Analyse de scènes dynamiques par des systèmes de perception multimodale embarqués

Sergio Alberto Rodriguez Florez

université
PARIS-SACLAY

Analyse de scènes dynamiques par des systèmes de perception multimodale embarqués

Habilitation à diriger des recherches
de l'Université Paris-Saclay

présentée et soutenue à Saclay, le 27 mars 2023, par

Sergio Alberto RODRÍGUEZ FLÓREZ

Composition du jury

Philippe MARTINET Directeur de Recherches, INRIA Sophia-Antipolis Méditerranée	Rapporteur
Pascal VASSEUR Professeur des Universités, Université de Picardie Jules Verne	Rapporteur
Abdelaziz BENSRHAIR Professeur des Universités, INSA Rouen Normandie	Rapporteur
Valérie GOUET-BRUNET Directrice de recherches, Institut National de l'Information Géographique et Forestière IGN	Examinatrice
Samia BOUCHAFA-BRUNEAU Professeur des Universités, Université d'Evry Val d'Essonne	Examinatrice
Pascal LAZABAL Professeur des Universités, Université Paris-Saclay	Examinateur

université
PARIS-SACLAY

Titre : Analyse de scènes dynamiques par des systèmes de perception multimodale embarqués

Mots clés : Vision par ordinateur, Fusion de données multi-capteurs, Localisation, Perception multimodale, SLAM, Systèmes embarqués

Résumé : Dans le domaine de la robotique mobile, mes travaux de recherche portent sur l'analyse de scènes dynamiques par des systèmes de perception multimodale embarqués. Ceci demeure un verrou scientifique complexe, mais essentiel pour des applications avec un fort impact sociétal tels que les systèmes de transport intelligents où se situent les systèmes d'assistance à la conduite et les voitures à conduite automatisée. Une synthèse et des perspectives structurés sur deux axes thématiques sont présentés.

Title : Embedded multi-modal perception systems for dynamic scene understanding

Keywords : Computer vision, Multi-sensor data fusion, Localization, Multi-modal Perception, SLAM, Embedded Systems

Abstract : In the context of mobile robotics, this work reports research addressing scene understanding using embedded multi-modal perception systems. Scene understanding remains a complex problem, essential for high society impact applications such as Intelligent Transportation Systems, involving in particular ADAS and Autonomous Vehicles. Results and outlook based on the conducted research are detailed and structured following two research perspectives.

Table des matières

Table des figures

Chapitre 2

Synthèse des travaux de recherche

Sommaire

2.1 Introduction

Ce chapitre restitue une synthèse explicative de mes travaux de recherche. Elle se structure sur les axes de recherches et leurs verrous associés comme ils ont été introduits dans la Sec. 1.4.1. Les travaux sont détaillés en apportant un intérêt particulier sur les démarches scientifiques menées. Enfin, ils seront mis en perspective en tenant compte des avancées du domaine et des enjeux sociétaux.

Mes travaux évoluent dans le domaine applicatif des systèmes de transport intelligents, et plus particulièrement des véhicules terrestres. Mes expériences et mes productions scientifiques ont commencé à se construire à partir de mon master recherche en Technologies de l'Information et des Systèmes. Dans ce cadre, je me suis intéressé aux problématiques liées à l'étalonnage multi-capteur LiDAR-vision dans le cadre du projet de recherche ANR LOVe (Predit/Pôle System@tic). Par la suite, mes travaux de thèse m'ont permis de connaître les systèmes de vision et d'étudier leur apport aux problématiques de localisation et de suivi d'objets pour des applications de véhicules à conduite automatisée (voir Sec. 2.2). Mes travaux ont été approfondis et associées au projet de recherche ANR CityVIP s'étalant sur une année de recherches postdoctorales (voir Sec. 2.3).

Mon recrutement au sein du laboratoire d'Électronique Fondamentale de l'Université Paris-Sud et mon intégration par la suite au laboratoire SATIE de l'ENS Paris-Saclay ont accompagné mes objectifs thématiques. Ces objectifs ont été adressés par des multiples actions que j'ai menée en étroite coordination à mes missions en tant qu'enseignant-chercheur. Elles sont présentées dans la Sec. 2.4.

Ce document dresse le bilan en Chap. 3 et formule les perspectives vers lesquelles je projetterai mes actions et mes efforts dans la direction des recherches.

2.2 Travaux de thèse

Les travaux de recherche menés dans le cadre de ma thèse ont porté sur la problématique sociale de l'accidentalité routière, propre à notre civilisation moderne. Dans cette étude, je me suis focalisé principalement sur les accidents routiers dans des milieux urbains complexes et dynamiques. En effet, les statistiques confirment un taux élevé d'accidents dans ce contexte, produits principalement par des erreurs humaines.

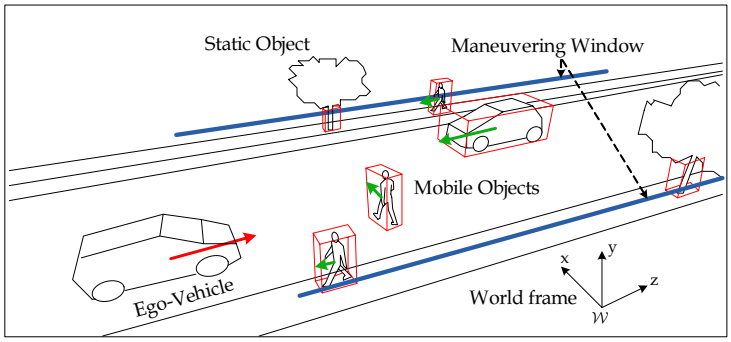


Figure 2.1 – Concept de la carte locale dynamique

Les systèmes d'aide à la conduite, les fonctions de conduite automatisé et les voitures autonomes peuvent améliorer la sécurité routière en aidant les utilisateurs via des avertissements de situations dangereuses ou en déclenchant des actions appropriées en cas de collision imminente (airbags, freinage d'urgence, etc). Dans ce cas, la connaissance de la position et de la vitesse des objets mobiles

aux alentours constitue une information clé. Cette connaissance est construite essentiellement par l'analyse d'informations issues d'une perception proprio et extéroceptive du porteur.

Ainsi, je me suis focalisé sur **la détection et le suivi d'objets dans une scène dynamique**. En remarquant que les systèmes multi-caméras sont de plus en plus présents dans les véhicules et sachant que le LiDAR est performant pour la détection d'obstacles, je me suis intéressé à l'apport de la vision stéréoscopique à la perception géométrique multimodale de l'environnement [10, 11].

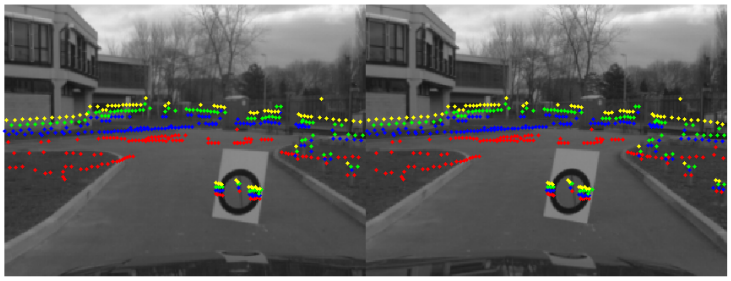


Figure 2.2 – Projections de données LiDAR grâce à l'estimation de paramètres d'étalonnage LiDAR/vision

Une première solution à cette problématique de perception consiste à détecter et suivre les objets alentours en se basant uniquement sur des informations télémétriques aperçues depuis la plate-forme mobile. Cette perception ego-centrée et dynamique engendre l'observation de trajectoires complexes et rarement approchées par des modèles linéaires qui peuvent, néanmoins, être simplifiées par la compensation du mouvement propre. C'est ainsi que nous avons développé un système capable de fournir une carte locale dynamique de l'environnement proche dans laquelle les objets observés sont localisés et suivis par rapport à un repère de référence fixe. Ce concept est illustré par la Fig. 2.1, il requière l'échange d'informations entre les systèmes de perception (LiDAR et vision) ainsi qu'une estimation précise du mouvement du porteur.

Afin de fusionner les informations géométriques entre le LiDAR et le système de vision, nous avons développé un procédé de calibrage qui détermine de manière robuste les paramètres extrinsèques et évalue les incertitudes sur ces estimations [39, 31]. Cette méthode utilise un ensemble d'observations, par un système mono-caméra et un LiDAR, d'une mire de calibrage caractérisée par deux cercles concentriques. La position relative des capteurs est ensuite obtenue en résolvant le problème d'orientation absolue. Les résultats obtenus sont illustrés dans la Fig. 2.2 où les informations LiDAR ont été projetées sur les images du système de vision à travers les paramètres extrinsèques estimés.

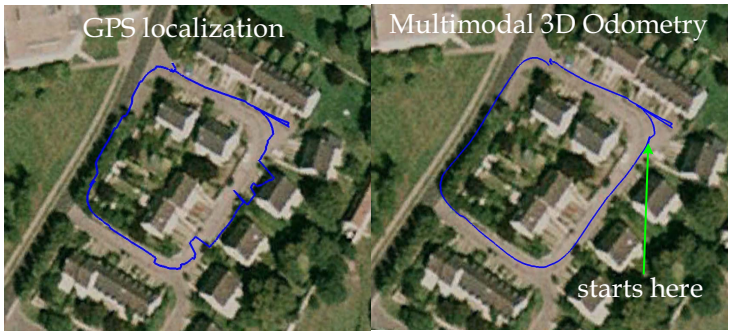


Figure 2.3 – Comparaison des résultats de localisation donnés par la solution GNSS et ceux de la méthode d'odométrie multimodale (Odométrie visuelle et capteurs proprioceptifs)

Nous avons ensuite proposé une méthode d'odométrie visuelle temps-réel permettant d'estimer le

mouvement propre du véhicule afin de simplifier l'analyse du mouvement des objets dynamiques [30]. Elle combine, dans un critère non-linéaire, l'estimation de l'*ego-motion* fondée sur une mesure éparse du mouvement apparent sur des images stéréo et une estimation obtenue par des contraintes de rigidité de la scène modélisée par un tenseur quadrifocal. L'utilisation d'un schéma robuste permet d'estimer le mouvement en minimisant les biais induits par les objets mobiles et les erreurs d'appariement.

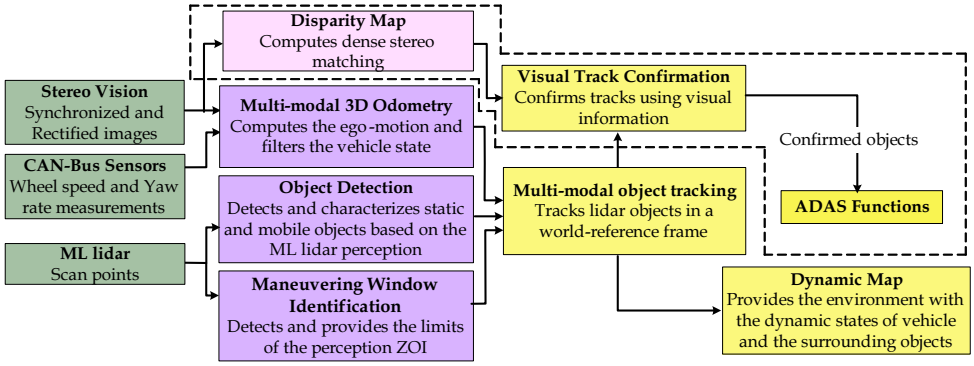


Figure 2.4 – Diagramme fonctionnel du système de perception multi-capteur étudié dans le contexte de la thèse.

Cette méthode a permis la conception d'un démonstrateur d'odométrie visuelle pour la localisation du véhicule en temps réel dans un environnement urbain (voir Fig. 2.3).

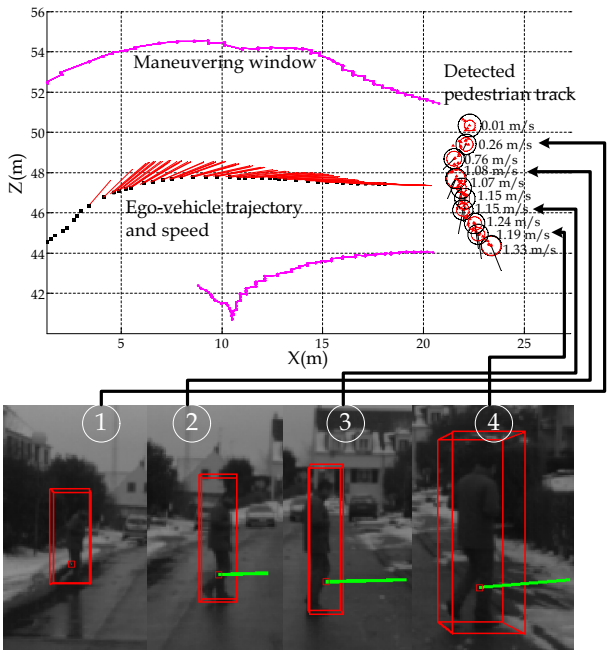


Figure 2.5 – Trajectoire suivie par un piéton qui traverse la route. La figure illustre la carte locale dynamique avec la trajectoire, vitesse et taille du piéton estimés ainsi que l'état dynamique du porteur. La suite des images illustre la confirmation visuelle de l'objet suivi (boîte englobante rouge) et la projection de son vecteur vitesse (vert).

Dans un second temps, nous avons montré comment l'intégrité de la détection et du suivi d'objets par LiDAR [29] peut être améliorée en utilisant une méthode de confirmation visuelle qui procède par reconstruction dense de l'environnement 3D [28]. Cette méthode vise à réduire les taux de fausses

alarmes dans la fonction de détection et de suivi. Les stratégies de perception asynchrone déployées dans le système proposé ont requis une gestion de l'asynchronisme des données à l'aide de filtres prédictifs.

Pour finir, le système de perception multimodale a été prototypé et intégré sur une plateforme automobile (voir Fig. 2.4), permettant ainsi d'analyser expérimentalement, les différentes approches proposées dans des situations routières en environnement non contrôlé comme illustré dans la Fig. 2.5.

L'évaluation expérimentale de l'approche proposé a démontré une bonne performance en termes du taux de détection du système avec une intégrité augmentée (i.e. faible taux de fausses alarmes) dans des cas d'usage réalistes. En considérant l'asynchronisme des sources de données du système de perception, la solution proposée garanti également la consistance temporelle des données fusionnées.

2.3 Activité de recherche postdoctorale

La navigation des véhicules à conduite automatisée (VCA) ou des véhicules autonomes (VA) impose le déplacement du véhicule sur un espace contraint par la chaussée et par la présence d'autres usagers de la route, dénommé l'espace navigable. En effet, la caractérisation et l'identification en temps-réel de l'espace navigable est un prérequis sémantique de compréhension d'une scène routière urbaine.
Je me suis intéressé à cette problématique de recherche dans le cadre de mon contrat postdoctoral et j'ai participé à son étude expérimentale au sein des activités fédérées par le projet ANR CityVIP au laboratoire Heudiasyc.

La caractérisation de l'espace navigable a été formalisée comme étant un problème d'inférence nécessitant la combinaison de deux contraintes spatiales : celle définie par la structure statique de la scène (e.g. le trottoir, la chaussée) et celle imposée par les éléments dynamiques pouvant l'occuper partiellement (e.g. usagers de la route). Le résultat attendu est l'identification de l'espace apte à la planification de trajectoire du véhicule. Cet espace satisfait, au sens d'un raisonnement conjonctif, toutes les contraintes issues des sources de données.

La stratégie adoptée pour cette étude se structure sur l'utilisation conjointe d'un système d'information Géographique (e.g. carte 3D), d'un système de localisation GNSS/INS et d'un capteur extéroceptif du type LiDAR permettant la localisation des objets pouvant induire une collision. La pertinence de la stratégie dépend étroitement de la qualité des informations issues de la carte et de la localisation du véhicule dans cette dernière.

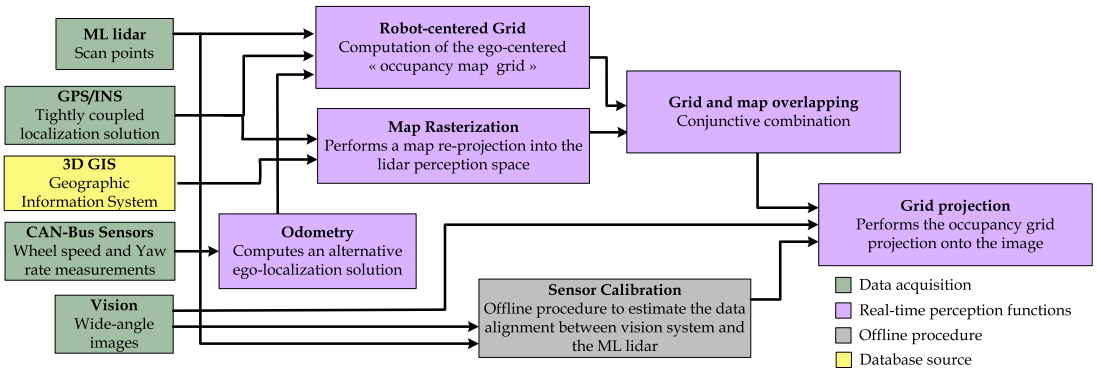


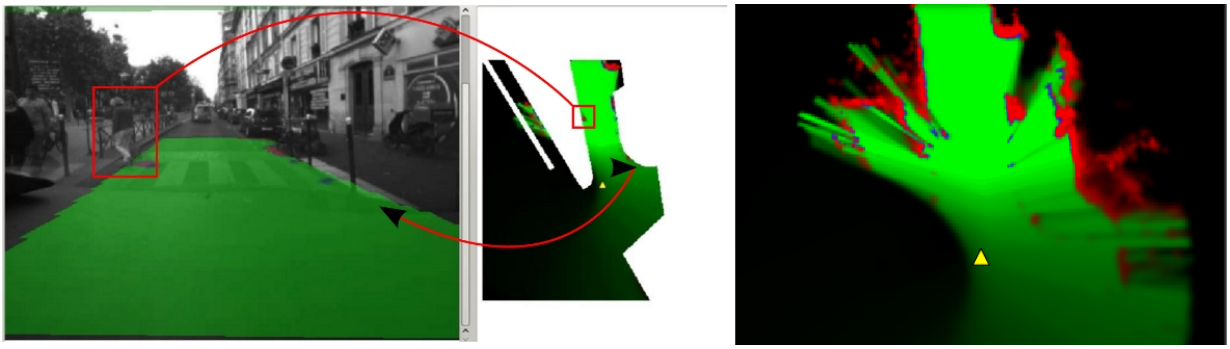
Figure 2.6 – Aperçu fonctionnel du système de perception multi-capteur dédié à la caractérisation de l'espace navigable.

Le système de perception multi-capteur illustré dans la Fig. 2.6, opère dans un premier temps la localisation du véhicule par la fusion des données d'un récepteur GNSS/INS et de l'odométrie du véhicule. Dans un deuxième temps, une analyse ego-centrée de la scène est réalisée par l'intégration spatiale et temporelle des données LiDAR dans la cadre d'une grille d'occupation crédibiliste. Cette approche a démontré être bien adaptée pour la détection d'objets [56]. Enfin, une combinaison conjonctive de la grille crédibiliste et des informations cartographiques est réalisée afin d'inférer l'espace navigable.

L'évaluation expérimentale du système de perception a été obtenue par la projection de l'espace navigable inféré sur des images de la scène observée. Pour ce faire, une campagne de collecte de donnés a été menée dans le 12^{éme}arrondissement de Paris. Les données utilisées couvrent une trajectoire de 45 Km. La taille de la grille d'occupation a été définie de 30$\times$30m et composée par 9000 cellules.

Les résultats obtenus ont confirmé la consistance de l'approche proposée où la grille d'occupation crédibiliste identifie l'espace libre tout en localisant les usagers de la route et l'empreinte des bâtiments. En complément, la carte contraint les estimations sur la chaussée pour enfin définir l'espace navigable ciblé comme l'illustre la Fig. 2.7.

Ces travaux ont permis la démonstration du véhicule prototype dans le cadre du congrès internationale IEEE IV2011 à Baden-Baden, Allemagne. Une communication au congrès international IEEE IV2012 a été également réalisée [27].



(a) Espace navigable issu de la fusion LiDAR - Carte (b) Grille d'occupation crédibiliste LiDAR

Figure 2.7 – Résultats expérimentaux obtenus pour la détection de l'espace navigable.

A l'issue de mes activités de recherche postdoctorales et ayant été motivé par mon goût pour l'enseignement et le transfert de connaissances, j'ai postulé sur un poste de Maître de Conférences à l'IUT d'Orsay de l'Université Paris Sud 11 (actuellement Université Paris-Saclay). Ce poste se situait en parfaite adéquation avec mes expériences et me permettait de poursuivre mes activités de recherche dans le même domaine disciplinaire que ma formation de Master, Doctorale et Postdoctorale, renforçant ainsi la cohérence globale de ma carrière.

2.4 Analyse de scènes dynamiques par des Systèmes de Perception Multimodale Embarqués (SPME)

L'introduction des automatismes et la récente robotisation des systèmes de transport ont été et sont aujourd'hui motivées par un taux croissant d'accidentalité routière et par le besoin accru d'améliorer la mobilité des personnes. Les études scientifiques identifiant la causalité des accidents ont démontré

qu'un nombre important d'accidents se produisent du fait d'erreurs humaines. D'autres études s'intéressent aux améliorations possibles dans le but de fluidifier le trafic et de réduire sa consommation énergétique [63].

De nombreuses contributions dans le domaine de la robotique mobile émergent depuis plus de 60 ans pour développer et améliorer les Systèmes de Transport Intelligents. Une première approche s'est intéressée à l'étude et à la conception des systèmes d'aide à la conduite dénommés ADAS[1]. Ces systèmes effectuent l'extraction et la mise à disposition des informations pertinentes pour faciliter la prise de décision au conducteur. Plus récemment, les avancées dans le domaine des systèmes de transports ont subi une accélération vertigineuse avec l'introduction d'automatismes pouvant engager une prise de décision et une action lors de la conduite. Cette accélération a été facilitée par des changements dans la législation et par la standardisation des systèmes de conduite automatisée suivant les niveaux SAE[2] [74].

Dans ce contexte, les systèmes de perception sont un élément structurel pour la conception des systèmes de conduite automatisée [43, 93]. Ainsi, les fonctions de prise de décision, de planification de trajectoire et de contrôle inhérentes à la conduite automatisée ne peuvent pas être opérées sans assurer l'acquisition et l'analyse d'informations décrivant l'état du véhicule et de son environnement. Les systèmes de perception doivent également répondre à des contraintes applicatives fortes pouvant se qualifier en termes de précision, de redondance, de la consistance de ses estimations, de sa cohérence temporelle et de l'embarquabilité du système.

Une étude menée sur les travaux de l'état de l'art m'a permis de déceler plusieurs verrous scientifiques associées à des problèmes dans un contexte applicatif tels que : i) la localisation, ii) l'analyse des scènes dynamiques, iii) l'analyse du comportement des usagers, des occupants et les interactions avec les systèmes de perception, iv) l'analyse de l'intégrité des systèmes et v) la conception des SPME. La liste de ces problématiques n'est pas exhaustive mais elle décrit une partie importante du large spectre disciplinaire de la perception pour les Systèmes de Transport Intelligents.

Un système de perception peut se synthétiser par l'association fonctionnelle des capteurs, des algorithmes et d'une architecture de calcul dédiée à leur fonctionnement embarqué. Mes travaux de recherche couvrent plus particulièrement l'étude et le développement des systèmes embarqués de perception exploitant de multiples modalités de mesure et d'estimation.

Les travaux que je décris dans ce manuscrit ont été effectués au sein de l'Université Paris-Saclay dans les Unités de Recherche de l'Institut d'Électronique Fondamentale (IEF), jusqu'à 2019, et par la suite au laboratoire des Systèmes et Applications des Technologies de l'Information et de l'Énergie (SATIE). Ces travaux ont aussi été possibles grâce à des collaborations sur des projets de recherche au niveau national et international. Ils sont structurés suivant deux axes majeurs :

- **Conception, étude et évaluation d'algorithmes de perception multimodale,**
- **Conception conjointe logicielle/matérielle d'algorithmes de perception multimodale**

Le premier axe porte sur la conception, l'étude et l'évaluation d'algorithmes de perception multimodale dédiées à la localisation et à l'analyse de scènes dynamiques. Ces travaux m'ont permis de proposer et d'évaluer de multiples stratégies adaptées à la gestion de sources d'information hétérogènes (sémantiques ou non) et la modélisation des erreurs et des incertitudes afin de répondre à des

1. *Advanced Driving Assitance Systems*
2. *Society of Automotive Engineers*

problèmes et des cas d'usage complexes. Dans le cadre de ces activités, j'ai pu également m'intéresser aux méthodologies et aux moyens de qualification fonctionnelle des systèmes de perception.

Une ouverture thématique se profile vers les interactions entre les systèmes de perception et les utilisateurs (usagers de la route ou occupants du véhicule). Ces interactions sont aujourd'hui mal connues et leur compréhension peut s'avérer complexe puisqu'elles dépendent fortement du contexte de la conduite. Pour ces raisons, il est nécessaire dans un premier temps de développer des outils permettant la caractérisation de ces interactions dans des situations critiques réelles.

Le deuxième axe s'oriente vers l'étude des systèmes de perception multimodale sous une perspective plus large incluant les architectures de calcul embarquées dans le processus de conception. En effet, la complexité algorithmique limite l'utilisation d'approches de perception dans des conditions et des scénarios réalistes. En revanche, ces contraintes peuvent être levées en adaptant les algorithmes à des architectures optimisées tout en préservant la consistance des estimations.

