensors for navigation* . Wright-Patterson Air Force Base, Ohio: Air force institute of technology.