Dans la suite du document, la synthèse de l'état de l'art et les contributions seront développées par axe. Elles seront supportées par les résultats obtenus et des conclusions sont dressées à leur issue. Mes travaux de recherche ont contribué à lever des verrous scientifiques du domaine de la robotique mobile appliquées aux véhicules intelligents. Ces travaux ont également répondu à des attentes technologiques exprimées par les activités de recherche et de développement industriels. De recherches en cours sont également décrites et des perspectives sont identifiées sur la base de ces expériences.

2.4.1 Conception, étude et évaluation d'algorithmes de perception multimodale

L'automatisation de la conduite des véhicules au sens large (terrestres, aériens ou maritimes) nécessite des systèmes instrumentés et embarqués. Ces systèmes doivent être capables de rapporter des informations pertinentes à l'inférence de l'état du vecteur (le système porteur) et celui de la scène (l'environnement aux alentours). L'état est généralement composé d'un ensemble d'attributs cinématiques (e.g. position, vitesse, orientation), spatiaux (dimensions) ou sémantiques (classe, nombre d'objets) définis dans le temps (discret).

Cette problématique a été étudiée, par la communauté scientifique, sous deux perspectives majeures. La première formalise le problème tout en découplant la localisation du vecteur de celle portant sur l'inférence de l'état de la scène. La deuxième stratégie de perception traite globalement le problème en inférant une description de l'environnement et en localisant le vecteur par rapport à cette description.

Suivant la première perspective, nous rencontrons un large nombre de travaux s'intéressant aux problèmes de la localisation (ici ego-localisation) et de la détection et du suivi multi-objets. Les systèmes de positionnement font appel principalement à l'utilisation de récepteurs GNSS [3] mais aussi à des couplages multimodaux avec des algorithmes odométriques basés sur des sources de données du type : encodeurs de vitesse, de la vision, de la télémétrie LiDAR ou de l'intégration inertielle, entre autres.

Dans la deuxième stratégie méthodologique, nous retrouvons les approches dénommées SLAM [4]. Ces approches ont été largement étudiés dans des conditions d'usage très limitées telles que les environnements statiques d'intérieur. Ces conditions sont incompatibles avec les verrous définis par le domaine des véhicules à conduite automatisée. Néanmoins, grâce aux contributions et aux avancées

3. *Global Navigation Satellite System*
4. *Simultaneous Localization And Mapping*

de l'état de l'art, les méthodes de SLAM s'adressent aux environnements dynamiques d'extérieur. Les méthodes d'inférence et les moyens de perception qui s'imposent aujourd'hui sont essentiellement des méthodes d'optimisation de graphes (e.g. GraphSLAM) exploitant des sources de type vision, télémétrie LiDAR assistées par des mesures inertielles.

Malgré les avancées technologiques dans le développement des capteurs, les limitations qu'entraîne chaque technologie complexifient la conception d'un système de perception fiable et précis dans un large spectre fonctionnel. C'est ainsi que la conception des systèmes de perception multimodale s'avère pertinente dans la recherche des solutions plus performantes au sens de leur précision, robustesse et fiabilité.

La performance de ces systèmes de perception est estimée à l'expérimentale par la qualification des informations en sortie suivant des indicateurs (RMSE[5], MAE[6], ATE[7], RPE[8]) et par la vérification de la consistance des estimations comprises dans des bornes d'erreur admissibles. La quantification et la propagation des incertitudes des estimations issues de ces méthodes est aussi diverse que les techniques d'estimation et de fusion de données. Cette diversité rend complexe la propagation des erreurs du système de perception et limite la consistance des estimations. En conséquence, les défauts du système sont peu explicables et vérifiables pour un large nombre de cas d'usage. Ceci est un prérequis pour l'analyse d'intégrité de systèmes de perception multimodale. Elle permettra d'assurer la qualité et la disponibilité des informations issues du système pour la prise de décisions et d'actions.

Afin de lever ce verrou, l'analyse d'intégrité des systèmes de perception multimodale pour les véhicules à conduite automatisée s'oriente vers la caractérisation d'indicateurs (aussi dénommés marqueurs) vérifiables répondant aux conditions nécessaires pour assurer un fonctionnement fiable du système de perception dans les cas d'usage. Ces techniques restent actuellement peu généralisables constituant ainsi une problématique scientifique ouverte et très dynamique.

Dans la suite de cette introduction, j'aborderai le problème de la localisation multimodale et je présenterai quelques contributions dans la perception de l'environnement par des systèmes multi-capteurs. J'introduirai mes travaux adressant la modélisation et l'introduction d'informations contextuelles dans les systèmes de perception. Je finirai avec une synthèse des activités de recherche vers l'étude des interactions des SPME et les usagers.

2.4.1.1 Contributions à la localisation précise et multimodale

Le problème de la localisation consiste à inférer, à partir des informations issues des moyens de perception embarqués, la position du porteur exprimée dans un référentiel fixe, orthonormé et direct.

La position du porteur s'exprimant dans l'espace 3D, dénotée ${}_{3D}\mathbf{p}$, est un vecteur d'état de dimension 6×1 pouvant se décomposer en un vecteur de position et un vecteur d'orientation (ici on évoque une représentation axes-angle) comme l'indique l'Equ. 2.1 :

$$ {}_{3D}\mathbf{p} = \left[\begin{array}{cccccc} x, & y, & z, & r_x, & r_y, & r_z \end{array} \right]^T \tag{2.1} $$

où les coordonnées x, y, z, exprimées en mètres, définissent la position du porteur par rapport à l'origine du repère de référence, le vecteur normalisé $\frac{[r_x, r_y, r_z]^T}{\left\| [r_x, r_y, r_z]^T \right\|}$ représente l'axe de rotation et $\theta = \left\| [r_x,\, r_y,\, r_z]^T \right\|$ représente l'angle de rotation autours de l'axe en radian.

5. *Root Mean Square Error*
6. *Mean Absolute Error*
7. *Absolute Trajectory Error*
8. *Relative Pose Error*

La représentation de la position du porteur peut dans certains cas être simplifiée sous l'hypothèse que le porteur est en occurrence un véhicule terrestre. Il peut évoluer dans un environnement présentant peu de changements en altitude. Ainsi, la localisation peut s'exprimer, dans sa forme simplifiée, comme dans l'Equ. 2.2 :

$$ {}_{2D}\mathbf{p} = \left[\begin{array}{ccc} x, & y, & \alpha \end{array} \right]^T \tag{2.2} $$

où x et y représentent la position du porteur sur le plan du sol et α est l'orientation du véhicule (i.e. cap). Cette simplification réduit significativement la dimension du problème. En revanche, elle induit des erreurs d'approximation dans la trajectoire estimée.

Dans ce cadre, la représentation des erreurs dépend étroitement du formalisme méthodologique, des hypothèses et de la modélisation du processus d'estimation. En effet, dans l'état de l'art, nous pouvons modéliser les incertitudes d'estimation par les moments statistiques de deuxième ordre (i.e. la moyenne et la variance) comme c'est le cas des estimations issues du filtre de Kalman et du filtre du Kalman étendu avec des hypothèses de bruit blanc Gaussien et la décorrélation des variables dans le vecteur d'état.

Les méthodes de filtrage particulaire permettent de soulever des hypothèses concernant la distribution statistique du bruit de mesure et la linéarité du modèle d'évolution. Ainsi, la distribution des particules dans l'espace de paramètres modélise les incertitudes d'estimation.

Les méthodes basées sur l'optimisation de paramètres modélisent les incertitudes d'estimation au travers des résidus et leur variance. Des modèles heuristiques peuvent également permettre la modélisation des erreurs par l'utilisation de scores/marqueurs estimés en rapport avec des contraintes contextuelles ou des informations *a priori*.

Pour étudier l'intégrité des systèmes de localisation multimodales, j'ai entrepris des recherches, dans un premier temps, afin de saisir l'impact de multiples sources d'information en vue d'augmenter la précision et de caractériser les incertitudes d'estimation. Dans un deuxième temps, je me suis intéressé à l'identification des propriétés adéquates à l'évaluation de l'intégrité de ces systèmes pour les véhicules à conduite automatisée.

Ainsi, mes premières recherches se sont portées sur la conception et le prototypage d'un système de localisation multimodale tout en s'intéressant à la modélisation des erreurs d'estimation. Le diagramme fonctionnel du système proposé dans [68] est illustré dans la Fig. 2.8.

Ce prototype a été étudié dans le cadre de la localisation d'un véhicule en milieu urbain. Les conditions dans ce type d'environnement sont particulièrement difficiles et dépendent des capacités sensorielles des moyens de perception individuelles. En effet, les récepteurs GNSS dans un contexte urbain sont fréquemment exposés à des situations de perte de signal (ex. tunnels, forêts) et à des multi-trajets en présence de bâtiments. Les systèmes de vision sont sensibles aux conditions d'exposition et aux occlussions induites par un trafic dense. Également, les Systèmes d'Information Géographiques se heurtent à certaines limites telles que les imprécisions, les approximations topologiques et les erreurs dues aux changements dans l'infrastructure routière. Certes, les multiples modalités de perception ont des limitations. Néanmoins, leur exploitation conjointe permet de lever un nombre important de contraintes tout en augmentant la précision du positionnement inféré.

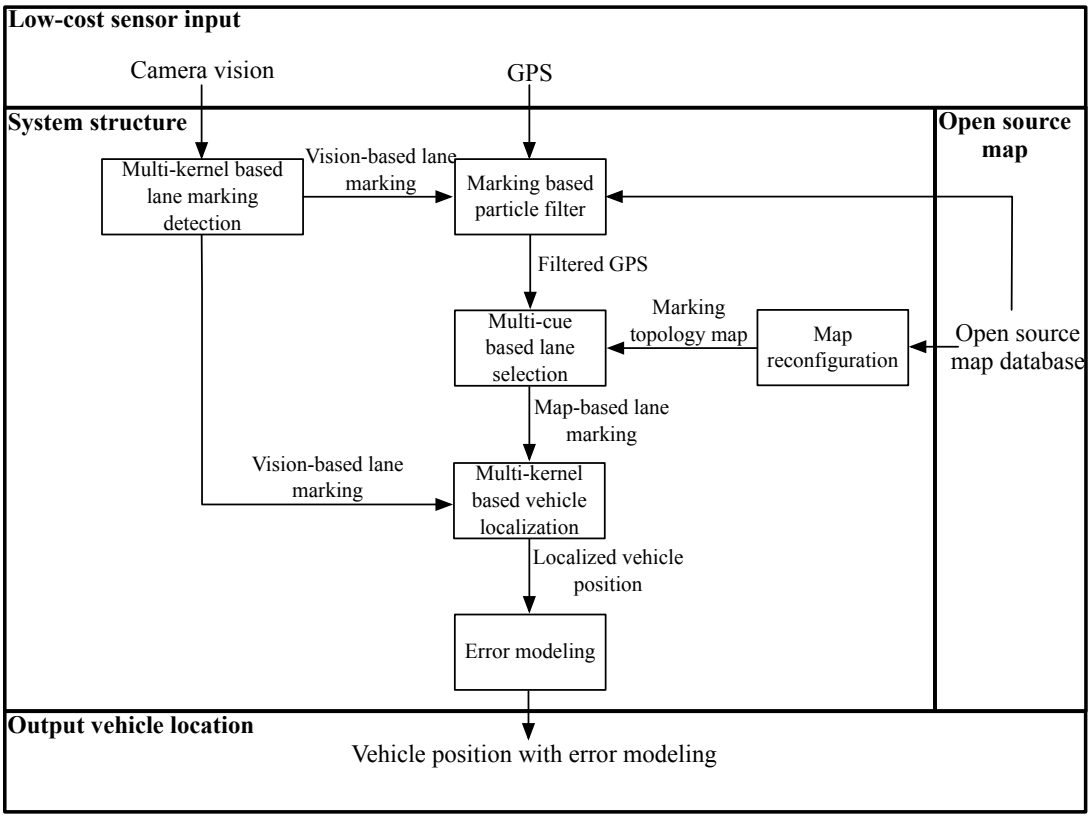


Figure 2.8 – Système de localisation multimodale avec modélisation des erreurs. Ce système opère l'inférence de la position d'un véhicule en utilisant trois sources distinctes d'information : un détecteur de marquage routier par vision monoculaire, un récepteur GNSS et un Système d'Information Géographique (SIG).

Les choix algorithmiques du système de perception de la Fig. 2.8 assurent les interactions entre un algorithme de détection de marquage routier basée sur la vision monoculaire, un algorithme de filtrage de données issues d'un récepteur GNSS et un système d'information Géographique.

La détection du marquage routier traite la séquence d'images issue d'une caméra projective. L'analyse des images est effectuée dans l'espace IPM[9] établi par d'une transformation homographique du plan image vers le plan routier (transfert 2D vers 2D). Cette transformation impose une hypothèse portant sur la planéité de la chaussée à la portée du champ de vue de la caméra et de faibles changements d'orientation, notamment en roulis et en tangage. Par la suite, la segmentation de l'image permet l'identification des pixels appartenant au marquage à travers un filtre de 2ème ordre et l'extraction de blobs contraint sur un partitionnement de l'image. La procédure de détection se conclue par l'optimisation paramétrique d'un modèle de marquage polynomial de deuxième ordre. Cette optimisation est opérée en utilisant une méthode dénommée Méthode d'Estimation Multi-kernel (MKE) avec une pondération hiérarchisée. Elle s'inspire de la Transformé de Hough Probabiliste introduite par [48]. Dans cette méthode on détecte les marquages en maximisant la probabilité d'un set de paramètres suivant l'expression :

$$p_{mke}(d,e) = \frac{1}{n_t^{mke}} \sum_{i=1}^{n_t^{mke}} w_{mke}\left(x_i^{bev}, y_i^{bev}\right) \cdot G_{pi}\left(c^*, d, e, x_i^{bev}, y_i^{bev}\right), \tag{2.3}$$

où n_t^{mke} est le nombre total de pixels blancs dans l'image, c^* est le paramètre du modèle repré-

9. acronyme en anglais de : *Inverse Projective Mapping*

sentant la position longitudinale des marquages (ici. de gauche, de droite), $w_{mke}\left(x_i^{bev}, y_i^{bev}\right)$ est une pondération hiérarchisée corrélée au contraste de l'image et $G_{pi}\left(c^*, d, e, x_i^{bev}, y_i^{bev}\right)$ est l'indice de vraisemblance d'un pixel par rapport à un ensemble de paramètres du polynôme. Ainsi, la méthode MKE permet l'obtention des paramètres du modèle de marquage dans l'espace cumulative probabiliste. Les variations induites par les conditions d'exposition sont gérées grâce à une pondération hiérarchisée des régions peu contrastés de l'image IPM. Nous avons proposé un indicateur associé à la confiance de la détection, dénommé $conf_{ld}$, se basant sur deux critères : le nombre des pixels ayant une forte vraisemblance au modèle de marquage ($G_{pi} > Th_{ld}^*$) et à la distribution des pixels.

Une première estimation filtrée de la position du véhicule, ${}_{2D}\mathbf{p}$, (voir Filtrered GPS dans la Fig. 2.8) est obtenue à travers un filtre à particules (PF). Ce filtre utilise, en entrée, les données issues d'un récepteur GNSS et il se sert des estimations du marquage routier comme modèle d'évolution lors de la phase de prédiction. Ce filtrage réduit le bruit de mesure du récepteur GNSS et isole efficacement les données aberrantes. Les incertitudes du processus d'estimation sont modélisées par les moments statistiques de 2ème ordre dans l'espace d'état des particules, $s_k^{(i)}$, défini comme suit :

$$s_k^{(i)} = \left[\begin{array}{cccc} x_{enu,k}^{(i)} & y_{enu,k}^{(i)} & v_{enu,k}^{(i)} & \alpha_{enu,k}^{(i)} \end{array} \right]^T \tag{2.4}$$

où $\left(x_{enu,k}^{(i)}, y_{enu,k}^{(i)}\right)$ sont les coordonnées en mètres dans le référentiel orthonormé direct East-North-Up (ENU), $v_{enu,k}^{(i)}$ est la vitesse linéaire en $m \cdot s^{-1}$ et $\alpha_{enu,k}^{(i)}$ est l'angle d'orientation en degrés.

Les informations du système de cartographie (SIG) nécessitent quant à elles, d'une transformation topologique afin de permettre leur traitement et leur fusion avec d'autres sources de données. L'algorithme proposé structure le réseau routier en utilisant deux niveaux topologiques : des segments et des cellules. Les segments représentent la ligne centrale des routes et les cellules représentent les voies sur un segment routier. Ainsi, la notion de cellule facilite l'association avec les limites de la voie définies par le marquage routier. Grâce à cette topologie, le système de localisation opère l'association de la position filtrée à un segment du SIG par la minimisation d'un critère de distance euclidienne. L'association à une cellule est effectuée par un mécanisme probabiliste inférant les changements de voie sur la base d'un critère temporel et de la détection d'un troisième marquage sur la voie.

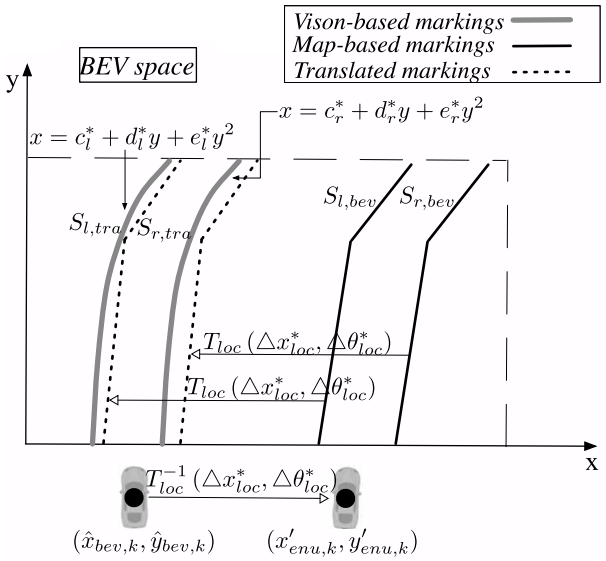
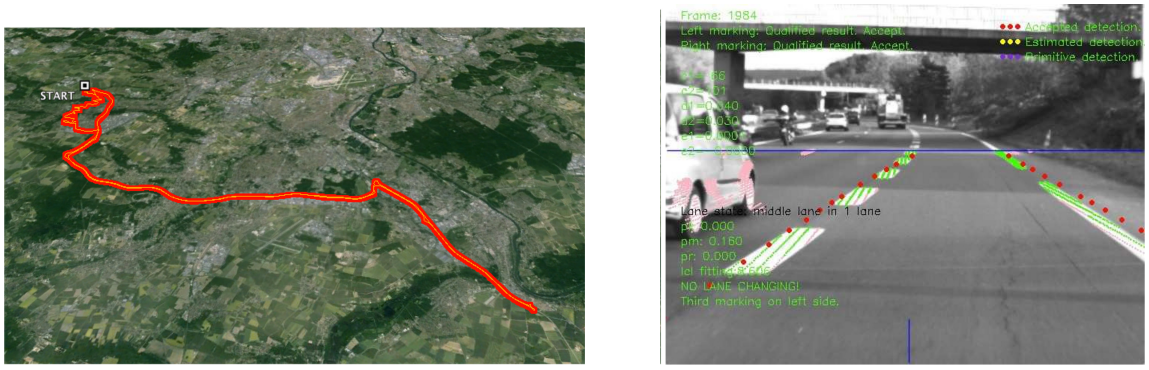


Figure 2.9 – Procédure d'alignement du marquage routier et la topologie issue d'un Système d'Information Géographique.



(a) La trajectoire sur fond d'une image satellite

(b) Aperçu des images acquises avec des résultats de détection de marquage incrustés

Figure 2.10 – Trajectoire expérimentale de 7,5 Km pour la collecte de données de localisation.

Enfin, la localisation du véhicule est obtenue par le système de perception multimodale en alignant les marquages détectés à la structure de la cellule de la carte associée. Cette procédure est illustrée par la Fig. 2.9. L'alignement estime les erreurs latérales et d'orientation, dénotées $(\Delta x^*_{loc}, \Delta\theta^*_{loc})$, en maximisant la vraisemblance comme suit :

$$(\Delta x^*_{loc}, \Delta\theta^*_{loc}) = \underset{\Delta x_{loc}, \Delta\theta_{loc}}{\arg\max} \left[\sum_{i=l,r} \sum_{(x,y,)\in S_{i,tra}} G_{pi}\left(c^*_i, d^*_i, e^*_i, x, y\right) \right] \tag{2.5}$$

où c^*_i, d^*_i, e^*_i sont les paramètres du modèle du marquage détecté et x, y la position du véhicule.

La caractérisation des erreurs de mesure et leur propagation sont indispensables à la consistance de la fusion de données de perception. Néanmoins, les erreurs de la mesure ne suffisent pas à caractériser toutes les sources d'erreurs auxquelles est exposé le système de perception. C'est ainsi qu'un modèle d'erreur du système de perception a été proposé en caractérisant 3 sources d'erreurs : i) les fausses détections du marquage routier, ii) les erreurs d'alignement avec la topologie du SIG et iii) les erreurs dans l'infrastructure routière du SIG.

i) Les fausses détections du marquage routier sont identifiées par deux critères : le niveau de confiance estimé lors de la détection, $conf_{ld}$, et la largeur de la voie du segment SIG associé.

ii) Les erreurs de localisation dues à l'alignement avec la topologie du SIG se produisent lors de la convergence de la méthode de minimisation vers un minimum local. Ces conditions aberrantes sont caractérisées par un indicateur de confiance quantifiant la cohérence dans la probabilité de l'alignement sur l'ensemble de marquages (ici gauche et droit).

iii) Les erreurs dans l'infrastructure routière du SIG ont été identifiées par la cohérence entre le nombre de voies issues de la détection visuelle et celles indiqués par le SIG.

Un prototype du système de perception multimodale dédiée à la localisation a été validé à l'expérimental en utilisant des bases de données publiques de référence (KITTI [61]) et des données collectées sur la plateforme du laboratoire SATIE. La Fig. 2.10 illustre le parcours de validation de $7,5\,Km$.

Les résultats de ces recherches ont permis d'évaluer, dans un premier temps, le MAE [10] de la localisation à $1.006\,m$ dans un scénario du dataset KITTI. La Fig. 2.11.a illustre un positionnement bruité initial en bleu claire, la position filtrée en rose et la position estimée en sortie du système de localisation proposé en rouge. Ces premiers tests ont également permis d'observer la réponse des indicateurs

10. acronyme en anglais *Mean Absolute Error*

d'erreurs. La Fig. 2.11.b présente les situations identifiées comme étant peu fiables à l'égard des trois seuils de tolérance. Ces seuils ont été définis à partir d'une courbe d'efficacité dites courbe ROC dans des scénarios similaires. Les franches horizontales en bleu indiquent une faible confiance dans la détection du marquage, les franches horizontales vertes une faible confiance dans l'alignement des marquages avec le SIG et les franches horizontales roses indiquent une faible confiance dans la structure routière du SIG.

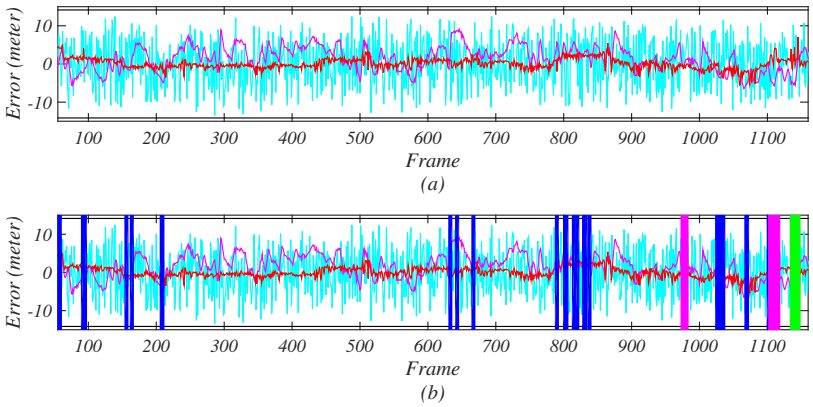


Figure 2.11 – Résultats du système de perception pour la localisation évaluée sur KITTI.

Dans un deuxième temps et afin de saisir l'impact et la contribution des indicateurs d'erreurs, nous avons identifié et référencé manuellement les situations où les données issues des sources limitent la capacité du système de localisation. Ces situations sont caractérisées par l'absence du marquage, les erreurs dans le SIG par rapport à la topologie et le nombre de voies répertoriés. La mise à l'écart de ces cas d'usage a permis la définition d'un MAE de référence de $0.86\,m$ pour ce scénario. Ainsi, une réduction du MAE à $0.925\,m$ a été obtenu en rejetant les estimations sur la base des indicateurs d'erreurs proposés.

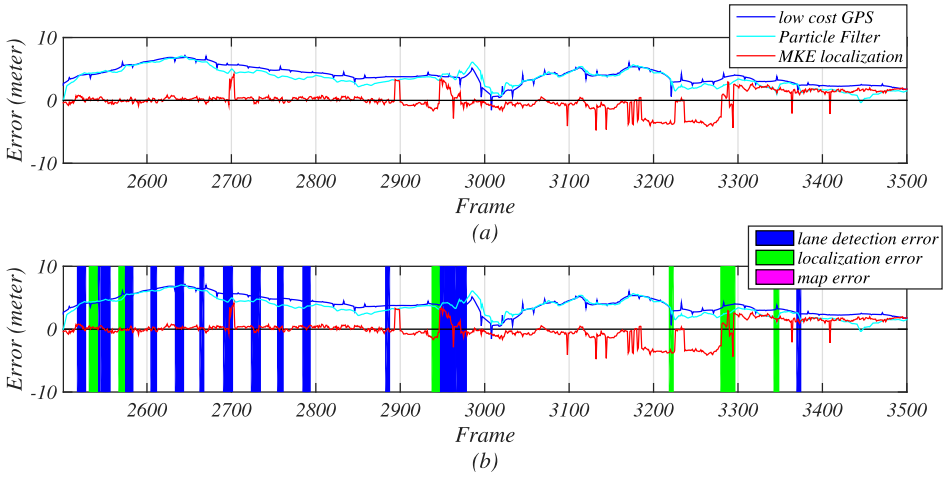


Figure 2.12 – Résultats du système de perception pour la localisation évaluée dans le scénario A du jeu de données collectées.

Les expériences menées en conditions de conduite réelles ont été analysées dans deux scénarios. Un récepteur GNSS Altus RTK a été utilisé comme référence de positionnement. Le système de localisation

utilise un récepteur GNSS mono-bande, une caméra et un SIG publique - OpenStreetMaps. Les résultats obtenus pour le scénario A ont confirmé un MAE de $1.10\,m$. La mise à l'écart des résultats en utilisant les indicateurs d'erreurs permettent une réduction du MAE à $0.98\,m$ par rapport au MAE de référence de $0.87\,m$. Le détail de ces résultats sont illustrés dans la Fig. 2.12.

Dans le scénario B, on constate également une réduction significative de l'erreur de localisation avec un MAE de $1.40\,m$. Certes, la phase de filtrage du système multimodal réduit et corrige le bruit de mesure mais elle ne permet pas d'éliminer les décalages en position induits par les incertitudes dans la propagation des signaux GNSS ou d'autres perturbations comme les multi-trajets. Ces décalages sont corrigés par les informations du SIG de l'infrastructure routière.

Les indicateurs d'erreur permettent également l'évaluation de la consistance des estimations en sortie. La mise à l'écart des estimations par les indicateurs d'erreur confirme leur pertinence par la réduction du MAE à $1.26\,m$ par rapport au MAE de référence situé à $1.11\,m$. Le prototype proposé reste néanmoins perfectible puisque les indicateurs d'erreurs sont sujets à des non-détections et à des fausses alarmes propres à tous les systèmes effectuant une classification binaire (ici l'occurrence ou non d'une erreur).

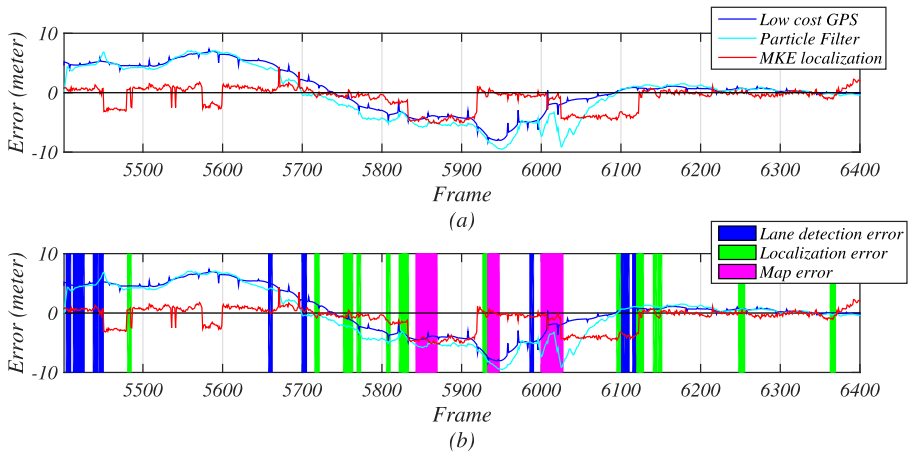


Figure 2.13 – Résultats du système de perception pour la localisation évaluée dans le scénario B du jeu de données collectées.

L'étude de ce système de localisation a permis de confirmer l'importance et la pertinence des indicateurs d'erreurs pour les applications de véhicules à conduite automatisée. Elle a également permis de quantifier l'apport d'un système de perception multimodal dans le but d'obtenir une localisation précise.

Sur la base de ces résultats, une deuxième étude a été conduite portant sur l'analyse de l'intégrité du système de perception multimodale pour la localisation. Les concepts d'intégrité pour la localisation ont été amplement étudiés dans le domaine de l'aviation. La transposition de ces concepts vers le domaine applicatif des systèmes de transports terrestres définit toute une nouvelle frontière à franchir. L'adoption de certains concepts s'est avérée pertinente, néanmoins, l'intégrité de sources d'information relativement récentes comme la télémétrie LiDAR et les SIG 3D dans le processus de localisation sont peu abordés dans l'état de l'art.

Les estimations issues d'un système de localisation peuvent être entachées d'erreurs provenant principalement des sources d'information ou du processus d'estimation (e.g. fusion de données, inférence statistique). Par conséquence, elles sont sujettes à une incertitude. L'utilisation d'une perception

redondante et/ou multimodale est une approche permettant de détecter et de mitiger l'impact des erreurs des sources dans les systèmes de localisation.

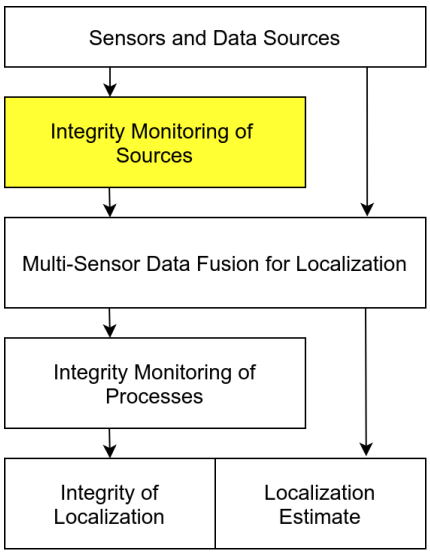


Figure 2.14 – Aperçu fonctionnel de l'analyse d'intégrité pour un système de localisation multimodale.

Par exemple, les systèmes EGNOS [11] et WAAS [12] exploitent la redondance de sources GNSS dans le but d'estimer l'intégrité des informations de positionnement. En France, des travaux émanant de l'Université Gustave Eiffel se sont souvent intéressés à l'intégrité de systèmes de localisation exploitant des sources GNSS, SIG et de l'odométrie [52, 89, 105]. Le laboratoire Heudisyc à l'Université de Technologie de Compiègne en collaboration avec l'industriel Renault [49, 66, 67] a mené plusieurs études portant sur l'intégrité. Dans ces travaux, il a été considéré l'utilisation de systèmes de localisation GNSS couplés à des SIG ainsi qu'une analyse temporelle des trajectoires répétitives. Un facteur commun des travaux de la littérature est l'étude de l'intégrité portant principalement sur le processus de localisation ou sur une source d'information du type GNSS. Peu de travaux de la littérature adressent l'analyse d'intégrité des sources pour les systèmes de localisation multimodale.

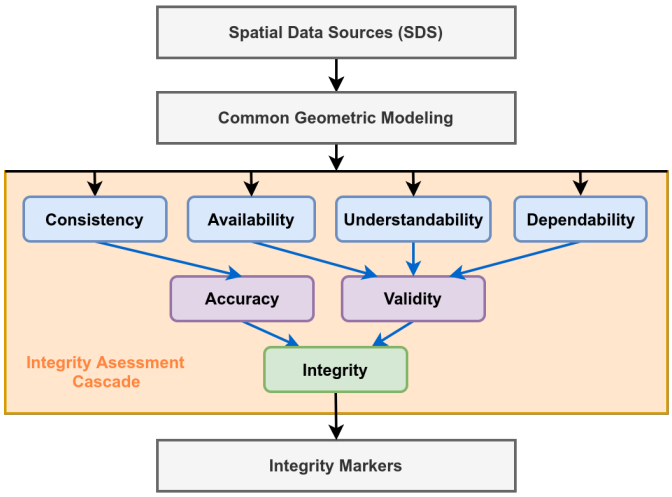


Figure 2.15 – Diagramme relationnel des critères pour l'évaluation des propriétés de l'information issues des sources en vue de la qualification de l'intégrité.

11. *European Geostationary Navigation Overlay Service*
12. *Wide Area Augmentation System*

Afin de statuer sur l'intégrité globale d'un système multimodal de localisation, il est nécessaire d'évaluer l'intégrité des sources de données et des processus de fusion qui y ont lieu comme l'illustre la Fig. 2.14.

J'ai décidé de poursuivre mes recherches sur l'évaluation de l'intégrité des sources dans un système de localisation multimodale. La méthode proposée se base sur l'identification de critères (dénommés en anglais *enablers*) permettant la caractérisation de deux propriétés de l'information : i) la précision et ii) la validité des informations issues des sources. Ces propriétés sont étroitement associées à la notion d'intégrité de l'information pour la localisation. Elles sont modélisées à l'expérimentale au travers de marqueurs d'intégrité. La synthèse de la méthodologie proposée est illustrée dans la Fig. 2.15.

Le diagramme de la Fig. 2.15 considère un ensemble de sources d'informations spatiales en entrée du système. Les informations de sources sont par la suite définies dans un espace de représentation commune où l'analyse d'intégrité est opérée. La **consistance** de l'information est estimée par l'intercorrélation des modèles géométriques obtenus. Une combinaison des estimations de la consistance des sources permet par la suite d'estimer la précision de chaque source. La **disponibilité**, la **compréhensibilité** et la **fiabilité** associée à des informations *a priori* sont également évaluées. Un mécanisme de prise de décision a été mis en œuvre afin de statuer sur la validité des données compte tenu de l'application et les modalités de perception disponibles dans le système. Les estimations de précision et les indicateurs de validité des sources sont enfin combinés pour obtenir des attributs d'intégrité pour chaque source.

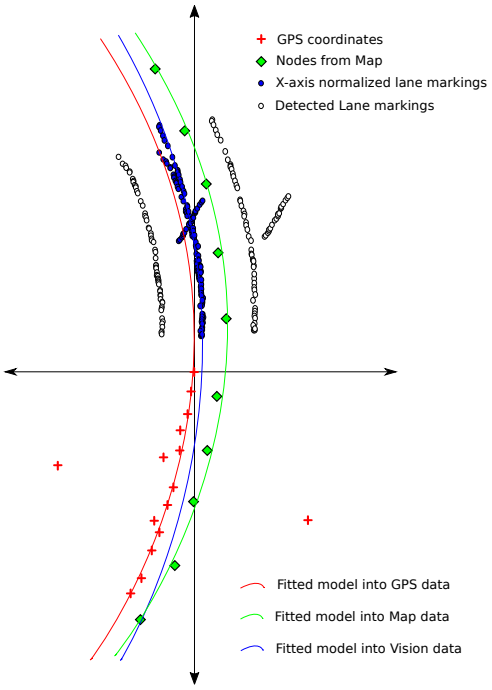


Figure 2.16 – Modélisation de données issues des sources de données dans un environnement routier. La figure illustre la trajectoire obtenue par la localisation GNSS en rouge, la géométrie observée par le marquage routier (vision monoculaire) en bleue et le segment routier issue d'un SIG en vert.

Les recherches conduites m'ont amené à étudier et valider expérimentalement la méthodologie proposée (voir Fig. 2.15) sous la forme de deux prototypes avec des représentations spatiales de l'environnement distinctes. L'analyse de résultats issus des prototypes a été réalisé en utilisant les données de référence KITTI.

Une première preuve du concept a été développée en utilisant un modèle quadratique polynomial représentant le segment routier où évolue le véhicule. Dans cette représentation spatiale de l'environnement, trois sources d'information ont été évaluées : la trajectoire obtenue par la localisation GNSS, la géométrie observée par le marquage routier (vision monoculaire) et le segment routier issue d'un SIG. La consistance des sources a été estimée par l'intercorrélation paramétrique des modèles polynomiaux. Un indicateur d'intégrité, dénommé poids d'intégrité, est obtenu pour chaque source sur la base des estimations de consistance. La disponibilité,la compréhensibilité et la fiabilité de l'information ont été représentées par 5 indicateurs binaires dénommés : prédicteurs de défauts et de faisabilité.

La Fig. 2.16 illustre les données du système de localisation ainsi que les modèles polynomiaux permettant l'évaluation de la consistance des sources. À l'origine du repère de la figure se situe le véhicule et la trajectoire obtenue par la localisation GNSS est illustrée en rouge. La ligne centrale de la voie est déterminée à l'aide de la détection des marquages routiers dans le flot vidéo, elle est représentée en

bleue. Les données issues du SIG sont représentées en vert. Grâce à la Fig. 2.16, nous pouvons constater la complémentarité et la redondance de l'information spatiale assurée par le système de perception. Les données GNSS représentent la trajectoire parcourue, les données issues de la vision représentent la trajectoire devant le véhicule et le SIG représente la géométrie globale de la trajectoire.

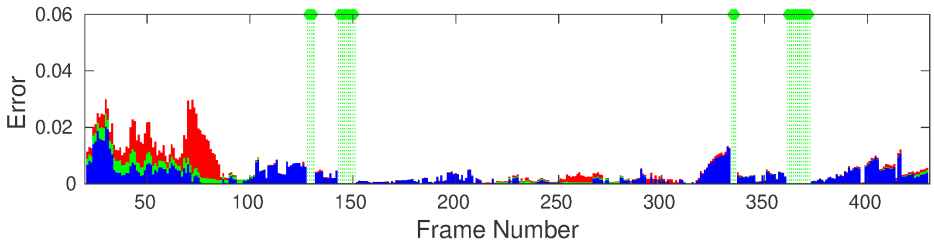
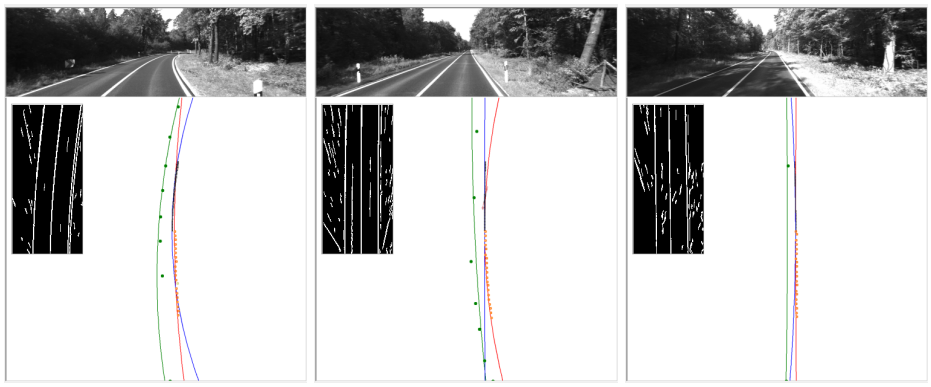


Figure 2.17 – Évolution temporel des indicateurs d'erreur (inconsistance) de sources dans une séquence routière - KITTI 2011_09_26_drive_0028. Indicateur d'erreur (inconsistance) des données GNSS en rouge, des données issues du SIG en vert et des données du système de vision en bleue. Les barres verticales vertes représentent l'activation du prédicteur de défaut et faisabilité du SIG.

La cohérence des estimations de consistance des sources a été validée comme l'illustre la Fig. 2.17. Cette figure trace l'évolution du score de l'inconsistance des sources (i.e. erreur) dans la séquence routière KITTI identifiée 2011_09_26_drive_0028. Les sources GNSS, vision et SIG sont représentées en rouge, bleue et vert respectivement. Les barres verticales vertes représentent l'activation du prédicteur de défaut et faisabilité du SIG. Ce prédicteur isole les situations où le SIG donne une représentation spatiale insuffisante (e.g. le nombre de nœuds du segment routier ne peut pas être approché par le modèle polynomial).



(a) Scène du frame 28 (b) Scène du frame 77 (c) Scène du frame 365

Figure 2.18 – Comparaison de modèles quadratiques polynomiaux obtenus par source. Scènes de la séquence routière KITTI 2011_09_26_drive_0028.

La séquence représentée par la Fig. 2.17 illustre une situation où globalement les données des trois sources sont consistantes. Néanmoins, les indicateurs de consistance proposés sont sensibles à des situations transitoires telles que l'entrée ou la sortie d'un virage. Ces situations sont détaillées dans la Fig. 2.18. Par exemple, l'image de la Fig. 2.18.a correspond à l'itération 28 où la courbure de la trajectoire obtenue par les données issue de la vision est plus importante que celle observée dans les données GNSS et SIG lors de l'entrée dans un virage, l'indicateur estime une faible consistance des données de la source vision. Dans l'itération 77, une situation équivalente se présente en sortie d'un virage où les données GNSS ont une faible consistance face à celles de la vision et le SIG (voir

Fig. 2.18.b). L'activation d'un prédicteur de défaut et faisabilité dans l'itération 365 confirme une très pauvre représentation spatiale du SIG comme illustré dans les Fig. 2.18.c et 2.17.

L'analyse expérimentale de ce premier prototype a confirmé sa pertinence compte tenu des scénarios et des situations étudiées. La solution implantée permet la vérification (monitoring) de l'intégrité d'un système de localisation multimodale avec 3 sources de données. La consistance de données est évaluée à travers un modèle paramétrique de la structure routière.

En revanche, les résultats de ces expériences ont également permis d'identifier les limites du prototype dans la gestion de situations complexes. Ces limitations sont induites par les contraintes qu'impose l'utilisation du modèle quadratique polynomial. Par exemple, ce type de modèle ne permet pas la représentation de virages abruptes ou des scénarios avec un très faible déplacement du véhicule. En effet, l'utilisation d'une seule est unique primitive (ici géométrie de la route) commune aux modalités facilite et réduit la complexité du processus de vérification d'intégrité des sources. Néanmoins, l'analyse d'intégrité est limitée aux situations où cette caractéristique est observable par l'ensemble des sources.

Un deuxième axe d'amélioration concerne l'extraction robuste des marquages routières nécessaires à l'analyse de consistance. Cette étape reste sensible aux conditions d'éclairage. Ainsi, une présence importante d'ombres peu nuire à l'analyse de consistance de la source de données visuelles.

L'étude mené a été étendue à une nouvelle preuve du concept de l'analyse d'intégrité de sources multimodales basée sous une représentation de l'environnement du type grille sémantique. Ce prototype surmonte les limitations révélées par l'utilisation d'un modèle paramétrique et introduit l'estimation de niveaux de protection en s'inspirant des concepts existants dans l'état de l'art.

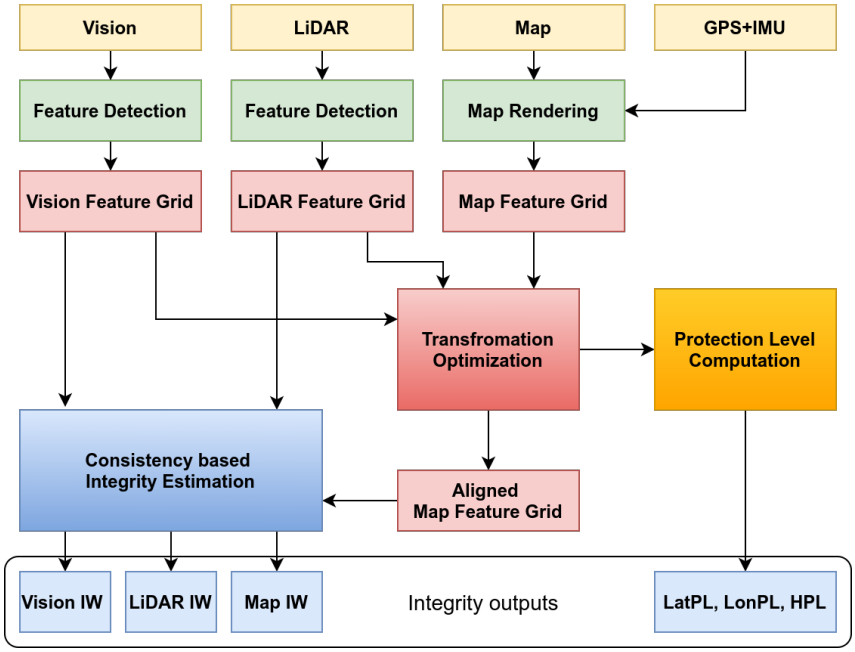


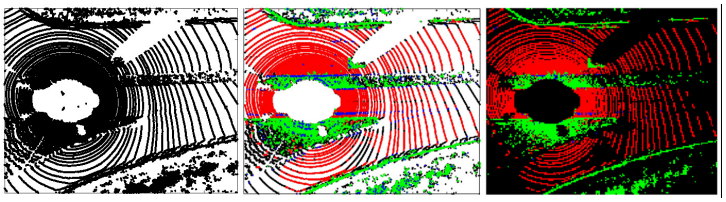
Figure 2.19 – Système de localisation multimodale avec analyse d'intégrité : Fusion d'information issue de la vision, la télémétrie LiDAR, d'un système de localisation GNSS/Inertielle et d'un SIG.

La Fig. 2.19 illustre la structure globale du système de perception multimodale proposé. Ce système qualifie le niveau d'intégrité de sources en évaluant les informations des capteurs dans une grille sémantique de l'environnement. L'utilisation d'un modèle du type grille d'occupation enrichie par des informations sémantiques permet d'évaluer l'intercorrélation des informations. Un processus d'opti-

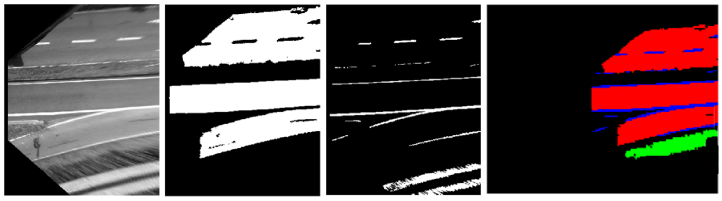
misation maximisant la cohérence des informations des sources permet d'estimer des niveaux de protection (e.g. LatPL, LonPL et HPL). Cet algorithme estime également en sortie l'intégrité de chaque source avec un indicateur normalisé dénommé poids d'intégrité (e.g. Vision, LiDAR et Map Integrity Weights).

La stratégie dédiée à l'analyse de l'intégrité des informations issues de sources repose sur la représentation de l'environnement du véhicule, par des primitives sémantiques dans une grille d'occupation. Les primitives identifiées sont observables par l'ensemble des sources à l'exception du récepteur GNSS-IMU, où les informations en sortie sont proprioceptives. Par cette raison, les sources SIG et GNSS-IMU ont été étroitement couplées lors de l'analyse d'intégrité comme l'illustre la Fig. 2.19.

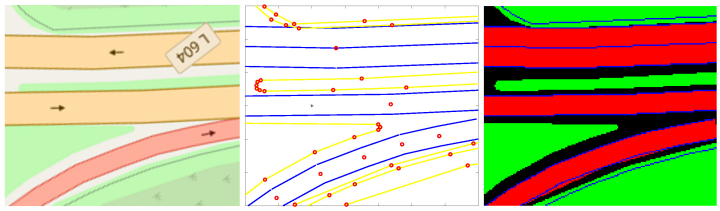
La description sémantique de la scène comprend 4 classes : les données appartenant à la route (en rouge), au marquage routier (en bleue), aux surfaces voisines (en vert) telles que les trottoirs et les espaces verts et les données non-classifiées (en noir). La classification est opérée dans les données de chaque source au travers d'un traitement indépendant détaillés dans [96]. Les résultats obtenus à l'issue du pré-traitement de données du capteur LiDAR, de la vision monoculaire et du SIG+GNSS-IMU sont illustrés respectivement dans la Fig. 2.20 a, b et c. Les grilles d'occupation sémantiques correspondantes pour une même scène routière sont présentées à droite de la Fig. 2.20.



(a) Gauche : Données LiDAR dans le voisinage du véhicule. Centre : Classification de données : route (rouge), marquage routier (bleu), autre surface (vert) et donnée non-classifiée (noir). Droite : Grille sémantique issue des données LiDAR.



(b) De gauche à droite : Projection d'image sur le plan routier (IPM), segmentation de la route, segmentation du marquage routier et grille sémantique des données Vision



(c) Gauche : SIG obtenu par Open Street Maps. Centre : Information sémantique du SIG. Droite : Grille sémantique des données du SIG

Figure 2.20 – Modélisation des données issues des sources d'un système de localisation multimodale : Classification de données route (en rouge), marquage routier (en bleue), autre surface (en vert) et les données non classifiées (en noir).

Les cellules de la grille sémantique sont associées à un label et à une incertitude individuelle. Cette dernière est modélisée par un score normalisé inversement promotionnel à la distance mesurée comme il a été proposé dans [81] et il a été appliqué pour les sources de données LiDAR et vision. En ce qui concerne la source SIG, un modèle de score uniforme a été utilisé. En effet, le modèle d'incertitude du SIG pourrait être amélioré par une propagation des incertitudes de localisation GNSS+IMU et par la prise en compte des incertitudes associées à la création du SIG. Néanmoins, cela n'a pas pu être abordé lors de cette étude.

Grâce à une représentation des données commune pour l'ensemble d'informations multimodales, la mesure d'intégrité est obtenue comme étant une estimation globale de la précision et de la consistance des sources de données [41]. Il est important de souligner que le concept de précision est associé à la justesse des données valides, et la consistance est une mesure de la cohérence entre les sources.

Ainsi, l'ensemble de N sources dénoté $\mathcal{S} = \{s_1, s_2, s_3, \cdots, s_N\}$ associé aux grilles sémantiques, s_pFG est exploité pour l'estimation des poids d'intégrité. Une cellule, c_k, de la grille s_pFG portant un label LB_x est définie consistante s'il existe au moins une autre cellule avec le même label dans une fenêtre de voisinage de 3×3 dans la grille s_qFG. Cette vérification de consistance dénotée par l'opérateur f_m est appliquée à toutes les cellules de la grille :

$$f_m(s_pFG, s_qFG) = \frac{N_m^{s_pFG}}{N_T^{s_pFG}} \tag{2.6}$$

où $N_m^{s_pFG}$ est le nombre de cellules consistantes de la grille s_pFG et $N_T^{s_pFG}$ est le nombre total de cellules labellisées de la grille s_pFG (les cellules de la grille non-classifiées sont exclues). Après l'estimation de consistance entre toutes les sources (ici 6 combinaisons), le poids d'intégrité associé à une source de données est estimé comme suit :

$$IW_p = \frac{\sum\limits_{\forall q,\, p \neq q} f_m(s_pFG, s_qFG)}{\sum\limits_{\forall p,q\, p \neq q} f_m(s_pFG, s_qFG)} \tag{2.7}$$

L'analyse d'intégrité des données issues de sources dépend également de l'alignement des grilles d'occupation sémantiques. Compte tenu des sources considérées dans cette étude, il faut noter que des incertitudes dans l'étalonnage entre les capteurs Vision-LiDAR peut introduire des erreurs qui ont été considérées négligeables dans cette étude. En revanche, les erreurs d'alignement sont substantiellement plus importantes par rapport à la source SIG+GNSS-IMU. Ainsi, une procédure d'alignement estime le décalage spatial entre l'ensemble de grilles en opérant une maximisation de la cohérence spatiale des primitives sémantiques. L'optimisation est formalisée par l'expression :

$$t\,(MFG, x, y, \theta) = R(\theta) * MFG + T(x, y) \tag{2.8}$$

où l'opérateur $t(\cdot)$ dénote la transformation rigide en 2D de l'ensemble de coordonnées des cellules de la grille de primitives MFG, $R(\theta)$ est une matrice de rotation 2D et T est un vecteur de translation 2D. MFG représente la grille de primitives sémantiques issue de la source SIG+GNSS-IMU. L'algorithme estime, à travers une optimisation séquentielle basée sur un filtre à particules, la transformation $(x^*, y^*, \theta^*)_{s_pFG}$ de la MFG par rapport à chacune des grilles sémantiques.

La distribution des particules du processus d'alignement de toutes les sources est approchée à une distribution Gaussienne. Cette approximation permet l'estimation des bornes tels que les niveaux de

protection latérale ($LatPL$), et longitudinale ($LonPL$) ainsi que le niveau de protection horizontal (HPL). Ces bornes sont estimées comme suit :

$$LatPL = K_Y \sqrt{\left(\sigma_{CY}^2 + \sigma_{LY}^2\right)/2} \tag{2.9}$$

$$LonPL = K_X \sqrt{\left(\sigma_{CX}^2 + \sigma_{LX}^2\right)/2} \tag{2.10}$$

$$HPL = K_H \sqrt{\left(\sigma_{CX}^2 + \sigma_{CY}^2 + \sigma_{LX}^2 + \sigma_{LY}^2\right)/4} \tag{2.11}$$

où σ_{CX}^2 et σ_{CY}^2 sont les variances latérales et longitudinales estimées à partir de la distribution des particules du filtre optimisant l'alignement des grilles sémantiques SIG+GNSS-IMU et Vision. σ_{LX}^2 et σ_{LY}^2 sont les variances latérales et longitudinales correspondantes à l'alignement des grilles sémantiques SIG+GNSS-IMU et LiDAR. K_X, K_Y et K_H sont des constantes choisis en considérant une limite de probabilité limitée 2σ.

Les résultats expérimentaux obtenus lors de la validation de ce deuxième prototype sur la base de données KITTI ont confirmé la pertinence des estimations d'intégrité du système de perception pour les poids d'intégrité des sources et pour les bornes des niveaux de protection. Une description plus riche de l'environnement permet ainsi une évaluation plus fidèle de l'intégrité des sources de données compte tenu de la variabilité des scènes propres des applications automobiles.

Ces recherches ont également permis de comparer les bornes de protection inférées par l'approche proposée pour le SPME étudié à celles estimées pour les systèmes GNSS et aux prérequis proposées par [94] pour les voitures à conduite automatisée. La Fig. 2.21 illustre l'évolution temporelle des bornes. Les courbes en tracé continu correspondent aux bornées inférées par la méthode proposée, celles en tracé discontinu représentent les estimations de référence.

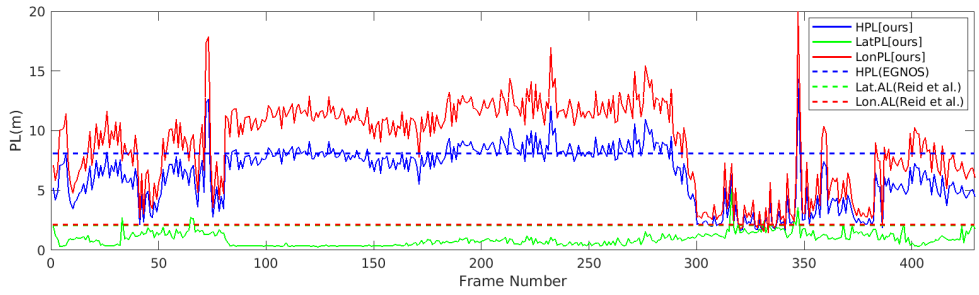


Figure 2.21 – Comparaison des bornes des protection (PL, Protection Levels)

On peut constater dans la Fig. 2.21 que la méthode proposée estime un haut niveau d'intégrité des données permettant d'assurer une borne de protection latérale rapprochée (voir courbe continue en vert). Cette borne confirme la capacité du SMPE à répondre aux prérequis estimés pour un système de conduite automatisé en milieu urbain. En revanche, la borne de protection longitudinale subie des fortes variations. Les expériences ont confirmé que ces variations sont corrélées à une pauvre configuration géométrique de la scène. En effet, les sections routières du type couloir présentent une incertitude longitudinale. Ces incertitudes peuvent être rapidement réduites par la présence de virages ou des éléments de structure qui lèvent ces ambiguïtés dans les données des capteurs (voir réduction au frame 300 de la Fig. 2.21).

A l'issue de ces travaux de recherches, j'ai pu proposer et étudier une méthodologie portant sur l'évaluation de l'intégrité des informations des sources d'un SPME dédié à la localisation. Cette méthodologie vérifie deux propriétés de l'information associées à l'intégrité : cohérence et précision spatiale. Elle a été déclinée sous deux prototypes et validé à l'expérimentale au travers de bases de données automobiles.

On constate dans l'état de l'art que l'évaluation de l'intégrité d'un système de perception a une forte dépendance aux caractéristiques intrinsèques des sources de données. Afin d'établir une certaine invariance aux sources, nous avons favorisé l'utilisation de primitives lors de l'évaluation d'intégrité.

L'utilisation d'une primitive polynomiale, en accord aux environnements structurés propres des applications pour les véhicules, s'est révélée pertinente et adéquate pour des sources de type vision, GNSS et SIG. Elle comporte également une faible complexité en adéquation aux applications embarquées. En revanche, ce type de primitive peut s'avérer partiellement limitée dans des environnements urbains par la sensibilité des sources aux conditions d'exposition et par la résolution des SIG.

En ce qui concerne les primitives sémantiques, les recherches ont permis d'établir une amélioration dans l'évaluation de l'intégrité des sources du SPME. La caractérisation de données des sources sous trois labels sémantiques a élargi le spectre fonctionnel de l'algorithme en surmontant les limites identifiées pour le modèle polynomial. Certes, la représentation spatiale des primitives sémantiques sous une grille d'occupation augmente fortement la complexité de l'approche mais ouvre, en revanche, la possibilité d'estimer des niveaux de protection latérale et longitudinale. Ces niveaux de protection sont d'une importance capitale pour les systèmes de conduite automatisée [94].

Les perspectives de ces travaux s'orientent vers l'introduction des critères d'intégrité temporelle des sources tout en exploitant une représentation de grille sémantique. Les concepts d'intégrité temporelle ont émergé dans [66].

L'extension de ces travaux devra assurer des efforts envers une généralisation de critères (invariance par rapport aux sources) et des primitives pour l'analyse d'intégrité des données. Il devra également tenir compte des écarts dans le volume d'information entre les sources. Ces écarts pourraient dans certaines conditions biaiser les indicateurs d'intégrité.

2.4.1.2 Contributions à la perception de l'environnement par des systèmes multicapteurs

La perception de l'environnement constitue un verrou scientifique avec un large spectre impliquant des multiples problématiques complexes en particulier celles s'adressant à la détection d'objets, à leur suivi temporel, à la fusion d'estimations, à la prise en compte du contexte et à la compréhension et/ou identification de comportements (voir Fig. 2.22). Ce verrou a pour objet l'inférence de la structure et de la dynamique d'une scène observée à partir d'un système de perception mobile en temps-réel. À l'issue, ces informations facilitent la compréhension et l'analyse de scènes dynamiques [44] et permettent ainsi la prise de décision pour de systèmes d'assistance à la conduite ou des systèmes dédiées à la navigation autonome.

Nombreux sont les travaux scientifiques s'adressant aux problématiques associées à la perception de l'environnement [85]. Ces travaux font appel à l'utilisation de données de multiples capteurs extéroceptifs tels que les systèmes de vision (vision monoculaire, stéréoscopique, omnidirectionnelle, RGB-D et évènementielle), le Radar, la télémétrie LiDAR et dans une moindre proportion les capteurs infrarouges. Il faut noter que les systèmes de localisation ne sont pas pertinents pour l'inférence de l'environnement, mais leur utilisation donnant accès à la position du porteur dans ce contexte applicatif, facilite l'association des informations contextuelles et/ou sémantiques provenant des SIG.

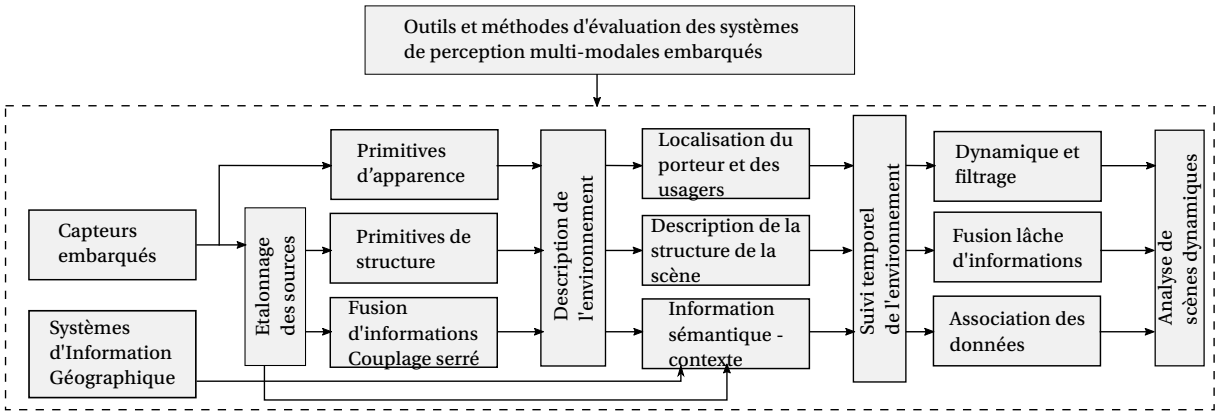


Figure 2.22 – Schéma fonctionnel de la perception de l'environnement pour les véhicules intelligents.

Les traitements des informations pourvues par les systèmes de perception embarqués se structure par les étapes suivantes : la détection d'objets ou des primitives, leur suivi temporel, la fusion d'informations et l'identification des objets tels que les usagers de la route mais aussi que celles des éléments de structure tels la chaussée, le trottoir et les marquages et les panneaux routiers. La composition et le référencement dans un repère commun de ces éléments décrit globalement l'environnement autour du véhicule en 4 dimensions (i.e. espace 3D et temps).

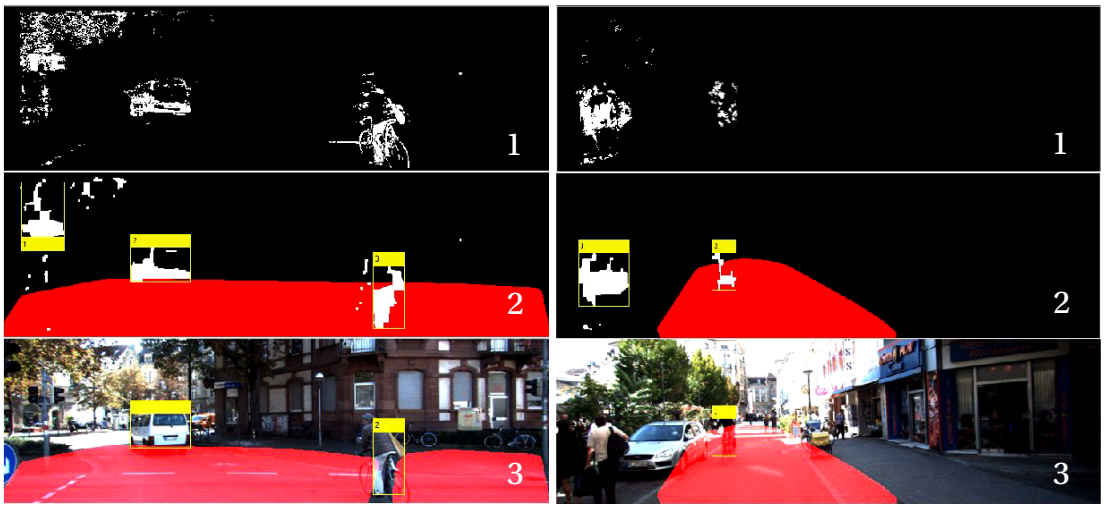
Les méthodes et les stratégies adoptées pour y parvenir sont très diverses et varient selon les technologies des capteurs. Dans l'objectif de construire un aperçu global des contributions que j'ai accompli au travers de mes recherches, elles seront synthétisées dans la suite de cette section, suivant trois volets **la détection, le suivi et la fusion d'informations et l'évaluation des systèmes de perception**. Il est important de mettre en évidence que l'évaluation et la comparaison des méthodes et des systèmes de perception est établie majoritairement sur des bases de données accessibles publiquement par des critères de performance adaptés à la problématique et aux scénarios applicatifs.

La détection

La détection d'objets peut être classifiée en trois grandes familles d'approches : i) les méthodes basées sur l'apparence [95], ii) les méthodes basées sur la structure [108] et iii) les méthodes combinant des primitives hybrides d'apparence et de structure [90, 109, 47].

Les recherches que j'ai entreprises ont été orientées, dans un premier temps, vers l'étude d'un système de perception multi-capteur de vision embarqué. Ces travaux ont été motivés par l'existence prédominante des recherches portant sur la vision stéréoscopique. Ces méthodes favorisent l'association géométriquement contrainte d'informations élémentaires de l'image (ici association des pixels suivant la géométrie épipolaire) afin d'obtenir une perception 3D (carte de disparité ou nuage de points) où la détection d'objets est opérée. D'autres approches opèrent dans des espaces cumulatives où une segmentation de la scène peut être effectuée [103, 69]. Plus récemment avec l'émergence des approches basées sur l'apprentissage profond, les stratégies de détection ont fortement évoluées favorisant ainsi une détection par l'apparence [95].

Dans ce contexte, une stratégie de perception pour un système de vision stéréoscopique mobile a été étudiée en s'orientant vers une détection redondante d'objets (informations avec un haut niveau d'abstraction) sous un mode monoculaire. L'intégrité de ces détections est augmentée par les contraintes de stéréoscopie (ici U et V-disparité) et par un suivi temporellement à l'intérieur d'une zone d'intérêt définie par la chaussée.



(a) Exemple de détection dans la séquence 1. (b) Exemple de détection dans la séquence 2.

Figure 2.23 – Exemples de détection d'objets dynamiques par contraintes de structure en utilisant un système de vision monoculaire. L'image 1 est l'image de vraisemblance combinée obtenue par les contraintes épipolaires et trifocales. L'image 2 est le résultat de détection dans la zone d'intérêt (ici chaussée) et le regroupement et densification des régions par l'utilisation de blobs. L'image 3 est le résultat final incrusté dans les images de la séquence.

Les contributions méthodologiques pour la phase de détection du système proposé ont porté sur l'analyse du mouvement apparent des pixels dans l'image et la segmentation de ceux appartenant à des objets mobiles par le renforcement des contraintes de structure dénommées plan-parallaxe. Cette méthode s'applique à une séquence monoculaire identifiant dans un premier temps des régions dynamiques de l'image par une soustraction du fond de la scène s'opérant dans une fenêtre temporelle de n images. Dans les régions dynamiques identifiées, deux contraintes géométriques sont vérifiées : la contrainte épipolaire et la contrainte trifocale. Chaque contrainte permet l'identification des régions dynamiques de l'image respectant une cohérence structurelle au cours du temps et donne en sortie une image de vraisemblance au sens de la contrainte. À l'issue, les images de vraisemblance sont combinées et conditionnées afin d'obtenir une liste de régions incluant les objets mobiles à l'intérieur d'une zone d'intérêt.

La qualité des résultats de détection obtenus par cette méthode originale a été évaluée en utilisant la base de données KITTI sur des séquences se déroulant en milieu urbain. La Fig. 2.23 illustre à titre d'exemple les résultats dans deux images extraites des séquences vidéo particulièrement complexes par les conditions d'exposition changeantes. Les résultats de détection obtenus sont encourageants non seulement par la complémentarité du critère de structure utilisé vis-à-vis des méthodes basées sur l'apparence mais aussi parce que la méthodologie est exploitable sous un mode monoculaire.

Dans la Tab. 2.1, l'évaluation générale de performance de la méthode comporte le taux de détection, le taux de non détection et des fausses alarmes mais également par leur taux de détection redondante. Ce dernier critère fait référence à la fragmentation des objets détectés.

L'étude expérimentale de l'algorithme proposé a conduit à l'obtention un taux de détection d'objets mobiles en milieu urbain compris entre 50.08% et 74.64% dans une portée de 35m. Il est important de souligner que l'évaluation de la fonction est contenue dans une zone d'intérêt (ici la chaussée illustrée dans la Fig. 2.23 en rouge). Cela peut certainement limiter le taux de fausses alarmes (ex. mouvement de la végétation) et augmenter ainsi l'intégrité et la pertinence des résultats. Néanmoins,

elle peut induire également à des non-détections pour des objets se situant dans les limites de la région d'intérêt. Les détections redondantes sont causées principalement par des occlusions partielles.

	Taux de détection	Taux de non détection	Taux de fausses alarmes	Taux de détection redondante
Séquence 1	50.80%	49.20%	29.84%	3.66%
Séquence 2	74.64%	25.36%	6.69%	28.03%

Table 2.1 – Évaluation générale de la détection d'objets dynamiques par contraintes de structure en utilisant un système de vision monoculaire. Les séquences 1 et 2 correspondent aux bases de données dénommées 2011_09_26_drive_0005 et 2011_09_29_drive_0071.

Enfin, cette méthode se présente comme une modalité de perception opérant la détection d'objets par la dynamique et la structure de la scène observée. En revanche, elle est tributaire des d'erreurs d'estimation du flot optique (ici mouvement apparent semi-dense) et des estimations du mouvement du porteur nécessaires à l'obtention des contraintes multi-vues (épipolaire et trifocale) [16]. Les observations à l'issue de ces travaux m'ont conduit à m'intéresser à la détection d'éléments de structure de la scène portant une valeur sémantique. Ces éléments de structure ont un apport fondamental dans la définition du contexte de la perception et ils augmentent la pertinence des informations issues des pré-traitements.

La détection des primitives dans l'environnement routier (ex. la chaussée, le trottoir, les marquages et les panneaux routiers) sont d'une grande importance pour la perception et la compréhension de la scène. En effet, ces primitives comportent intrinsèquement une forte saillance facilitant leur détection avec peu d'informations. Les primitives peuvent également apporter des contraintes spatiales pertinentes à la localisation et à la prise de décisions pour la navigation où elles peuvent être associées à un nombre important d'informations contextuelles et sémantiques. Les travaux de l'état de l'art portant sur la détection des marquages routiers se situent parmi les premières thématiques abordées dans la recherche associée aux transports intelligents et à la conduite autonome [85, 64].

Les recherches menées m'ont permis de confronter deux approches méthodologiquement différentes pour la détection des marquages routiers. Néanmoins, les approches étudiées et proposées dans les travaux de thèse [68, 96] opèrent dans l'espace projective IPM obtenu par une transformation homographique du plan image vers le plan routier (transfert 2D vers 2D). Cette transformation introduit une hypothèse de plan routier et elle nécessite une connaissance précise du positionnement du système de vision par rapport au plan routier (position et orientation). Ces informations sont très souvent assumées connues ; néanmoins, sous certaines conditions d'utilisation (variations du tangage) elles peuvent changer dégradant sensiblement la qualité de la projection et les traitements qu'on pourrait y effectuer. Cela étant dit, l'utilisation du IPM simplifie les traitements et les modèles utilisés lors de la détection du marquage en éliminant les effets projectives des éléments présents sur le plan routier.

Je me suis ainsi intéressé aux notions sémantiques associées à la chaussée et à l'espace navigable. Ces notions ont été approfondies dans le cadre des travaux de thèse de Bihao Wang [73]. Dans ces travaux, l'exploitation conjointe des critères d'apparence et de structure ont été la stratégie de recherche favorisée.

L'approche algorithmique proposée [23] permet d'inférer la surface libre de la chaussée (i.e. exclusion des zones occultés par des objets) à partir d'un traitement monoculaire et stéréoscopique. La segmentation de la chaussée appliquée à une séquence monoculaire d'images opère une transformation de l'espace couleur vers un espace chromatiquement invariant aux changements de luminance. Dans

cet espace, l'influence des ombres est fortement atténuée et une classification binaire probabiliste (distribution uni-modale) est appliquée par rapport à la valeur chromatique associée à la chaussée. Cette dernière peut être obtenue par exemple grâce à un échantillonnage aléatoire du bas de l'image où la présence de la chaussée est assumée dominante.

Figure 2.24 – Résultats de détection de la chaussée par couplage de critères d'apparence et de structure dans le jeu de données KITTI-ROAD.

Les estimations obtenues ont été combinées à une segmentation de la chaussée sous la contrainte géométrique dénommée V-disparité obtenue par stéréoscopie. Leur fusion a été effectuée sur la base de fonctions de vraisemblance. Pour la détection monoculaire, la confiance de détection a été qualifié en relation au voisinage (contrainte spatiale). Pour la détection par V-disparité, la confiance a été évaluée par rapport à la déviation de la valeur médiane de disparité. Les résultats de cette stratégie de perception ont été évalués sous les critères de performance de précision et de rappel dans le jeu de données KITTI (voir Fig. 2.24).

URBAN - BEV space						
	F_{max}	AP	Prec.	Rec.	FPR	FNR
SPRAY [58]	86.33	90.88	86.75	85.91	7.55	14.09
BM	**82.32**	**68.95**	**76.15**	89.56	16.15	10.44
MonoBM	79.45	66.16	69.22	**93.23**	22.83	**6.77**
CNN [57]	78.92	79.14	76.25	81.79	14.67	18.21
BL [60]	75.61	79.72	68.93	83.73	21.73	16.27

Table 2.2 – Résultats d'évaluation de détection de la chaussée le jeu de données KITTI

La performance de la méthode dans ces variantes monoculaire et stéréo (MonoBM et BM) a atteint un niveau de rappel supérieur à ceux des méthodes de référence (BL) et à une méthode d'apprentissage basée sur des réseaux convolutionnelles (CNN) comme l'illustre la Tab. 2.2. Cette recherche m'a permis également de constater que la caractérisation de la confiance de détection des primitives permet une gestion efficace des cas d'usage complexes comme celles où la planéité de la chaussée est compromise ou celles où les images sont fortement saturées. Par ailleurs, les performances de cette méthode ont été indexées dans le benchmark publique de KITTI consultable sur le site de « KITTI Vision Benchmark Suite » lien.

Afin d'approfondir cette méthode de perception et de l'éteindre vers un système multi-capteur vision-LiDAR, j'ai poursuivi et étudié à l'expérimentale la stratégie illustrée dans la Fig. 2.25. Cette étude a permis d'introduire, lors de la détection de la chaussée, multiples contraintes spatiales à travers l'utilisation de la télémétrie LiDAR [24].

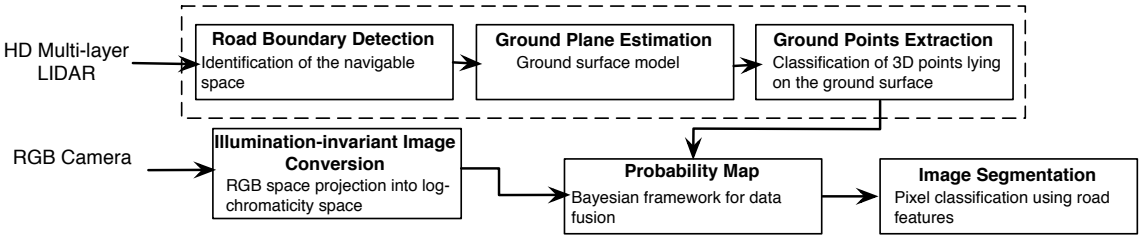


Figure 2.25 – Schéma fonctionnel d'un système de perception multi-modale Vision-LiDAR pour la détection de la chaussée.

En contraste aux estimations obtenues par la méthode BM [23] qui exploite les contraintes de structure issues de la V-disparité, la télémétrie LiDAR permet l'identification robuste du plan routier et l'estimation des limites de la chaussée. L'association de données LiDAR aux pixels de l'image par reprojection facilite également l'accès à une caractérisation plus fidèle des propriétés chromatiques associées à la chaussée. Le couplage étroit des modalités Vision-LiDAR permet d'améliorer la segmentation sur l'image et réduit considérablement les perturbations pouvant être générées par le marquage routier. L'évaluation de cette méthode a été effectuée sur la séquence de données du jeu données identifié KITTI-UMM et synthétisée dans la Tab. 2.3. On peut constater que la méthode Vision-LiDAR améliore significativement le rappel et retrouve une performance globale (Fmax) supérieure aux méthodes comparées.

Method	F_{max}	Precision	Recall
Vision-LiDAR [24]	**81.84 %**	72.66 %	**93.68 %**
BM [23]	80.64 %	80.35 %	80.93 %
BL [60]	76.17 %	65.02 %	91.95 %

Table 2.3 – Évaluation et comparaison des performances des méthodes multi-modales pour la détection de la chaussée.

Les contributions et les expériences scientifiques réalisées dans la phase de détection des systèmes de perception étudiés m'ont permis de prouver l'importance d'une détection multi-source exploitant et couplant étroitement des primitives d'apparence et de structure. Elles m'ont également permis de mettre en évidence l'impact positif des informations contextuelles (ex. pour l'environnement routier : marquage, chaussée) pour augmenter la pertinence des informations issues de la phase de détection.

La complexité et la variabilité des cas d'usage propres aux systèmes de transport intelligents exigent aux systèmes de perception la capacité d'une analyse temporelle de la scène. En effet, les informations relatives à l'apparence et celles spatiales doivent être accompagnées d'une dimension temporelle afin d'assurer une perception complète et dynamique de l'environnement.

Sur la base de ce postulat, j'ai poursuivi mes recherches sur les phases de suivi et de fusion d'informations pour les SPME.

Le suivi et la fusion d'informations

Les recherches entamées se situent dans la phase du suivi d'informations (voir Fig. 2.22) au sein des systèmes de perception. Je les ai orientées vers les problématiques scientifiques suivantes : i) l'analyse d'une scène par la dynamique observée, ii) la fusion d'informations de structure pour le suivi d'objets et iii) la modélisation et la prise en compte d'informations contextuelles pour le suivi d'objets. Dans ce cadre, je me suis intéressé à la conception de stratégies de détection et de suivi d'objets [100] (aussi dénotés par le sigle en anglais DATMO [13]) associant étroitement des primitives pertinentes à l'analyse de la scène à des techniques de filtrage probabiliste tels que le filtre de Kalman, le filtrage particulaire et le filtre Probability Hypothesis Density (PHD).

En ce qui concerne la problématique portant sur l'analyse d'une scène par la dynamique observée, j'ai mené, dans le cadre des travaux de thèse d'Hernan Gonzalez [91], l'étude, la formalisation et la validation expérimentale d'un algorithme effectuant le suivi des régions dynamiques de la scène. Ce suivi des régions dynamiques permet, à la convergence, l'inférence d'objets dynamiques dans la scène [7].

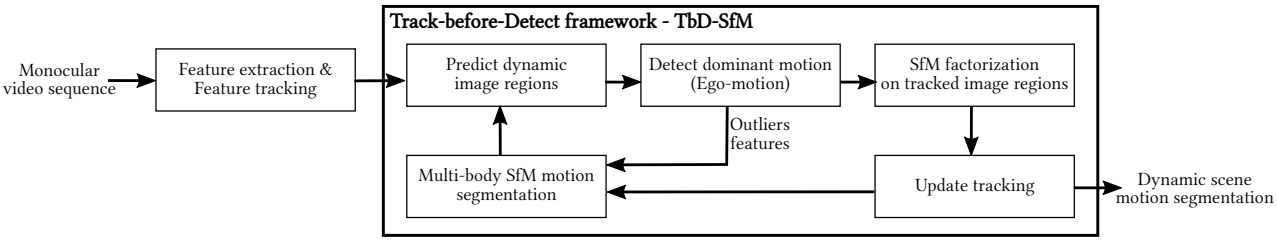


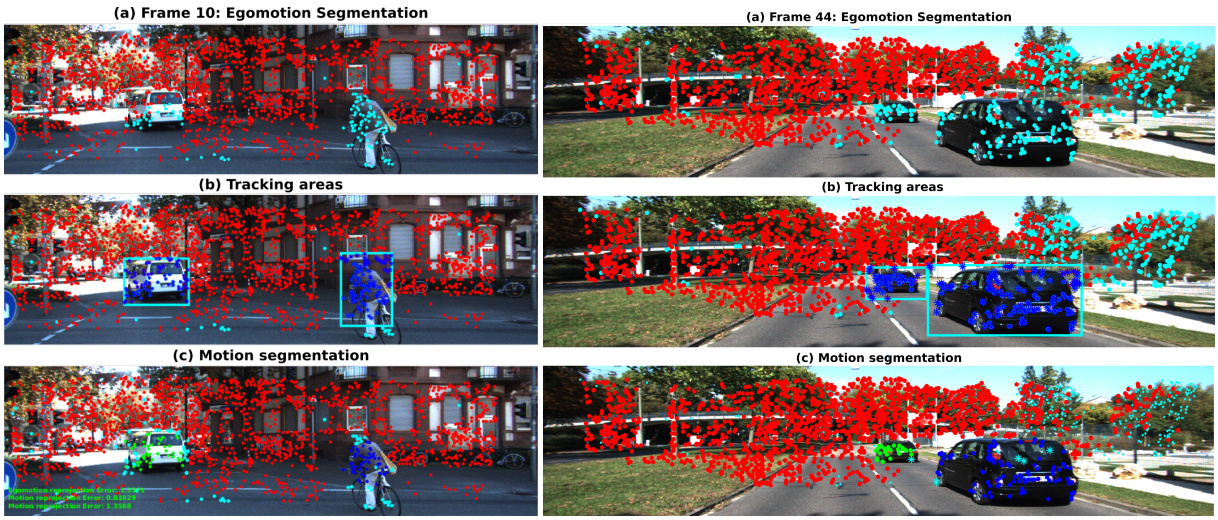
Figure 2.26 – Stratégie de perception pour le suivi avant détection d'objets dynamiques

Cette méthode s'inspire de [78] où une méthodologie permettant la factorisation des multiples mouvements a été introduite en adoptant une formulation mathématique propre à celle utilisée pour l'estimation de la structure par le mouvement. Cette méthode est dénommé en anglais *Multi-body Structure from Motion*. Ce formalisme étendu aux environnements dynamiques présente une haute complexité algorithmique par la recherche itérative et robuste opérée pour l'identification de mouvements indépendants. Ainsi, nous nous sommes intéressées à intégrer ce formalisme dans le cadre d'une approche effectuant un suivi de régions dynamiques des images pour aboutir à une détection d'objets dynamiques. La synthèse de cette approche est illustrée par la Fig. 2.26.

Les résultats expérimentaux obtenus à l'aide des séquences incluses dans la base de données KITTI ont permis de confirmer une réduction de la complexité de l'algorithme tout en améliorant les performances dans la détection et le suivi d'objets dynamiques. Cela a été possible grâce au découplage du mouvement dominant dans l'analyse du flot optique et à l'intégration du formalisme de suivi multi-objets basé sur un banc de filtres de Kalman. Comme l'illustre la Fig. 2.27, la méthode proposée débute l'analyse de la scène en découplant le mouvement dominant, en rouge, des autres primitives, bleue claire (voir la 1ère ligne de la séquence illustrée dans la Fig. 2.27). Par la suite, multiples hypothèses de mouvements indépendants sont identifiées afin d'effectuer une mise à jour de régions dynamiques de la scène. Le suivi de ces multiples régions permet l'inférence des objets dynamiques, en vert et bleu, et isole des primitives aberrantes, bleue claire. Ce dernier groupe de primitives est issue généralement des zones d'ambiguïté de l'image comme la chaussée et la végétation. La qualité des mouvements segmentés a été évaluée et comparée à l'état de l'art en termes d'erreurs de reprojection moyens et du pourcentage d'erreur de segmentation. Ces travaux ont pu atteindre des erreurs

13. *Detection and tracking of moving obstacles*

de reprojection moyennes par mouvement segmenté de l'ordre de seulement 1.53 px et des erreurs de segmentation de 1.45%.



(a) Résultats dans une séquence en milieu urbain. (b) Résultats dans une séquence en milieu semi-urbain.

Figure 2.27 – Résultats de détection et suivi d'objets par segmentation de mouvements dans une séquence en milieu urbain. En rouge le mouvement propre observé, en vert et bleue les mouvements indépendants segmentées et en bleue claire les observations identifiées comme aberrantes.

L'étude réalisée m'a permis d'approfondir et d'explorer une méthodologie algorithmique permettant d'introduire une dimension temporelle pour aboutir au travers du suivi des régions à la détection d'objets pour un système de perception monoculaire. Certes, la complexité algorithmique de l'approche est moindre mais elle n'est pas encore au point de son déploiement à l'embarqué avec des contraintes d'exécution réels. Néanmoins, ce point pourra sans doute constituer une perspective d'adéquation vers une architecture embarquée dédiée.

Dans [18], je décide également de conduire des recherches dans le suivi d'objets en m'intéressant à un système de perception stéréoscopique effectuant la détection d'objets par une application conjointe des contraintes de structure V- et de U-disparité. Dans ce cadre nous avons contribué avec cette étude à l'évaluation d'un système de suivi multi-objets basé sur un filtrage mono-hypothèse de filtres à particules opérant dans le plan image. Le vecteur d'état des pistes modélise les objets par des régions englobant leur projection sur le plan image caractérisés par la position et vitesse du centroïde, la largeur et la hauteur de la région ainsi que la disparité. Cette dernière a permis d'intégrer la prise en compte des effets projectives dans le modèle d'évolution des pistes.

Le système de perception proposé a été évalué en quantifiant le taux de fausses alarmes, de non-détections, de fragmentation et le pourcentage de recouvrement d'objets suivis par rapport à la référence comme l'illustre le Tab. 2.4.

Les résultats obtenus par le couplage de la détection stéréoscopique et le suivi d'objets s'est avéré efficace à la vue des critères d'évaluation avec un très faible pourcentage de non-détections et de fausses alarmes. Également, une détection précise d'objets a permis d'obtenir une faible fragmentation de pistes. Néanmoins, les systèmes de vision stéréoscopique sont intrinsèquement limités par rapport à la distance de détection assurant un fonctionnement fiable à partir d'environ 35m (ceci dépend sensiblement de l'entraxe des caméras).

Séquence KITTI	Fausses alarmes (%)	Non-détection (%)	Détection redondante (%)
Dataset 1	8.6%	4.0%	3.3%
Dataset 2	1.1%	3.3%	1.3%
Dateset 3	10.8%	8.1%	0%
Dataset 4	4.6%	3.1%	4.2%

Table 2.4 – Résultats d'évaluation d'une approche de détection par U-V disparité et d'un suivi d'objets par un filtre particulaire mono-hypothèse sur la base de données KITTI. Les séquences de test correspondent respectivement à une zone urbaine (Dataset 1), autoroutière (Dataset 2), rural (Dataset 3) et une avec trafic dense (Dataset 4).

Les expériences avec les systèmes de perception monoculaires et stéréoscopiques m'ont conduit à constater les limitations des stratégies de suivi mono-hypothèses et les difficultés inhérentes à l'association de données piste-objet. Dans le but d'essayer de surmonter ces problématiques j'ai orienté mes recherches vers d'autres sources d'information pouvant augmenter la performance du suivi d'objets et des méthodes de suivi facilitant l'association de données.

C'est ainsi que dans [21], j'ai mené une étude dans le cadre des travaux de thèse d'Egor Satarov qui a abouti à la définition d'une méthodologie algorithmique facilitant l'inclusion de données contextuelles dans le suivi d'objets. Pour ce faire, nous nous sommes basés sur un filtre Probability Hypothesis Density (PDH) pour la gestion des pistes afin de réduire le bruit des observations et combler les non-détections tout en utilisant un formalisme Bayésien avec une complexité algorithmique linéaire au nombre d'objets suivis.

La contribution de cette étude se situe dans l'étroite association des informations contextuelles au processus du suivi. Cela a été obtenu par l'introduction de particules dans le filtre PHD représentant la dynamique *a priori* des objets. Ces informations ont été déterminées à l'aide d'un SIG tenant compte des attributs associés à la structure routière (ici sens de circulation et limitation de vitesses). L'impact de cette contribution a été évalué suivant deux indicateurs de qualité du suivi, le taux de recouvrement des pistes et la continuité des associations piste-objet.

La mise en œuvre d'une preuve de concept a nécessité la modélisation d'une représentation des informations contextuelles issues du SIG sous la forme d'une carte de vecteurs en accord avec la circulation et les vitesses admises sur la structure routière. Il est important de signaler que la phase de détection dans le cadre de cette étude n'a pas été abordée. Ainsi, nous avons donc utilisé des détections qui ont été dégradées par l'ajout d'un bruit de mesure et par le manque aléatoire d'observations.

Les résultats des premières expériences ont conforté le fait que les informations contextuelles accélèrent la convergence du filtre et facilitent l'association piste-objets pour les cas d'usage où les objets suivis décrivent un mouvement en accord avec le contexte. Les situations où les objets observés décrivent un mouvement en opposition avec le contexte, la performance du filtre décroit en termes de continuité du suivi. Pour faire face à cette limitation, nous avons introduit un mécanisme adaptatif capable d'augmenter le ratio d'informations contextuelles du filtre PHD si les objets suivis sont en accord avec le contexte et limite ce ratio pour les objets en opposition.

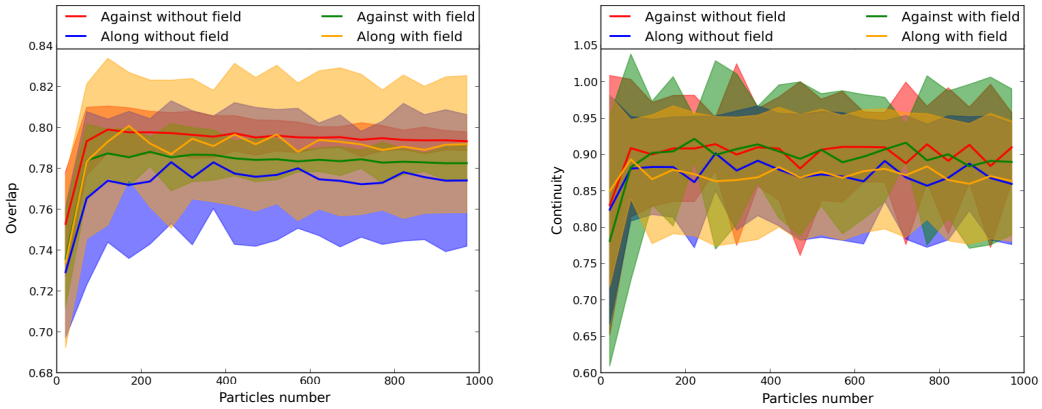


Figure 2.28 – Évaluation des performances du suivi d'objets assisté avec un ratio d'informations contextuelles fixe. En bleue et rouge, les performances pour un filtre sans information contextuelle avec des objets en accord et opposition aux mouvements *a priori* respectivement. En vert et jaune, les performances pour un filtre avec des informations contextuelles observant des objets en accord et opposition aux mouvements *a priori* respectivement.

Les observations à l'issue de cette étude ont démontré un fort potentiel de la méthode pour les applications sécuritaires associées aux véhicules à conduite automatisée. En effet, l'algorithme proposé permet la classification d'objets en opposition au contexte défini par un SIG par le suivi temporel. Ceci facilite l'identification de potentiels dangers tels que les véhicules en contre-sens ou des vulnérables en situations à risque. De plus, le formalisme et le mécanisme adaptatif proposé peuvent être transférés à d'autres filtres particulaires et exploiter d'autres modèles de distributions probabilistes dans un large spectre applicatif.

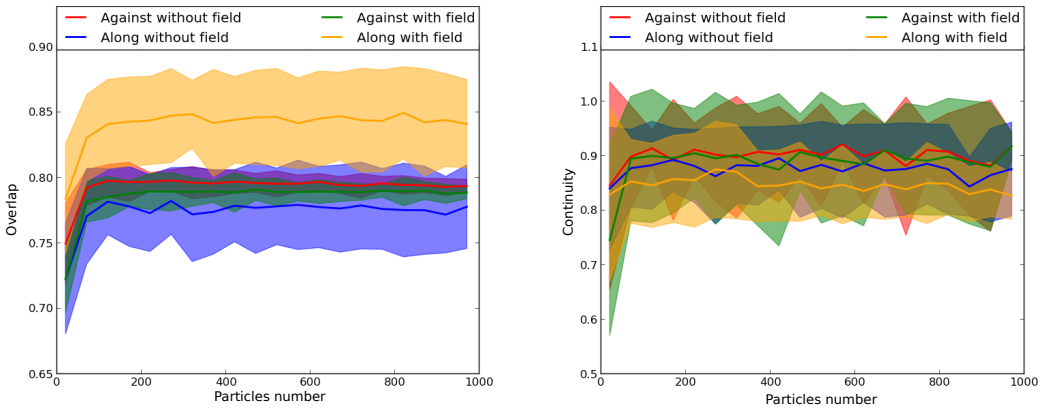


Figure 2.29 – Évaluation des performances du suivi d'objets assisté avec un ratio d'informations contextuelles adaptatif. En bleue et rouge, les performances pour un filtre sans information contextuelle avec des objets en accord et opposition aux mouvements *a priori* respectivement. En vert et jaune, les performances pour un filtre avec des informations contextuelles observant des objets en accord et opposition aux mouvements *a priori* respectivement.

Au travers des recherches conduites dans le suivi d'objets et la fusion de données, j'ai pu répondre avec mes contributions à des problématiques distinctes. Ces contributions ont été validées avec des systèmes de perception multi-modale embarqués abord de véhicules terrestres. La première permet d'exploiter le suivi temporel des régions dynamiques comme une primitive de détection d'objets dynamiques par vision monoculaire. La deuxième contribution couple étroitement une détection d'objets par U-V disparité à un suivi temporel par un filtre à particules mono-hypothèse afin d'améliorer la

performance du système. Enfin, un cadre méthodologique de suivi d'objets couplé à une source d'informations contextuelles a été proposé. Cette dernière contribution a démontré avoir un fort potentiel pour les applications sécuritaires dans le domaine des transports intelligents.

L'ensemble de ces études ont nécessité l'utilisation de données expérimentales et de mesures de référence afin de pouvoir qualifier et comparer la qualité et l'impact des contributions énoncées. Les bases de données, les moyens de référencement ainsi que les indicateurs de qualité non biaisés sont des éléments clés pour assurer une progression dans la production de connaissances au sein de la communauté scientifique. Dans la suite de cette section, une synthèse de mes contributions concernant la création d'outils et des méthodes d'évaluation pour les SPME est présentée.

Outils et méthodes d'évaluation des systèmes de perception multi-modale embarqués

Mon goût pour la recherche expérimentale et les compétences acquises dans ma carrière m'ont permis de réaliser également des contributions dans la production de bases de données et de méthodes d'évaluation des SPME. Ces contributions se positionnent sur trois niveaux : i) la détection et les modes d'acquisitions pour les systèmes de vision RGB-D, ii) les scénarios référencés pour le suivi multi-objets en extérieur et iii) une méthodologie pour l'évaluation et la qualification des systèmes de perception.

Figure 2.30 – Le véhicule expérimental robotisé pour la perception, la localisation et le contrôle automatique du laboratoire SATIE. Ce véhicule est pourvu des systèmes de vision RGB-D, des récepteurs GNSS RTK et différentiel ainsi qu'un LiDAR et un ordinateur dédié à l'enregistrement et la datation de données.

Dans l'état de l'art, nous pouvons constater la prolifération de bases de données intégrant de multiples capteurs dans des conditions d'usage diverses. Certaines sont plus accessibles que d'autres, présentent des environnements d'extérieur ou d'intérieur, d'autres sont plus riches par la diversité de modalités de perception. Un nombre réduit entre elles s'oriente vers des conditions d'utilisation très spécifiques (ex. la nuit, le brouillard, la pluie, saisons multiples) et inclue des moyens de référencement (i.e. vérité du terrain) non corrélés aux capteurs de perception (ex. étiquetage d'objets, localisation précise du porteur). D'autres disposent de dispositifs de comparaison de performance.

Comme l'illustre le Tab. 2.5, les bases de données sont aujourd'hui essentielles pour évaluer les systèmes de perception avant leur déploiement. Elles ont dans une certaine mesure démocratisé la recherche algorithmique dans ce domaine d'application en facilitant l'accès aux données de capteurs de dernière génération. Elles permettent également d'établir un étalon de performance entre les méthodes essentielles à la chaîne de perception (voir Fig. 2.22). La mise à disposition des bases de données profite aussi à l'industrie automobile avide des technologies de perception prometteuses.

Au sein du laboratoire SATIE, j'ai participé à la conception et je m'investis activement dans le fonctionnement d'une plateforme expérimentale de perception et de contrôle. Cette plateforme est implantée abord d'un véhicule électrique Renault ZOE (voir Fig. 2.30). Les équipements sensoriels que s'y trouvent m'ont permis l'étude d'algorithmes dans des conditions réelles ou contrôlées tout en maîtrisant le système de perception de bout à bout : du capteur en passant par le traitement, l'évaluation et l'analyse.

Base de données	Application	Capteurs	Référencement	Benchmark
Hopkins 155 [45]	Algorithmes de segmentation de mouvement	Vision	Flot optique	Non
KITTI [62]	Compréhension de scène	Vision, LiDAR, GNSS et IMU	Segmentation d'image, Flot 3D, étiquetage d'objets, boites englobantes (image et 3D)	Oui
MOTChallenge [70]	Suivi d'objets, en particulier des piétons	Vision	Étiquetage d'objets, boites englobantes	Oui
BDD-Nexar [82]	Compréhension de scènes et apprentissage profond	Vision, GNSS et IMU	Étiquetage d'objets	Non
nuScenes [97]	Compréhension de scènes	Vision, LiDAR, GNSS et IMU	Étiquetage d'objets, boites englobantes	Non
Waymo [101]	Compréhension de scènes dans diverses conditions d'usages	Vision, LiDAR	Segmentation de scène, étiquetage d'objets, boites englobantes	Oui
Audi A2D2 [99]	Compréhension de scène	Couverture 360° en Vision et LiDAR et données CAN	Segmentation de scène, étiquetage d'objets, boites englobantes	Non
CityScapes [75]	Compréhension de scènes dans diverses conditions d'usages et apprentissage profond	Vision	Segmentation de scène, étiquetage d'objets	Oui
Berkley DB100K [104]	Compréhension de scènes dans diverses conditions d'usages et apprentissage profond	Vision, GNSS et IMU	Segmentation de scène, étiquetage d'objets, espace navigable, marquage routier	Oui
ApolloScape [92]	Compréhension de scènes dans diverses conditions d'usages et apprentissage profond	Vision, LiDAR, GNSS et IMU	Segmentation d'image, Marquage routier, étiquetage d'objets, contour et boites englobantes (image et 3D)	Oui
Argoverse [106]	Compréhension de scènes dans diverses conditions d'usages (villes) et apprentissage profond	Vision, LiDAR, Localisation 6-dof, HD-GIS	Segmentation d'image, Marquage routier, étiquetage d'objets, contour et boites englobantes (image et 3D)	Oui
Honda H3D-HRI-US [88]	Compréhension de scènes, analyse de comportements	Vision, LiDAR, GNSS, IMU, CAN	Étiquetage suivant 4 niveaux de comportement)	Non
Oxford [84]	Compréhension de scènes	Vision, LiDAR, GNSS et IMU	Localisation	Non

Table 2.5 – Liste (non-exhaustive) de bases de données pour l'étude d'algorithmes de perception appliqués aux transport intelligents.

Ce fut le cas des travaux de thèse de Wenjie Lu [68] où la création d'une base de données a été réalisée. Les expériences menées ont comporté l'enregistrement asynchrone horodaté d'un flot vidéo monoculaire avec des données issues de trois récepteurs GNSS distincts en milieu urbain et semi-urbain. Les multiples récepteurs GNSS ont assuré l'utilisation d'un récepteur bas-coût pour les algorithmes de perception, d'un récepteur comme base comparative et d'un récepteur GNSS-RTK pour obtenir une trajectoire de référence.

Lors des recherches effectuées sur le suivi d'objets dans le cadre de la thèse d'Egor Satarov [79], j'ai constaté la nécessité de la création d'une base de données pour la validation d'algorithmes de suivi et d'association de données LiDAR-vision en extérieur, avec des moyens de référencement en position. C'est ainsi que j'ai entrepris la conception d'un système distribué d'enregistrement de données GNSS différentiel pour la localisation d'un groupe de piétons. Ces données ont été collectées et synchronisées à des données LiDAR, vision et GNSS RTK du véhicule expérimental.

Le post-traitement et la mise en forme de la base de données a nécessité l'étalonnage multi-capteurs, l'estimation de l'odométrie, le recalage de données dans un repère global (ENU) et la création des

fichiers BAGS[14]. Cela a abouti à 4 séquences d'une durée de 3 min environ incluant de multiples croisements entre les pistes, des occlusions et de situations avec une visibilité partielle des pistes. La Fig. 2.31 illustre un exemple du rendu des données où on peut constater à droite les ROIs obtenues par la projection de la localisation GNSS-D des pistes sur les images, la segmentation et la reprojection des données LiDAR sur les pistes imagées. À gauche, la Fig. 2.31 restitue le positionnement des cibles localisées à partir des balises GNSS différentielles et des données LiDAR dans le repère du véhicule expérimental.

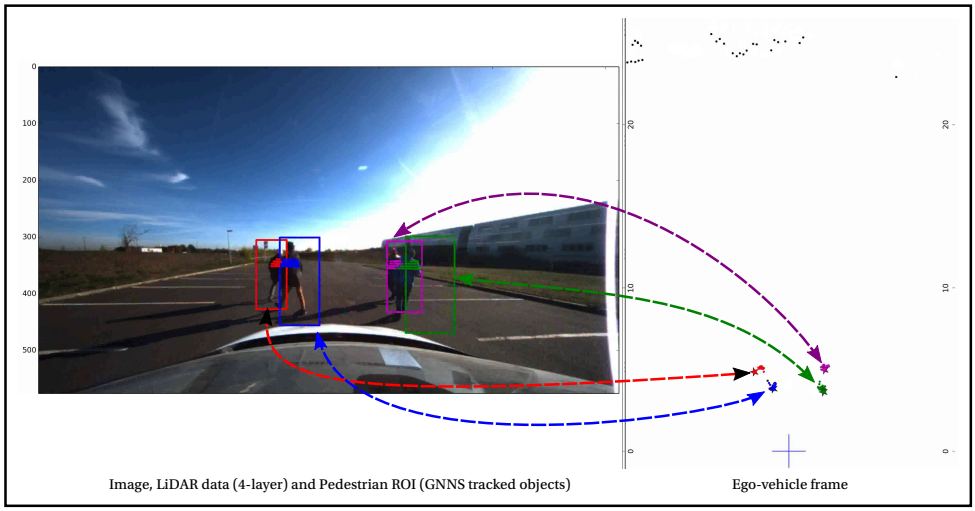


Figure 2.31 – Séquence de la base de données collectés pour la validation expérimentale d'algorithmes de suivi d'objets. À droite, les ROIs obtenues par la projection de la localisation GNSS-D des pistes sur les images, la segmentation et la reprojection des données LiDAR sur les pistes imagées. À gauche, le positionnement des cibles localisées à partir des balises GNSS-D et des données LiDAR dans le repère du véhicule expérimental.

Il est important de constater que le protocole expérimental établi lors de la création de cette base de données a permis l'extraction automatique d'imagettes nécessaires à l'entraînement des méthodes d'apprentissage pour la détection de piétons. Ceci grâce à la reprojection des données GNSS-D de chaque piéton sur la séquence vidéo. L'asynchronisme des capteurs peut néanmoins induire une perte de précision dans la projection des ROIs sur la séquence. La gestion de ce type d'asynchronisme peut être opérée par l'utilisation d'un filtre prédicteur.

Certaines études, par sa spécificité, peuvent nécessiter la récolte de données dans de conditions contrôlées. Par exemple, afin d'analyser l'impact des modes d'acquisitions possibles dans un système de vision RGB-D sur les algorithmes d'extraction de primitives et de leur suivi temporel. En effet, les modes d'acquisition des systèmes de perception (ex. fréquence échantillonnage, résolution, positionnement des capteurs) utilisés pour la création des bases de données est un élément qui s'impose. À ce jour, aucune base de données intègre des variations dans les paramètres du système multi-capteurs. Ainsi, une base de données d'un système de vision RGB-D en intérieur a été produite dans le but de supporter expérimentalement les recherches menées dans le cadre de travaux de thèse d'Imad El Bouazzaoui [111].

Dans cette base de données [2], trois séquences ont été produites en milieu intérieur (i.e. couloirs du laboratoire et parking). Les séquences produites donnent accès à des images infrarouges, couleur et à la carte de profondeur en mode passif (stéréoscopie) et en mode actif (stéréoscopie et projecteur de lumière structurée). La définition dans la base de données de la localisation, en intérieur, du système

14. Fichiers binaires respectant les structures de données utilisées par ROS.

de perception a suscité mon intérêt. Ce point a été relevé comme l'illustre l'image de la Fig. 2.32, par l'utilisation en post-traitement d'un algorithme de reconstruction par SfM [80].

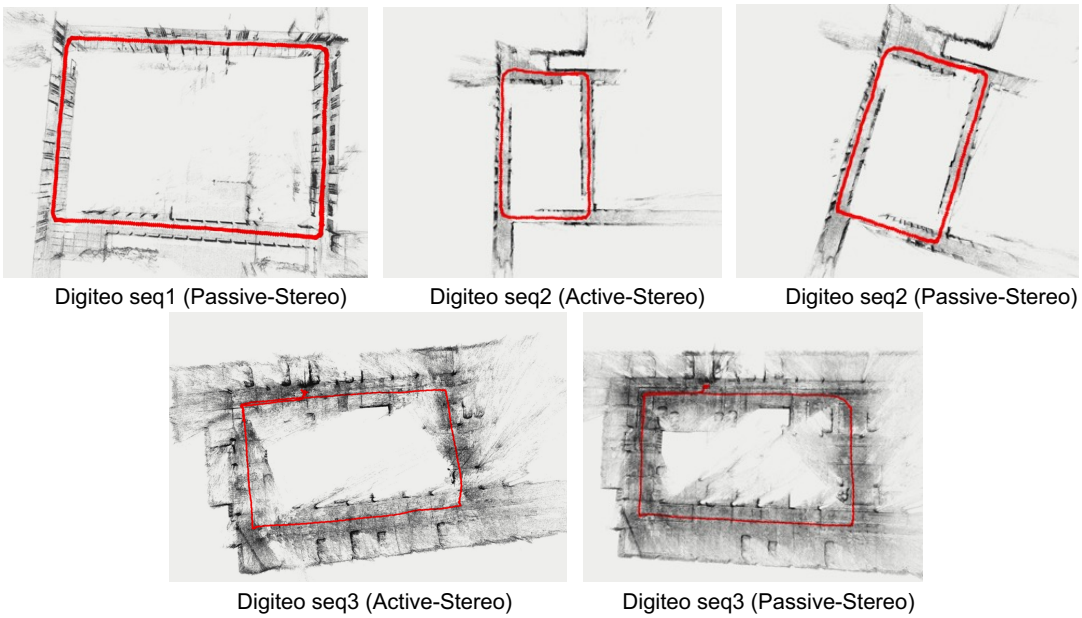


Figure 2.32 – Trajectoire de référence de la base de données [2] obtenue par post-traitement en utilisant un algorithme de reconstruction par SfM.

La vocation qualifiante des bases de données peut également être limitée par la variabilité des scénarios. Certes, il existe aujourd'hui des bases de données très larges, néanmoins, l'étude et la conception d'un système de perception nécessite la définition et l'identification de cas d'usages spécifiques comme dans [88]. La collecte de données de ces cas d'usage doit pouvoir s'effectuer dans des environnements contrôlés (ex. piste d'essais) assurant leur répétabilité et ne représentant aucun risque (ex. collision avec des usagers de la route).

Dans ce contexte, je me suis intéressé à la définition d'une méthodologie capable de qualifier automatiquement un système de perception de l'environnement (ici on restreint la notion d'environnement aux éléments statiques de la scène) dans un milieu contrôlé et connu. Ces travaux de recherche ont été conduits dans le cadre de la thèse de Rémi Defraiteur [107] en collaboration avec un industriel du secteur automobile.

La méthodologie proposée [14] réalise, dans un premier temps, le transfert des informations sémantiques (label des classe) contenues dans un SIG 3D de l'environnement vers une représentation volumétrique composée de voxels. Cette représentation volumétrique est par la suite utilisée comme une donnée de référence. Dans un deuxième temps, un algorithme de détection (ex. détection de chaussée, marquage ou des panneaux routiers) peut être automatiquement qualifié en associant ses estimations à la représentation volumétrique enrichie par la sémantique. En sortie, la méthodologie est capable d'estimer des indicateurs de performance de précision et de rappel sur la fonction de détection évalué dans le scénario étudié.

Ce protocole est répétable et invariant sous hypothèse que les éléments statiques cartographiés dans le SIG 3D de la scène soient fidèles à l'environnement d'essai et que la position du véhicule soit comprise dans une borne d'incertitude adéquate à la catégorie détectée. La première hypothèse est vérifiée lors de la génération de la représentation volumétrique par l'utilisation d'un capteur de télémétrie

LiDAR HD. En ce qui concerne le prérequis de localisation nous avons caractérisé les bornes admissibles selon les catégories comme l'illustre la Fig. 2.34.

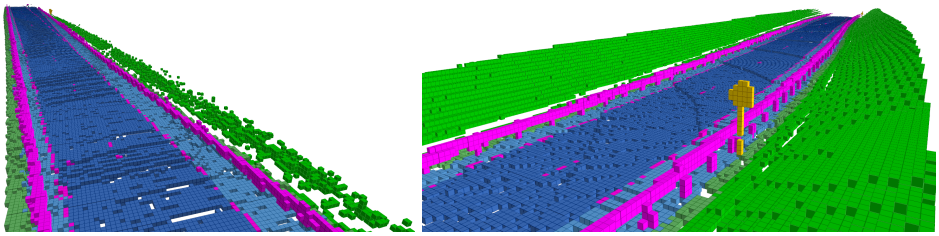


Figure 2.33 – Illustration de la représentation volumétrique et sémantique générée pour qualifier automatiquement un système de perception dédié à la détection des éléments de structure de la scène (ex. chaussée, marquage routier, barrières, panneaux de signalisation).

Les recherches conduites ont abouti au prototypage et la preuve du concept de la méthodologie. Deux types de fonctions de détection ont été évaluées : une fonction de détection de la chaussée basée sur la télémétrie LiDAR et un détecteur du marquage basée vision. Ce dernier est un module de détection fournit par un équipementier automobile. Le prototype a permis non seulement d'établir un comparatif standardisé, mais aussi, il se profile comme un outil essentiel pour la caractérisation des conditions d'usage dans lesquelles les fonctions de détection soufrent par leur perte de performance. Ainsi, à mon sens, l'outil est d'une importance clé dans le but de formaliser l'intégrité des informations exploitées par les systèmes de perception embarquées pour les véhicules à conduite automatisée et les fonctions de sécurité active.

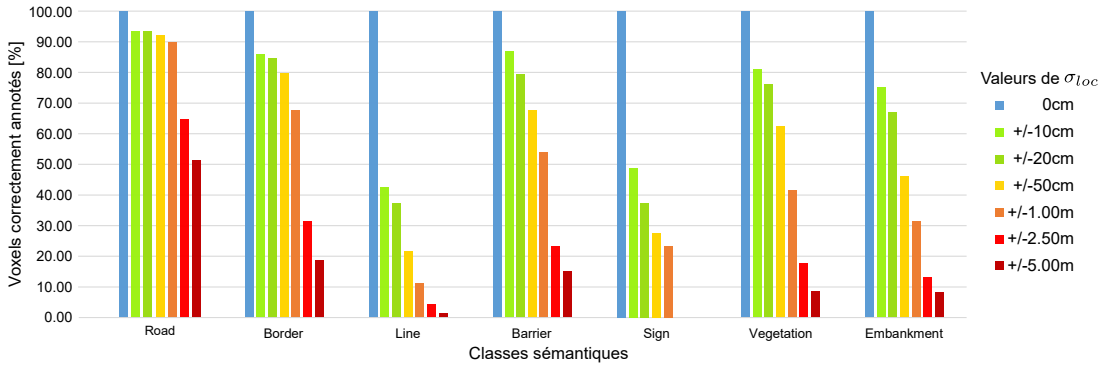


Figure 2.34 – Représentation graphique des prérequis de localisation sur la méthode de référencement automatique : avec le pourcentage de voxels automatiquement annotés en fonction de la classe sémantique et de la borne d'incertitude sur la localisation du véhicule porteur.

L'ensemble de mes travaux, associées à cet axe de recherche, couvre un large spectre fonctionnel des systèmes de perception multi-modale embarqués. Les contributions algorithmiques effectuées ont été axées sur des méthodes et des stratégies pertinentes ayant un fort potentiel d'embarquabilité, avec de nombreuses preuves de concepts. Dans la suite du manuscrit, j'introduis le deuxième axe de recherche portant sur la conception conjointe logicielle/matérielle de systèmes de perception multi-modale embarqués.

2.4.1.3 Vers l'étude des interactions entre les SPME et les utilisateurs

Cet axe constitue une ouverture à mes thématiques de recherche, s'intégrant dans une dynamique de l'équipe MOSS-SATIE. Dans ce cadre, en 2018, j'ai formalisé cette problématique dans [32] en m'asso-

ciant dans le périmètre de l'Université Paris-Saclay à d'autres laboratoires de recherche comme le laboratoire IBISC. L'axe se fonde sur la compréhension et la gestion des interactions occupants/véhicule. Ces interactions sont un verrou scientifique majeur dans le but de garantir la sécurité non seulement des passagers mais aussi des autres usagers de la route [76].

Les interactions entre les occupants et les automatismes des systèmes ADAS/VA ont été étudiées dans l'état de l'art en structurant cette problématique selon trois cas d'intérêt : la reprise du contrôle [72], le retour d'information véhicule/conducteur et la modélisation de la conduite. L'étude de ces interactions a souvent été réalisée dans des situations nominales sur des simulateurs de conduite.

Pour répondre pleinement à cet enjeu sociétal, j'ai entamé des actions pour initier la caractérisation de situations dites critiques, par la dynamique relativement élevée à laquelle les interactions par l'intervention des automatismes opèrent (ex. vitesse du véhicule de 70km/h). En effet, l'étude de cette problématique nécessite des compétences méthodologiques et des moyens très spécifiques pour la mise en œuvre et la validation expérimentale des contributions innovantes. En France, par exemple, on peut citer les contributions des acteurs du domaine comme le laboratoire Heudiasyc CNRS/UTC, l'UGE-IBISC, VEDECOM et l'IRT-SystemX. En Europe, le laboratoire KIT (Karlsruhe Institute of Technology) et l'un des principaux acteurs de la recherche pour les voitures autonomes s'intéressant également aux itérations conducteur/voiture autonome. Aux USA, le Crash Imminent Safety (CrIS) University Transportation Center à l'Université de l'Ohio effectue entre autres, l'expérimentation de situations sous « stress dynamique » pour l'amélioration de fonctions d'assistance.

Ainsi grâce au financement obtenu par le projet émergent INVAHSIvE de l'Université Paris-Saclay, dont j'ai été le porteur, le véhicule expérimental du laboratoire SATIE a été robotisé par une solution que l'équipe MOSS-SATIE a développée, déployée et validée expérimentalement. Cette progression a permis à l'équipe de se positionner en tant que partenaire dans le projet européen H2020 Drive2TheFuture, portant l'objectif d'étudier l'acceptabilité des automatismes dans les systèmes de transport à conduite automatisée (terrestre et maritime). En particulier, je me suis intéressé à la création d'une base de données permettant la corrélation des comportements des conducteurs pendant une manœuvre d'évitement d'obstacle automatique sur piste d'essai.

La collecte de ces données a été effectuée pour un ensemble de 22 participants sous l'approbation du comité d'éthique de l'Université Gustav Eiffel (UGE). Ces expériences ont également été évaluées par l'utilisation de questionnaires normalisés par les partenaires du projet. L'utilisation des données collectées permettra d'entamer prochainement une étude exploratoire d'analyse et de caractérisation des comportements des conducteurs. Les résultats pourront aboutir aux premières contributions dans cette problématique.

Plus récemment, je participe, en tant que partenaire, au projet de recherche ANR eMC2 et je suis responsable de l'étude exploratoire vers l'analyse du comportement de conducteurs de motocyclette électriques pour l'identification de situations de conduite à risque. Dans ce cadre, l'analyse du regard du conducteur et son association à la sémantique de la scène définiront les fondations des contributions scientifiques à venir.

En conclusion, les actions menées dans le cadre de cet axe de recherche illustrent l'investissement et les efforts dévoués vers cette thématique. Cet axe a abouti à une production scientifique soutenue touchant tous les éléments qui composent la perception embarquée pour les véhicules intelligents. Il a également permis d'établir un nombre important de collaborations scientifiques enrichissantes au niveau national et européen. Une ouverture thématique ver l'étude des interactions entres les SPME et les utilisateurs alimente et met en place une dynamique de recherche avec des perspectives à court, moyen et long terme.

2.4.2 Conception conjointe logicielle/matérielle d'algorithmes de perception multimodale

Les SMPE dédiées à des applications d'assistance à la conduite ou à la conduite automatisée sont contraints intrinsèquement en énergie, en ressources de calcul et en temps de traitement. Dans la Section 2.4.1, les recherches menées ont permis l'identification et la caractérisation des SPME pour la localisation du porteur et pour l'analyse des scènes dynamiques. Les problématiques traitées m'ont permis de déceler l'impact de multiples modalités de perception, d'identifier et valider des modèles et des représentations communes aux modalités sensorielles pour leur exploitation et analyse, et enfin de proposer et valider une méthodologie d'évaluation des SPME.

Une problématique amplement étudiée par la communauté scientifique est celle du SLAM [15] [36]. Le SLAM formalise un processus d'estimation conjointe de la localisation d'un porteur (i.e. vecteur mobile) ainsi que de l'inférence d'une représentation de l'environnement où le porteur évolue. Le traitement simultané du processus de localisation et de cartographie aboutit à une solution de localisation précise localement et à une représentation riche en primitives d'apparence et/ou de structure. Il est à noter que le SLAM ne s'intéresse pas à l'analyse de la dynamique de la scène avec des exceptions comme dans le SLAM+MOT [46] ou très récemment au SLAM dynamique [98]. L'isolation et l'exclusion des informations dynamiques est essentiel pour préserver la consistance du processus du SLAM.

Le SLAM, dans sa variante basée sur l'optimisation de graphes, GraphSLAM [42], est un formalisme qui s'est rapidement répandu par sa capacité à gérer des environnements à grand échelle en limitant la complexité algorithmique de l'approche, rendant ainsi sa portabilité atteignable. Depuis son apparition, le SLAM basé sur l'optimisation de graphes n'a pas cessé d'évoluer et se consolide aujourd'hui comme l'approche la plus adaptée pour les applications de véhicules terrestres.

Dans le contexte de mes recherches, je me suis intéressé aux systèmes de SLAM embarqués. L'hypothèse est fondée sur la prise en considération de trois éléments structurels du système et de ses interactions : les moyens de perception, l'algorithme et l'architecture de calcul. Sur cette prémisse, j'ai engagé une collaboration sur cet axe thématique transverse avec Abdelhafid El Ouardi, spécialiste dans le domaine de l'adéquation algorithme-architecture.

Les premières études ont été portées vers la compréhension et l'évaluation de la complexité algorithmique. Les orientations et les choix stratégiques se sont très rapidement orientés vers un algorithme de référence de l'état de l'art dénommé ORB-SLAM [71]. Il s'agit d'un algorithme de SLAM monoculaire basé sur l'optimisation de graphes qui s'articule en deux blocs de traitements : le *front-end* et le *back-end*. Le premier bloc de traitement opère l'extraction de primitives et leur suivi temporel pour achever à l'issue une estimation de mouvement (i.e. odométrie). Le deuxième bloc construit une représentation spatiale éparse de l'environnement en utilisant des images clés et ses primitives organisées dans une structure de type dictionnaire. Cette structure facilite l'identification des lieux revisités pouvant configurer une trajectoire en fermeture de boucle. Ce type de trajectoire permet l'optimisation du graphe, limitant efficacement l'accumulation des erreurs (dérives) par l'intégration odométrique.

En s'inspirant de ORB-SLAM, j'ai participé à la définition d'un algorithme de vision stéréoscopique se fondant sur l'utilisation des primitives bio-inspirées HOOFR [77]. Ces primitives ont démontré avoir un meilleur rendement en précision et en répétabilité que les primitives ORB, tout en utilisant un descripteur binaire, adéquat à une implémentation matérielle. La stratégie de calcul de pose a été également redéfini par l'utilisation d'un calcul robuste de pose, filtré et pondérée par un modèle

15. *Simultaneaous Localization and Mapping*

d'erreur, d'une faible complexité. La synthèse fonctionnelle du *front-end* de l'algorithme est illustrée par la Fig. 2.35.

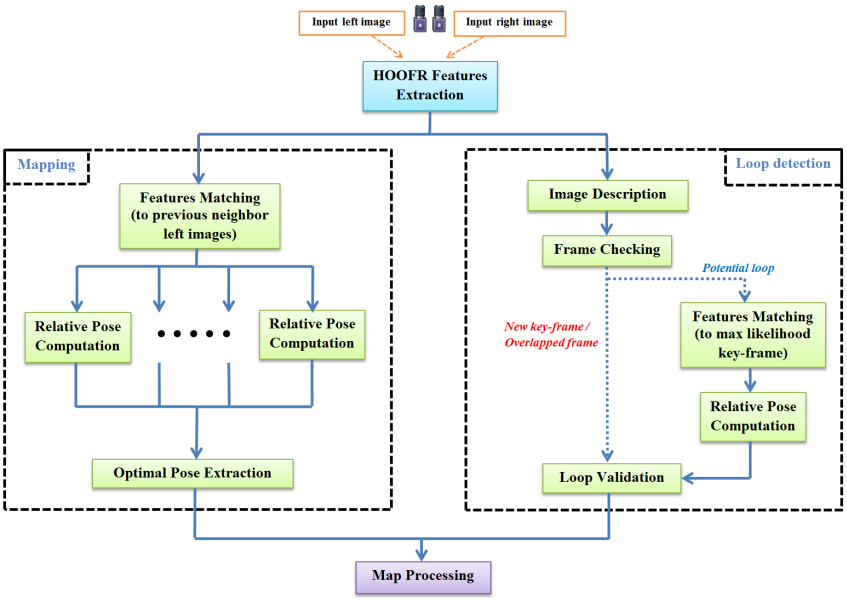


Figure 2.35 – Schéma fonctionnel du HOOFR Stéréo SLAM [8].

L'impact des contributions a été démontré au travers d'une évaluation complète dans six bases de données KITTI [61], Oxford [84], Malaga [65], MRT [55], St-Lucia [53] et NewCollege [50]. La qualité des résultats est illustrée par les trajectoires de la Fig. 2.36 où nous avons comparé la trajectométrie d'ORB, en rouge, d'HOOFR Stéréo SLAM, en vert, et du positionnement GNSS de référence, en bleu. Le bilan de cette comparaison établie un niveau de performance de l'algorithme proposé, étant supérieur ou équivalant à celui d'ORB.

L'évaluation quantitative en termes de la racine de l'erreur quadratique moyenne, RMSE, sur la base de données KITTI, a relevé une erreur moyenne de 1% de la longueur du parcours par rapport à la trajectoire de référence. Cela avec une réduction considérable de la complexité algorithmique en adoptant un formalisme facilitant l'optimisation matérielle. Il faut noter que ces contributions et améliorations algorithmiques se situent au *front-end* de l'algorithme. L'étude complète et détaillée a été publiée dans [8].

Ces recherches ont été conduites dans le cadre des travaux de thèse de Dai-Duong Nguyen [87]. Ces travaux ne se sont pas arrêtés à l'algorithmique mais ont été poursuivis par le déploiement d'une première preuve de concept embarquée adéquate à une architecture de calcul hétérogène du type CPU-GPU. L'implantation proposée a consisté à déporter la fonction d'association de primitives (temporelle et stéréoscopique) sur GPU assurant un traitement partiellement parallèle du *front-end*. La qualification de ce prototype a permis d'établir avec des résultats équivalents en précision de localisation, un temps de traitement pouvant attendre en moyenne 20Hz dans une cible à haute performance Intel et 6Hz sur l'architecture embarquée NVIDIA Tegra X1.

Par la suite, et tenant compte des contraintes en énergie consommée des systèmes SLAM embarqués, nous nous sommes investis, en toute cohérence, vers des architectures hétérogènes de calcul incluant de circuits de traitement massivement parallèles avec une faible consommation, FPGA-CPU. Pour ce faire, un deuxième prototype d'implantation basée sur OpenCL, cette fois-ci, de la fonction d'extraction de primitives HOOFR a été réalisée. L'évaluation de ce bloc de traitement a permis d'établir un

gain étant entre 7 à 9 fois plus rapide que sur un CPU, en conservant la même qualité de résultats. En comparaison avec une implantation CPU-GPU, non seulement l'implantation CPU-FPGA reste plus rapide mais elle consolide un bilan énergétique très faible.

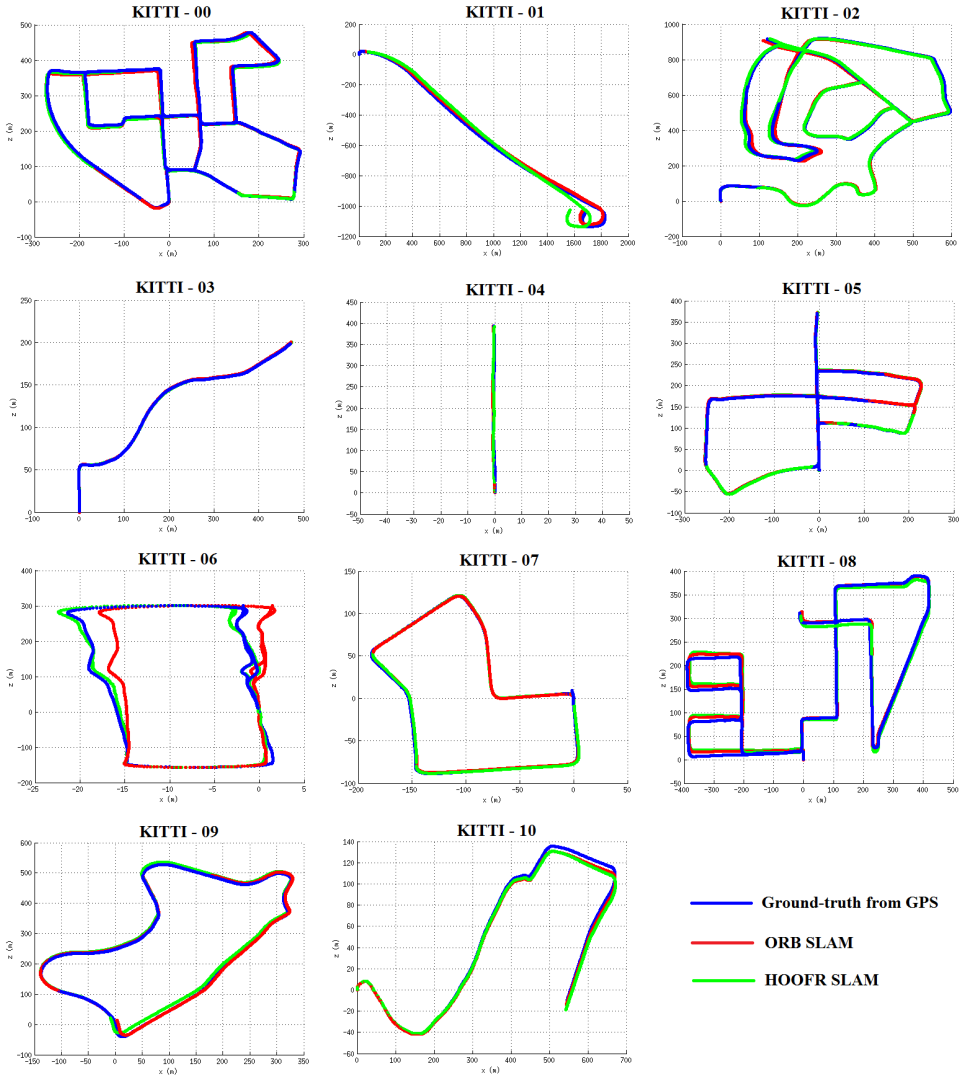


Figure 2.36 – Résultats expérimentaux du HOOFR Stéréo SLAM dans la base de données KITTI. Ils représentent la trajectométrie de ORB, en rouge, HOOFR Stéréo SLAM, en vert, et du positionnement GNSS de référence, en bleu.

J'ai poursuivi mes recherches et mon étroite collaboration sur l'axe des systèmes de SLAM embarqués réconforté par les résultats obtenus mais aussi parce que je suis formellement convaincu que la conception d'un système de SLAM embarqué doit tenir compte des interactions entre les éléments structurels du système.

J'ai entamé l'étude d'interactions entre la perception et l'algorithme SLAM, l'objectif étant d'améliorer le couplage entre eux. J'ai dirigé ainsi des recherches assurant l'évolution d'HOOFR Stéréo SLAM vers sa variante en vision RGB-D. Dans une démarche progressive, je me suis référé à la méthode ORB-SLAM2 [83] en couplage à un système de vision RGB-D [16] effectuant un traitement de données au plus près du capteur.

16. Intel Realsense D435i/455

Une étude préliminaire, sur de données collectées en environnement d'intérieur [2], a permis d'identifier la modalité du capteur RGB-D réalisant une meilleure localisation en termes de l'erreur translationnelle. Les expériences ont comporté un comparatif entre ORB-SLAM2, RTAB-Map et une localisation de référence établit en post-traitement par COLMAP [80]. Les résultats ont démontré que l'utilisation de la modalité Active IR-D (i.e. images infrarouges + carte de profondeur + lumière structurée) assure un compromis entre précision de localisation et la gestion de cas d'usage. Ces observations sont fondées sur la base de critères d'évaluation et sur les caractéristiques intrinsèques du capteur. Ces dernières concernent l'ouverture du champ, l'utilisation d'un imageur à exposition globale, un meilleur alignement entre l'image et la carte de profondeur et une estimation plus dense de la carte de profondeur.

L'étude paramétrique de l'algorithme ORB-SLAM2 a permis d'identifier trois paramètres ayant un impact sur les performances en localisation de l'algorithme. Parmi ces paramètres, le seuil de profondeur s'est avéré être le paramètre en lien avec les caractéristiques du système de vision RGB-D. Les deux autres, nombre de primitives et seuil du détecteur FAST [51], concernent l'extraction de primitives. Il est important de souligner que les paramètres intrinsèques et extrinsèques du système de vision exercent une forte influence sur les algorithmes de SLAM. Néanmoins, j'ai restreint l'étude au seuil de profondeur afin de limiter la complexité du problème.

Le couplage perception-algorithme a été formalisé comme étant une optimisation paramétrique basée sur une matrice de confusion. Ce formalisme s'inspire des travaux [54] et de [59]. La matrice de confusion est construite sur la base de deux critères recherchés : maximiser le nombre de primitives suivies et minimiser l'erreur translationnel de la localisation du porteur. Ainsi, ce protocole classifie les conditions de l'entrée et de la sortie de l'algorithme afin d'améliorer le couplage avec le système de vision. Une courbe ROC [17] est enfin employée pour la prise de décision suivant deux paramètres : le taux de vrai positifs,TPR, et le taux de faux positifs, FPR définis comme suit :

$$TPR = \frac{TP}{TP+FN} \tag{2.12}$$

$$FPR = \frac{FP}{FP+TN} \tag{2.13}$$

où TP est le nombre de vrai positifs, FP est le nombre de faux positifs, FN est le nombre de faux négatifs et TN le nombre de vrai négatifs. Le point de fonctionnement objectif est caractérisé comme étant celui avec le plus haut TPR et le plus bas FPR.

Les résultats d'une partie des expériences, détaillées dans le Tab. 2.6, confirment et supportent l'amélioration, en pourcentages d'erreurs, obtenue en optimisant le couplage perception-algorithme. On peut constater dans la séquence Digiteo_seq1, par exemple, une réduction d'erreurs de 50% entre l'algorithme ORB-SLAM2 utilisant une modalité IR-Stéréo, en comparaison à la version bénéficiant d'une optimisation paramétrique et utilisant la modalité Passive IR-D. L'algorithme RTAB-Map bénéficie de 28.57% d'amélioration par le changement de modalité de perception. Cette étude, faisant partie des travaux de thèse d'Imad El Bouazzaoui [111], a été publiée dans [3].

17. *Receiver Operating Curve*

		IR-Stereo		Passive IR-D		RTAB error %	ORB error %
		RTAB	ORB	RTAB	ORB(Optim)		
Digiteo_seq1	Tr (m)	0.14	0.14	0.1	0.07	28.57 ↓	50 ↓
	Rot (°)	29.11	11.08	27.98	11.11		
Digiteo_seq3	Tr (m)	0.21	0.23	0.21	0.05	0 —	78.26 ↓
	Rot (°)	20.23	9.39	19.91	9.4		

Table 2.6 – Résultats détaillant une réduction d'erreurs par l'amélioration du couplage perception-algorithme. Deux modes de perception, IR-Stéréo et Passive IR-D, et deux algorithmes, RTAB-Map et ORB-SLAM2, sont comparés.

La formalisation de l'algorithme HOOFR-SLAM dans sa variante RGB-D a été réalisée par la suite. Cette variante intègre les adaptations nécessaires au *front-end* pour la prise en compte des informations issues du système RGB-D. L'algorithme opère un filtrage robuste basé sur l'écart absolu médian (sigle en anglais MAD, *Median Absolute Deviation*) des primitives et de leur mesure de profondeur associée. Ce filtrage a pour vocation d'exclure les primitives ayant une mesure de profondeur non pertinente pour l'estimation de mouvement (e.g. primitives situées dans le ciel ou à l'horizon). L'estimation de pose basé sur la structure d'estimation robuste de RANSAC a été remplacée par une estimation robuste basé sur PROSAC [40]. Cette méthode accélère la convergence significativement en conservant une estimation précise. Les expériences de validation ont montré que cette méthode robuste obtient un gain d'accélération de 5x sans perte de précision dans l'estimation.

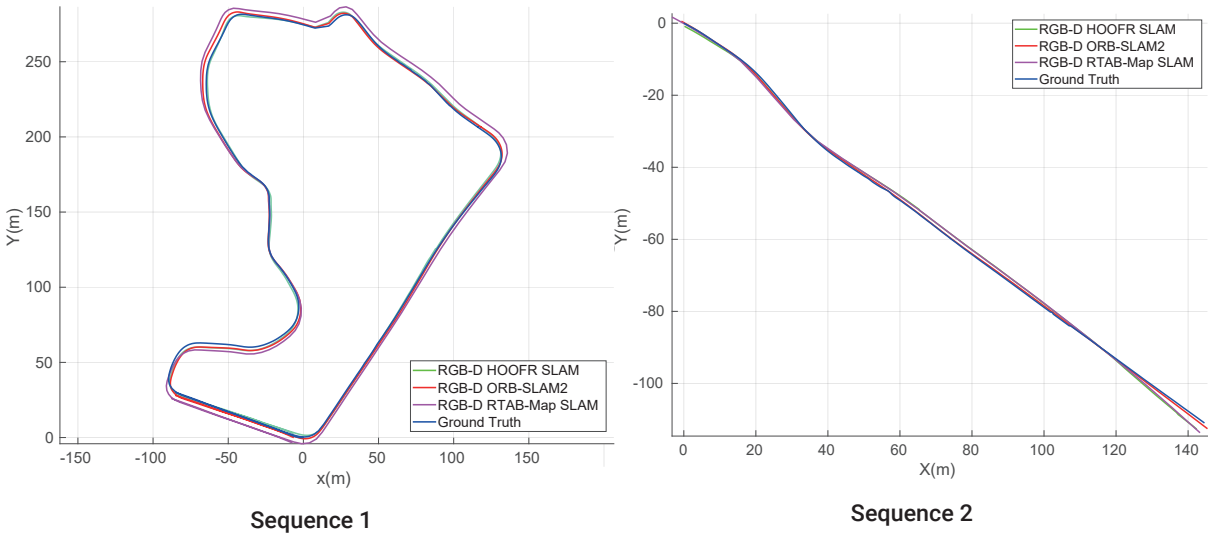


Figure 2.37 – Trajectometrie comparative de trois systèmes de SLAM RGB-D. La trajectoire estimée par RGB-D HOOFR-SLAM en vert, par RGB-D RTAB-Map en violet, par RGB-D ORB-SLAM2 en rouge et la trajectoire de GNSS-RTK de référence en bleu.

En synthèse, les modifications introduites dans la variante RGB-D HOOFR-SLAM réduisent la complexité des traitements, augmentent la robustesse du processus d'estimation et rapprochent les traitements au plus près des capteurs. Cette variante algorithmique d'HOOFR-SLAM étroitement couplée au système de perception a été validée et évaluée sur deux séquences en milieu semi-urbain. La base de données collectée par le véhicule expérimentale du laboratoire SATIE (voir Fig. 2.30) inclue le flot vidéo d'une caméra Intel Realsense D455, un télémètre LiDAR Velodyne Puck-16 et d'un système de positionnement GNSS-RTK Altus APS-3. Les données représentent deux trajectoires respectivement de

1km et 200m de distance avec un trafic régulier (i.e. présence des véhicules et autres usagers de la route).

	RGB-D HOOFR		**RGB-D ORB**		Évolution des erreurs		**RGB-D RTAB-Map**		Évolution des erreurs	
	RMSE (m)	FPS	RMSE (m)	FPS	RMSE (%)	FPS (%)	RMSE (m)	FPS	RMSE (%)	FPS (%)
Seq1 (L : 978.57m)	**1.47**	**27.54**	2.27	17.11	-35.24	+60.96	4.39	13.77	-66.51	+100
Seq2 (L : 189.81m)	**0.93**	**28.57**	1.18	18.02	-21.19	+58.55	3.18	16.41	-70.75	+74.1

Table 2.7 – Bilan des performances des systèmes de SLAM RGB-D évalués selon le RMSE(m) et leur fréquence de traitement (FPS).

La Fig. 2.37 illustre la trajectométrie obtenue par le système RGB-D HOFFR-SLAM, celle estimée par deux autres méthodes de l'état de l'art et la trajectoire de GNSS-RTK de référence. Il est important de constater que la Séquence 2 a pour but d'évaluer la qualité des estimations à la sortie du front-end du SLAM, car la séquence ne décrit pas une fermeture de boucle. Ainsi, aucun système de SLAM dans ce type de configuration ne peut faire usage du schéma d'optimisation du graphe. Dans la séquence 1, les résultats rapportés sont ceux obtenus à fermeture de boucle optimisée.

Le bilan obtenu, détaillé par le Tab. 2.7, confirme la faible dérive du *front-end* proposé avec le temps de traitement le plus faible. Les temps de traitement ont été obtenus par l'exécution temps-réel sur une station de travail.

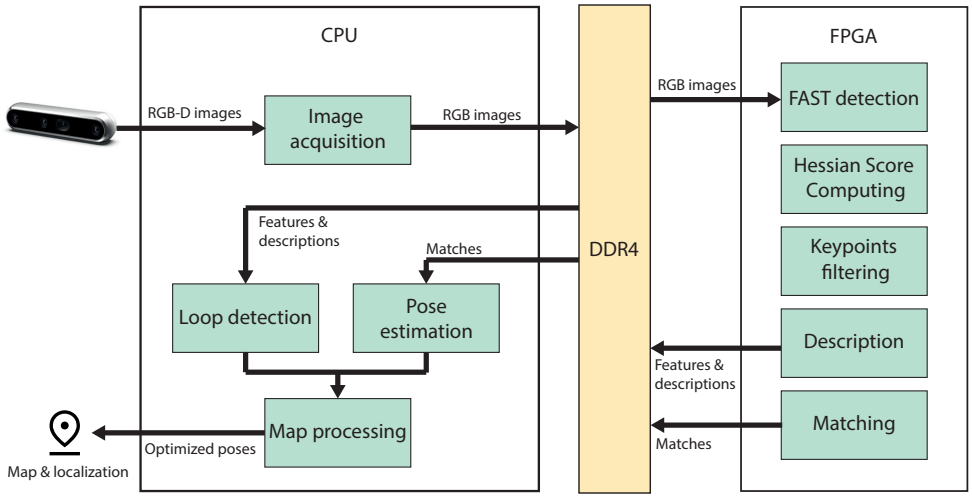


Figure 2.38 – Architecture hétérogène CPU-FPGA pour le système RGB-D HOOFR-SLAM embarqué et son partitionnement en blocs fonctionnels.

Sur la base du système du RGB-D HOOFR-SLAM, les travaux ont été poursuivis pour obtenir une solution embarquée qui puisse satisfaire les exigences du temps de traitement en accord avec les contraintes de l'application. Ainsi, l'objectif qu'a été défini est celui d'un débit de traitement supérieur à 30Hz. Ce débit assure non seulement le traitement intégral du flot d'information du système de vision RGB-D, mais aussi permet d'assurer des estimations d'une scène dynamique urbaine [18].

Le processus d'adéquation de l'algorithme RGB-D HOOFR-SLAM avec un architecture matérielle de calcul a été poursuivi, étant donné que les recherches initiales sur HOOFR Stéréo SLAM nous ont permis de porter l'extraction de primitives sur deux architectures hétérogènes CPU-GPU et CPU-FPGA. En occurrence, l'association temporelle (mise en correspondance temporelle inter-images) des primitives

18. Équivalant à un déplacement interframe de 50cm pour un véhicule évoulant à 50 Km/h.

a été réalisée et évaluées sur deux cibles embarquées : la cible NVIDIA AGX Xavier CPU-GPU et la cible Intel Arria 10 CPU-FPGA.

L'évaluation de performances du système RGB-D HOOFR-SLAM sur l'architecture CPU-GPU embarquée a permis d'attendre un temps de traitement de 41.85ms l'équivalent à 23,89fps. En revanche, le déploiement sur la cible CPU-FPGA du modèle d'architecture illustré par la Fig. 2.38, a permis d'attendre une période de traitement de 33ms l'équivalent à 30fps avec des images à haute résolution (i.e. 720p).

La démarche méthodologique de la recherche appliquée aux systèmes SLAM embarquées et les avancées synthétisées dans cet axe de recherche, confirment l'importance et la pertinence de prémisses établies pour la conception de ce type de systèmes. Ainsi, je considère pertinent de poursuivre mes actions dans cet axe vers une extension de cette méthodologie aux systèmes de SLAM multi-modal. Les recherches devront également s'étendre vers le *back-end* du système SLAM. En effet, les représentations multi-modales de l'environnent font partie des problématiques abordées par l'état de l'art afin de répondre aux exigences de robustesse et d'adaptabilité aux changements de l'environnement.

Chapitre 3

Conclusions et perspectives

Des travaux de recherche couvrant le large spectre de la perception multimodale pour les systèmes embarqués a été présenté. En particulier, les détails des contributions, mettent en évidence la méthodologie scientifique suivie ainsi que l'impact et l'étendu des recherches qui ont été conduites.

Articulé selon deux axes de recherche majeurs, j'ai pu réaliser des contributions qui ont répondu à diverses problématiques associées aux systèmes de perception multimodale embarqués pour les véhicules intelligents. Mes travaux m'ont permis de consolider une vision globale de la thématique partant du système multi-capteur jusqu'à l'architecture du calcul embarqué. Cette exploration a toujours nécessité une étude en passant par la complexité algorithmique et l'intégrité du système.

Les recherches menées pour l'analyse de scènes dynamiques ont été focalisées sur deux directions celle de la localisation et celle s'intéressant à la perception de l'environnement. En ce qui concerne la localisation, les études conduites ont permis l'instanciation d'une méthode qualifiant l'intégrité d'un système de localisation multimodale. Ces concepts ont été validés à l'expérimental. Les expériences ont vérifié, pour un SPME dédié à la localisation, les prérequis d'intégrité en termes de localisation latérale d'un véhicule en milieu urbain. En revanche, les niveaux d'intégrité de la localisation longitudinal n'ont pas pu attendre les niveaux requis pour une conduite automatisée. Cela reste une problématique ouverte à ce jour qui évoluera avec l'émergence de nouveaux moyens de perception plus performants, des Systèmes d'Information Géographiques HD plus riches en sémantique et des stratégies de perception distribuée s'appuyant sur des infrastructures communicantes V2X.

La perception de l'environnement est un vaste sujet avec des problématiques transverses. Mes contributions ont suivi une approche systémique ayant confirmé la nécessité impérative d'une description multidimensionnelle et multi-physique (capteurs se fondant sur des principes de mesure distincts) de la scène combinant l'apparence, la structure et leur évolution dans le temps, ainsi que des informations sémantiques. Il est temps, à mon sens, d'orienter les recherches vers la conception de systèmes multimodales pouvant répondre non seulement aux critères de performance fonctionnels mais aussi aux prérequis d'intégrité et de redondance nécessaires à des applications de transport. Dans ce cadre, j'ai pu proposer un outil de qualification de moyens de perception facilitant le processus de conception et de caractérisation des systèmes de perception complexes. Ces outils devront évoluer et ne pas se limiter aux systèmes de détection. Leur extension à la qualification des systèmes d'analyse et de compréhension de la scène nécessitera des moyens de référencement pour les objets dynamiques.

Les méthodes de détection multimodales basées sur l'apprentissage profond se développent ainsi que les modalités technologies de perception comme la vision évènementielle et la télémétrie LiDAR à état

solide. Ces méthodologies et moyens de perception permettront l'achèvement de nouvelles avancées dans la compréhension de la scène. L'intégration des modalités comme vision évènementielle constitue un défi dans la conception de représentations de l'environnement (explicables et vérifiables) facilitant la fusion de sources multiples d'information. La conception d'indicateurs d'intégrité ainsi que leur adéquation sur des architectures embarquées énergétiquement efficaces pour les méthodes de détection basées sur l'apprentissage devront également faire l'objet d'études approfondies.

Actuellement, je mène des travaux sur la conception conjointe de SPMEs fondés sur des algorithmes d'intelligence artificielle étroitement couplés à des architectures matérielles de calcul. De nombreuses questions et problématiques émergent parallèlement aux avancées méthodologiques de l'apprentissage profond et de technologies sensorielles énergétiquement efficiente. J'ai lancé des travaux sur ces aspects dans le cadre d'une thèse en cotutelle avec le laboratoire LASTIMI de l'Université M. V de Rabat (thèse F. Guerrouj, soutenance prévue 2024).

Les expériences et les résultats obtenus dans les recherches conduites pour l'analyse de scène dynamique m'ont amené à m'intéresser aux systèmes de SLAM embarqués. En considérant un modèle de système composé par les capteurs, les algorithmes et l'architecture de calcul, je me suis focalisé sur les méthodes permettant d'améliorer le couplage capteur-algorithme et algorithme-architecture. Il en résulte l'obtention d'une méthode dénommée RGB-D HOOFR SLAM ayant démontré à l'expérimentale, une notable amélioration par une optimisation paramétrique du couplage entre le capteur et l'algorithme. A court terme, une démarche assurant une progression vers l'objectif de consolider un système SLAM multimodale embarqué est entamée. Les travaux réalisés ont consolidé une solution sur architecture hétérogène FPGA-CPU où le partitionnement des processus du *front-end* est majoritairement parallélisé. En revanche les processus associés au *back-end*, sont peu optimisés pour le traitement embarqué par des importants prérequis en ressources de mémoire. Ainsi, deux questions focalisent mon attention à ce sujet :

i) Pouvons-nous obtenir une représentation multimodale de l'environnement adéquate au traitement optimisé à l'embarqué ?
ii) Pouvons-nous améliorer et augmenter le niveau d'intégrité de la reconnaissance des fermetures de boucles par une approche multimodale ?

La première question a été largement étudiée pour les systèmes SLAM fondés sur la perception unimodale où de nombreuses primitives ont été évaluées parmi lesquels on peut évoquer les primitives d'image, les primitives de structure (e.g. droites, plans) et les primitives sémantiques. Plus récemment, l'apparition des représentations de la scène obtenues par l'encodage de données capteurs émergent [102, 86, 110] se profilant comme une rupture algorithmique. Elles promettent une amélioration significative du processus de cartographie et de relocalisation inhérents au SLAM par optimisation de graphe. Ces approches réconfortent la nécessité d'introduire des représentations multidimensionnelles de l'environnement.

La deuxième question se situe dans une problématique complexe se fondant sur la définition d'une trajectoire décrivant une boucle. Les boucles sont caractérisées par une forte corrélation entre une scène observée et une scène clé faisant partie de la carte (i.e. lieu revisité). Une détection multi-source des boucles peut ainsi amener à des multiples situations d'ambigüité/conflits où chaque source peut identifier la fermeture dans de moments distincts de la trajectoire. La résolution de ces ambigüités et l'identification des boucles avec le plus fort taux de réduction de la dérive sont des questionnements traités dans le cadre de travaux en cours de Mohammed Chghaf (Soutenance prévu en 2023).

À moyen terme, l'aboutissement d'une approche de SLAM multimodal embarqué nous dirigera certainement à des questions portant sur la consommation énergétique des systèmes conçus. Il sera alors pertinent de s'orienter vers l'identification de stratégies de perception reconfigurables sous contraintes.

L'ouverture thématique vers l'étude des interactions entre les systèmes et les utilisateurs m'a permis d'initier des activités de recherche visant des objectifs à long terme. En effet, l'analyse d'interactions dans une contexte de conduite critique (dynamique rapide) relève la compréhension et la caractérisation de phénomènes sous contraintes temporelles fortes (temps-réel). Pour répondre à cette problématique, il est nécessaire d'optimiser les moyens de perception en accord à la dynamique en faisant appel à l'utilisation d'architectures de calcul dédiés. Les observations issues des expériences serviront à des études pluridisciplinaires (d'ergonomie par exemple) faisant appel à des professionnels de spécialité STAPS (psychologues, ergonomes), neuro-sciences, mais aussi à des algorithmes d'intelligence artificielle.

Références bibliographiques

L'ordre des références suivantes suit l'indexation préalablement indiquée dans la Sec. 1.5.

[40] O. Chum and J. Matas, "Matching with prosac-progressive sample consensus," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 220–226, IEEE, 2005.

[41] J. E. Boritz, "Is practitioners' views on core concepts of information integrity," *International Journal of Accounting Information Systems*, vol. 6, no. 4, pp. 260–279, 2005.

[42] S. Thrun and M. Montemerlo, "The graph slam algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[43] E. Dickmanns, *Dynamic Vision for Perception and Control of Motion*. Springer London, 2007.

[44] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3d scene analysis from a moving vehicle," in *2007 IEEE conference on computer vision and pattern recognition*, pp. 1–8, IEEE, 2007.

[45] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *2007 IEEE conference on computer vision and pattern recognition*, pp. 1–8, IEEE, 2007.

[46] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007.

[47] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Computer Vision – ECCV 2008* (D. Forsyth, P. Torr, and A. Zisserman, eds.), (Berlin, Heidelberg), pp. 44–57, Springer Berlin Heidelberg, 2008.

[48] R. Dahyot, "Statistical hough transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1502–1509, 2009.

[49] O. Le Marchand, P. Bonnifait, J. Ibañez-Guzmán, and D. Betaille, "Automotive localization integrity using proprioceptive and pseudo-ranges measurements," in *Accurate Localization for Land Transportation*, vol. 125 of *Les Collections de l'INRETS*, (Paris, France), pp. 7–12, June 2009.

[50] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, 2009.

[51] D. G. Viswanathan, "Features from accelerated segment test (fast)," in *Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK*, pp. 6–8, 2009.

[52] R. Toledo-Moreo, D. Betaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 100–112, 2010.

[53] M. Warren, D. McKinnon, H. He, and B. Upcroft, "Unaided stereo vision based pose estimation," in *Australasian Conference on Robotics and Automation*, vol. 47, p. 60, Citeseer, 2010.

[54] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[55] F. Moosmann and C. Stiller, "Velodyne slam," in *2011 ieee intelligent vehicles symposium (iv)*, pp. 393–398, IEEE, 2011.

[56] J. Moras, V. Cherfaoui, and P. Bonnifait, "Moving Objects Detection by Conflict Analysis in Evidential Grids," in *IEEE Intelligent Vehicles Symposium (IV 2011)*, (Baden-Baden, Germany), pp. 1120–1125, June 2011.

[57] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *Computer Vision–ECCV 2012 : 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pp. 376–389, Springer, 2012.

[58] T. Kühnl, F. Kummert, and J. Fritsch, "Spatial ray features for real-time ego-lane extraction," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 288–293, IEEE, 2012.

[59] M. J. Milford and G. F. Wyeth, "Seqslam : Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE international conference on robotics and automation*, pp. 1643–1649, IEEE, 2012.

[60] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 1693–1700, IEEE, 2013.

[61] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics : The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, pp. 1231–1237, Aug. 2013.

[62] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics : The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[63] A. Fotouhi, R. Yusof, R. Rahmani, S. Mekhilef, and N. Shateri, "A review on the applications of driving data and traffic information for vehicles' energy conservation," *Renewable and Sustainable Energy Reviews*, vol. 37, pp. 822–833, 2014.

[64] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection : a survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.

[65] J.-L. Blanco-Claraco, F.-A. Moreno-Duenas, and J. González-Jiménez, "The málaga urban dataset : High-rate stereo and lidar in a realistic urban scenario," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014.

[66] C. Zinoune, P. Bonnifait, and J. Ibañez-Guzmán, "Integrity monitoring of navigation systems using repetitive journeys," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 274–280, 2014.

[67] C. Zinoune, *Autonomous integrity monitoring of navigation maps on board intelligent vehicles*. PhD thesis, 2014. Thèse de doctorat dirigée par Bonnifait, Philippe et Ibanez-Guzman, Javier Technologies de l'Information et des Systèmes : Unité de recherche Heudyasic (UMR-7253) Compiègne 2014.

[68] W. Lu, *Contributions to Lane Marking Based Localization for Intelligent Vehicles*. Theses, Université Paris Sud - Paris XI, Feb. 2015.

[69] H.-E. Deghdache and S. Bouchafa, "Driving space detection by combining v-disparity and c-velocity," in *2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 219–224, 2015.

[70] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015 : Towards a benchmark for multi-target tracking," *arXiv preprint arXiv :1504.01942*, 2015.

[71] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam : a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[72] M. Walch, K. Lange, M. Baumann, and M. Weber, "Autonomous driving : investigating the feasibility of car-driver handover assistance," in *Proceedings of the 7th international conference on automotive user interfaces and interactive vehicular applications*, pp. 11–18, 2015.

[73] B. Wang, *Geometrical and contextual scene analysis for object detection and tracking in intelligent vehicles*. Theses, Université de Technologie de Compiègne, July 2015.

[74] On-Road Automated Driving Committee, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Sept. 2016.

[75] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[76] Z. Lu, R. Happee, C. D. Cabrall, M. Kyriakidis, and J. C. de Winter, "Human factors of transitions in automated driving : A general framework and literature survey," *Transportation research part F : traffic psychology and behaviour*, vol. 43, pp. 183–198, 2016.

[77] D.-D. Nguyen, A. El Ouardi, E. Aldea, and S. Bouaziz, "Hoofr : An enhanced bio-inspired feature extractor," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2977–2982, IEEE, 2016.

[78] R. Sabzevari and D. Scaramuzza, "Multi-body motion estimation from monocular vehicle-mounted cameras," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 638–651, 2016.

[79] E. Sattarov, *Contributions of context-aided multi-modal perception systems for detection and tracking of moving objects*. Theses, Université Paris Saclay, Dec. 2016.

[80] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

[81] R. Feng, X. Li, W. Zou, and H. Shen, "Registration of multitemporal gf-1 remote sensing images with weighting perspective transformation model," in *2017 IEEE International Conference on Image Processing*, pp. 2264–2268, 2017.

[82] V. Madhavan and T. Darrell, "The bdd-nexar collective : A large-scale, crowsourced, dataset of driving scenes," *Master's thesis, EECS Department, University of California, Berkeley*, 2017.

[83] R. Mur-Artal and J. D. Tardós, "Orb-slam2 : An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[84] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km : The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[85] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, "Overview of environment perception for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2584–2601, 2017.

[86] G. Kim and A. Kim, "Scan context : Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4802–4809, 2018.

[87] D.-D. Nguyen, *A vision system based real-time SLAM applications*. Theses, Université Paris Saclay (COmUE), Dec. 2018.

[88] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding : A dataset for learning driver behavior and causal reasoning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[89] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, "Gnss position integrity in urban environments : A review of literature," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2762–2778, 2018.

[90] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.

[91] H. Gonzalez, *Complex dynamic scene analysis through multi-body motion segmentation : application to intelligent vehicles*. Theses, Université Paris Saclay (COmUE), Dec. 2019.

[92] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2702–2719, 2019.

[93] I. Salhi, M. Poreba, E. Piriou, V. Gouet-Brunet, and M. Ojail, "Chapter 8 - multimodal localization for embedded systems : A survey," in *Multimodal Scene Understanding* (M. Y. Yang, B. Rosenhahn, and V. Murino, eds.), pp. 199–278, Academic Press, 2019.

[94] T. G. Reid, S. E. Houts, R. Cammarata, G. Mills, S. Agarwal, A. Vora, and G. Pandey, "Localization requirements for autonomous vehicles," *SAE International Journal of Connected and Automated Vehicles*, vol. 2, Sept. 2019.

[95] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years : A survey," *Computing Research Repository*, vol. abs/1905.05055, 2019.

[96] A. Balakrishnan, *Integrity Analysis of Data Sources in Multimodal Localization System*. Theses, Université Paris-Saclay, Dec. 2020.

[97] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes : A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[98] M. Henein, J. Zhang, R. Mahony, and V. Ila, "Dynamic slam : The need for speed," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2123–2129, 2020.

[99] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2 : Audi Autonomous Driving Dataset," 2020.

[100] A. Llamazares, E. J. Molinos, and M. Ocana, "Detection and tracking of moving obstacles (datmo) : A review," *Robotica*, vol. 38, no. 5, p. 761–774, 2020.

[101] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving : Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[102] H. Wang, C. Wang, and L. Xie, "Intensity scan context : Coding intensity and geometry relations for loop closure detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2095–2101, 2020.

[103] X. Li, L. Chen, S. Li, and X. Zhou, "Depth segmentation in real-world scenes based on u–v disparity analysis," *Journal of Visual Communication and Image Representation*, vol. 73, p. 102920, 2020.

[104] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k : A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2636–2645, 2020.

[105] N. Zhu, D. Betaille, J. Marais, and M. Berbineau, "Gnss integrity monitoring schemes for terrestrial applications in harsh signal environments," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 3, pp. 81–91, 2020.

[106] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2 : Next generation datasets for self-driving perception and forecasting," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[107] R. Defraiteur, *Évaluation de systèmes d'aide à la conduite. Génération automatique de vérité terrain augmentée à partir d'un capteur haute résolution et d'une cartographie sémantique et 3D ; Evaluation de fonctions de perception tierces*. Theses, Université Paris-Saclay, June 2021.

[108] Z. Li, Y. Du, M. Zhu, S. Zhou, and L. Zhang, "A survey of 3d object detection algorithms for intelligent vehicles development," *Artificial Life and Robotics*, pp. 1–8, 2021.

[109] R. Ravindran, M. J. Santora, and M. M. Jamali, "Multi-object detection and tracking, based on dnn, for autonomous vehicles : A review," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 5668–5677, 2021.

[110] X. Xu, H. Yin, Z. Chen, Y. Li, Y. Wang, and R. Xiong, "Disco : Differentiable scan context with orientation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2791–2798, 2021.

[111] I. El bouazzaoui, *Hardware Software Co-design of an Embedded RGB-D SLAM System*. Theses, Université Paris Saclay, Dec. 2022.

Annexes

Sélection de publications

El Bouazzaoui, Imad, S. Rodriguez Florez and A. El Ouardi, "Enhancing RGB-D SLAM Performances Considering Sensor Specifications for Indoor Localization," IEEE Sensors Journal, pp. 1-1, Apr. 2021.

Cet article présente une étude expérimentale approfondie pour mettre en évidence l'impact des modes d'acquisition d'un capteur RGB-D sur la précision de localisation d'un algorithme SLAM en environnement d'intérieur. Un protocole est proposé pour optimiser le couplage capteur/algorithme. L'évaluation est réalisée sur une base de données accessible publiquement. L'analyse des résultats est fondée sur différentes métriques. Elles sont comparées à celles obtenues avec deux algorithmes de l'état de l'art et une référence.

D. D. Nguyen, A. El Ouardi, S. Rodriguez Florez and S. Bouaziz, "FPGA implementation of HOOFR bucketing extractor-based real-time embedded SLAM applications," Journal of Real-Time Image Processing, Jun. 2020.

Dans ce travail, nous présentons l'implantation d'un algorithme d'extraction de primitives dédiée pour des applications SLAM sur une architecture hétérogène de type SoC-FPGA. Une parallélisation de l'algorithme de l'extracteur HOOFR respectant la complexité de l'algorithme est proposé. Le design est évalué sur une architecture de calcul Intel Arria 10 Soc-FPGA avec un débit de traitement de 54fps avec une résolution de 1226×370 pixels.

Balakrishnan, Arjun, S. Rodriguez Florez and R. Reynaud, "Integrity Monitoring of Multimodal Perception System for Vehicle Localization," MDPI Sensors, vol. 20, no. 16, p. 4654, 2020.

Le travail de recherche présenté dans cet article est un protocole de qualification de l'intégrité d'un système de perception multimodale embarqué dédié à la localisation. La méthodologie proposée évalue l'intégrité des sources fondée sur l'inter-corrélation des informations sous une représentation de type gille d'occupation sémantique. La méthode proposée applique les concepts d'évaluation d'intégrité dans le domaine de l'aviation aux véhicules au sol et fournit les marqueurs de niveau de protection (Horizontal, Latéral, Longitudinal) pour les systèmes de perception utilisés pour la localisation des véhicules.

Defraiteur, Rémi, S. Rodriguez Florez M.-A. Mittet, R. Reynaud, and N. E. Zoghby, "Towards a Reference Data Generation Framework for Performance Assessment of Perception Systems," in IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, Oct. 2019.

Cet article présente un algorithme permettant l'évaluation de performances d'un système de perception. La méthode proposée se fonde sur l'utilisation conjointe de trois sources d'information précises : un SIG 3D sémantique, un LiDAR HD et un système de positionnement GNSS-RTK/INS. Une preuve de

concept a été réalisé et validé à l'expérimentale. Le prototype effectue, sans supervision, la génération de données de référence en combinant les informations issues de sources. Par la suite, il qualifie et caractérise les performances des détections d'un système de perception.

D.-D. Nguyen, A. Elouardi, S. Rodriguez Florez and S. Bouaziz, "HOOFR SLAM System : An Embedded Vision SLAM Algorithm and Its Hardware-Software Mapping-Based Intelligent Vehicles Applications," IEEE Transactions on Intelligent Transportation Systems, pp. 1–16, 2018.

Cet article détaille une méthode de SLAM visuel, dénommé HOOFR SLAM, fondée sur une estimation de mouvement pondérée entre multiples poses voisines. Cet algorithme estime la localisation du porteur en couplant deux modes de stéréoscopie : statique et multivue temporel. Un design d'adéquation de l'algorithme sur une architecture de calcul hétérogène de type CPU-GPU est proposé. Une évaluation expérimentale de l'algorithme démontre un équilibre entre consistance de localisation et temps de traitement.

Wang, Bihao, S. Rodriguez Florez and V. Frémont, "Multiple Obstacle Detection and Tracking using Stereo Vision : Application and Analysis," in The 13th International Conference on Control, Automation, Robotics and Vision (ICARCV 2014), Marina Bay Sands, Singapore, Dec. 2014, pp. 1074-1079.

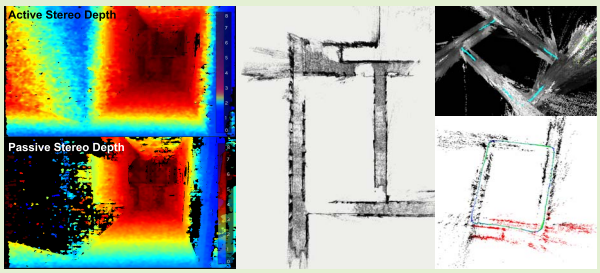
Les travaux présentés dans cet article un système de détection d'objets situés dans une ROI définie par l'espace navigable devant le véhicule. La phase de détection des hypothèses d'objets est opéré par U-disparité. Ces hypothèses sont par la suite suivies par un ensemble de filtres à particules.

Dans la suite de cette annexe, je mets à disposition la source intégrale de ces publications.

Enhancing RGB-D SLAM Performances Considering Sensor Specifications for Indoor Localization

Imad El Bouazzaoui, Sergio A. Rodríguez Florez, and Abdelhafid El Ouardi

***Abstract*—Several works have focused on Simultaneous Localization and Mapping (SLAM), which is a topic that has been studied for more than a decade to meet the needs of robots to navigate in an unknown environment. SLAM is an essential perception functionality in several applications, especially in robotics and autonomous vehicles. RGB-D cameras are among the sensors commonly used with recent SLAM algorithms. They provide an RGB image and the associated depth map, making it possible to solve scale drift with less complexity and create a dense 3D environment representation. Many RGB-D SLAM algorithms have been studied and evaluated on publicly available datasets without considering sensor specifications or image acquisition modes that could improve or decrease localization accuracy. In this work, we deal with indoor localization, taking into account the sensor specifications. In this context, our contribution is a deep experimental study to highlight the impact of the sensor acquisition modes on the localization accuracy, and a parametric optimization protocol for a precise localization in a given environment. Furthermore, we apply the proposed protocol to optimize a depth-related parameter of the SLAM algorithm. The study is based on a publicly available dataset in an indoor environment with a depth sensor. The reconstruction results' analysis is founded on the study of different metrics involving translational and rotational errors. These metrics errors are compared with those obtained with a state-of-the-art stereo vision-based SLAM algorithm.**



***Index Terms*—Depth map, indoor localization, RGB-D cameras, robotics, sensors specifications, SLAM.**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) addresses the problem of inferring maps of an unknown environment and simultaneously solving the ego-localization on the map using embedded sensors. SLAM systems are deployed in several areas including autonomous vehicles [1], indoor robots [2], Unmanned Aerial Vehicles UAV [3], Augmented Reality [4], and more [5], [6]. SLAM systems' importance encourages researchers to provide different solutions to achieve an efficient system with high consistency and performance. In general, SLAM can be classified according to its two main tasks. The first task focuses on processing sensor data, known as front-end processing, which is highly sensor-dependent. The second task is the kernel of SLAM, which is sensor agnostic, and which is categorized according to [7], into filter-based and optimization-based. One type of SLAM that has gained widespread use is visual SLAM. As the name suggests, it uses images as input from a camera or other imaging sensors, such as monocular, stereo, RGB-D cameras. Its popularity is due to the simplicity of embedding these low-cost sensors [8], while taking advantage of a rich supply of information from the environment. Visual SLAM algorithms are classified into two types: feature-based methods (indirect methods) that use the matching of feature points in images with algorithms such as [9], [10] and [11], image-based (direct methods) methods that use the global brightness of images with algorithms such as [12]–[14]. Sparse methods are the most widely used for real-time processing on an embedded system due to their low complexity. However, visual SLAM suffers from several problems related to the environment, which can be a scene with repetitive texture, less textured, or low illuminated [15], [16]. Such conditions are exacerbated by improper or loosely coupling of the sensor and the algorithm. Therefore, an efficient coupling between the sensor's acquisition mode and the SLAM algorithm becomes essential to achieve better performances localization accuracy. In this paper, we demonstrate how the choice of an acqui-

Manuscript received March 5, 2021; accepted April 12, 2021. Date of publication April 16, 2021; date of current version March 14, 2022. The associate editor coordinating the review of this article and approving it for publication was Dr. Valérie Renaudin. *(Corresponding author: Imad El Bouazzaoui.)*

The authors are with SATIE, CNRS UMR 8029, Paris-Saclay University, 91190 Gif-sur-Yvette, France (e-mail: imad.el-bouazzaoui@universite-paris-saclay.fr; sergio.rodriguez@universite-paris-saclay.fr; abdelhafid.elouardi@universite-paris-saclay.fr).

Digital Object Identifier 10.1109/JSEN.2021.3073676

sition mode directly impacts the trajectory reconstruction's accuracy. Besides, we propose a new parametric optimization protocol using the ROC (Receiver Operating Characteristic) curve through an automated process to determine the optimal values of the algorithm's parameters to minimize the error. The remainder of this paper is organized as follows: Section II provides an overview of the work done on Visual SLAM, RGB-D SLAM and the characterization of this type of sensor. In section III, we describe the evaluation methodology, the choice of the SLAM algorithms, the choice of the sensor, dataset description, the parameters analysis process and the metrics used for the evaluation. In section IV, we will give a brief description of the selected SLAM algorithms. In section V, we describe the experimental results. Section VI provides an analysis and a discussion of the reported results. Further, a conclusion and perspectives are drawn.

II. Related Work

Several works have been carried out to improve the visual SLAM, given its multiple advantages. Visual SLAM began by exploiting the images from a single camera called a monocular system [13], [14], [17], and evolved with stereo systems to solve the problem of scale drift [10], [18], [19].

A. Visual SLAM

Some of the stereo SLAM systems' contributions include Mur-Artal and Tardos [11], who contributed with the stereo version of the ORB-SLAM that fixes the problem of scale drift in their monocular version [17]. The stereo version applies the same approach of local bundle adjustment in a set of local keyframes so that the complexity is unaffected by the map's size, therefore making it usable in large scale environments. HOOFR-SLAM [10], a recent algorithm with competitive performance, exploits the HOOFR extractor for feature detection and matching [20]. HOOFR-SLAM implements a processing structure that maximizes parallelism and avoids the need to optimize camera poses by applying bundle adjustments on keyframes or saving the history of map points by estimating the relative poses of the current input frame with a set of previous neighbouring frames. The optimal pose is obtained by averaging the relative poses with weighted factors. PL-SLAM presented by Gomez-Ojeda *et al.* [18], provides a solution for low-textured environments, combining both points and line segments to operate robustly in a wider variety of scenarios, especially in those where point characteristics are rare or poorly distributed in the image. They also introduce a new bag-of-words that relies on combining the descriptive potential of the two types of features. Liu *et al.* [21] proposed a real-time stereo SLAM system based on the bionic eye inspired by the peripheral and central vision of the human eye. With the ability to mimic human eye movements, stereo cameras can improve the SLAM system's robustness in low-textured environments by actively searching for the rich textured area.

B. RGB-D SLAM

The emergence of RGB-D sensors allowed the evolution of 3D dense reconstruction. Many algorithms have exploited the RGB-D images to optimize performance and allow real-time execution, in addition to the embedding capability of these algorithms as on mobiles [22]. A well-known and open-source RGB-D system by Endres *et al.* [23] includes the front-end part dedicated to compute frame-to-frame motion using feature matching and ICP. The back-end part performs optimization of the pose-graph with loop closure constraints based on a heuristic search. Mur-Artal and Tardos [11] have upgraded the ORB-SLAM2 for a loosely coupled use of RGBD input. ORB-SLAM2 uses depth information to synthesize stereo coordinates for the elements extracted from the image. In this way, the system is adaptable whether the input is stereo or RGB-D. Labbé and Michaud [24] have proposed an extension of the RTAB-Map library to implement SLAM with different sensor configurations and processing capabilities. Fu *et al.* [16] proposed an RGB-D SLAM system using points and lines as features, which improved trajectory performance in low textured scenes. Despite the improvement brought by the cited works, their algorithms were evaluated on online datasets, which neglects the impact of the sensor-algorithm coupling. Designing algorithms considering the sensor's properties could significantly improve the localization accuracy.

C. RGB-D Sensors Assessment

Few works have been done on the characterization of RGB-D sensors. Notable works include Carfagni *et al.* [25], who have carried out a characterization of the Intel SR300 depth sensor using it as a 3D scanner. This sensor's performance was evaluated by applying the German association VDI/VDE 2634 on a raw dataset and a dataset with optimized parameters. Another study by the same authors was carried out on the Intel D415 sensor based on the same German standard for 3D scans [26]. Lachat *et al.* [27] performed an evaluation and calibration of the Microsoft Kinect depth camera to reconstruct small 3D objects. To our knowledge, most of the work carried out for the characterization of depth sensors is related to 3D scanning and 3D reconstruction. In this work, datasets are recorded with an RGB-D camera using the various available acquisition modes. Then, an RGB-D SLAM algorithm is evaluated on these datasets while adjusting the front-end part depth related parameters until the best results are achieved by comparing them with a referenced trajectory [28]. In this paper, our contributions are:

- Evaluation of the RGB-D sensor and its acquisition modes for indoor localization and trajectory reconstruction.
- A parametric optimization protocol of sensor-algorithm coupling to achieve better performance [11] in terms of trajectory reconstruction accuracy.
- An online-available RGB-D indoor dataset including different RGB-D acquisition mode sequences for contributing and facilitating further research comparisons [29]–[34].

III. Methodology

A. SLAM Algorithms Choice

Visual SLAM Direct methods estimate camera movement by minimizing photometric errors between consecutive

images. Their strength lies in high accuracy, thanks to the dense image exploitation. The drawback is the sensitivity of these methods to brightness variations or illumination changes. In contrast, feature-based methods use an indirect representation of images, usually in the form of feature points, tracked and then used to estimate the pose by minimizing projection errors. Although indirect methods provide relevant results in well-textured environments, they suffer from failure in poorly textured scenes or motion blur, temporarily wiping out feature points. Direct methods are likely to be computationally expensive, while indirect methods are less computationally expensive. This study's framework is part of the design of an embedded system using RGB-D SLAM for robotics applications. For this reason, the choice of an efficient algorithm regarding accuracy and complexity is a critical asset. Therefore, SLAM algorithms have been chosen to use stereo or RGB-D sensors and are less computationally expensive and consistent. Based on these criteria, we have selected ORB-SLAM2, a baseline algorithm providing satisfactory results and can be embedded [35], for its low complexity compared to other algorithms. ORB-SLAM2 is for monocular, stereo and RGB-D cameras, allowing us to compare stereo and RGB-D (front-end part) without worrying about the back-end. Therefore, the correlation between the sensor and the front-end can be seen. To consolidate our study, we chose RTAB-Map [24], a library implementing SLAM with different methods and supporting various sensors (including Stereo and RGB-D). RTAB-Map offers real-time processing thanks to its appearance-based loop closure approach with memory management making it suitable for large-scale and for long-term operation. The RTAB-Map is used as a baseline method for comparing the acquisition modes' effect without optimising its parameters.

B. Sensor Choice

For the evaluation, an RGB-D camera was chosen for its properties, which are well-suited to SLAM applications. The depth camera incorporates a Vision Processing Unit (VPU), left and right imagers for stereo vision with a wide IR projector, an RGB colour sensor and an Inertial Measurement Unit (IMU). Depth features, high resolution, long-range capability (up to approximately 10 m), global shutter technology enabling fast motion capture without blurring depth images. The depth map can be generated using either active stereo technology by turning the projector on and passive stereo technology by turning it off. The IR projector helps to increase the texture in low textured scenes by projecting a static IR pattern. The vision processor generates the depth map by matching each pixel in the right and left IR images, using the image on the left as a reference for stereo matching. The sensor has an RGB camera, rolling shutter technology, a high resolution, and a narrower field of view.

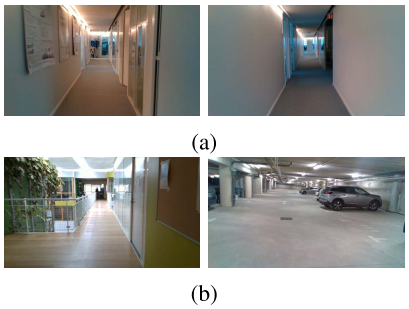
C. Dataset Description

The first stage of the data acquisition is the sensor's calibration, using a manufacturer's tool to perform a dynamic calibration through a target. The intrinsic camera parameters resulting from the calibration are presented in the Table I. The dataset was collected in the laboratory's corridors and the basement parking, as shown in Fig.1. We recorded using an RGB-D camera with a laptop equipped with an Intel Celeron N4100 Quad-Core CPU, 8G RAM and 512GB SSD memory. The experiment was carried out using different acquisition modes, including IR-Stereo, RGB-D Active Stereo (Active: IR Projector on), RGB-D Passive Stereo (Passive: IR Projector off) and IR-D Passive Stereo (Passive: IR Projector off). The IR-D could not be recorded in active stereo since IR patterns interfere with the features extraction generating spurious detections. The images are recorded with a resolution of 1280×720 pixels and a frame rate of 30 frames per second.

The recorded sequences are summarized in the following Table II:

TABLE I
INTRINSIC PARAMETERS OF RGB AND IR CAMERAS OF THE D435I CAMERA USED IN OUR DATASET, INCLUDING FOCAL LENGTH (FX/FX) AND OPTICAL CENTER (CX/CY)

	IR (Left & Right)	RGB
fx	638.14	912.36
fy	638.14	910.26
cx	639.75	648.57
cy	356.51	363.66



(a)

(b)

Fig. 1. Images illustrating the environment (laboratory corridor and basement parking) where the dataset was collected: (a) The scene on the right represents a narrow and textureless environment, on the left we have a narrow scene with more texture. (b) These scenes represent a textured and larger environments.

TABLE II
RECORDED SEQUENCES USING VARIOUS ACQUISITION MODES

Dataset	Description	Projector state	Acquisition mode
Digiteo_seq1 [29]	Lab corridors 1	Passive stereo	IR-D
			RGB-D
			Stereo
Digiteo_seq2 [30]	Lab corridors 2	Passive stereo	RGB-D
		Active stereo	RGB-D
Digiteo_seq3 [31], [32], [33], [34]	Basement parking	Passive stereo	IR-D
			RGB-D
			Stereo
		Active stereo	RGB-D

D. Study of the Sensor-Algorithm Parameters Coupling

Our parametric optimization protocol of sensor-algorithm coupling consists of evaluating the algorithm on different sequences with different acquisition modes. Then, the acquisition mode with the lowest error is used to globally tune the parameters of the algorithm. In this paper, we will focus on the optimization of the ORB-SLAM2 algorithm. ORB-SLAM2 has three parameters related to the algorithm

input, the number of features, the FAST detector threshold and the depth threshold. The number of features relies on the FAST detector, which depends on the exposure [36]. Both parameters are not tied to the depth camera. We have identified a physically correlated parameter to the sensor: the depth threshold [37]. This parameter allows the algorithm to classify near and far features. This parameter is tighly correlated to the error distribution of the RGB-D acquisition mode. We have varied the value of this parameter over a well-defined range. The various tests were carried out on a powerful computing station, equipped with a 24-core Intel Xeon W-2265 processor running at 3.5GHz, 64GB RAM and an NVIDIA Quadro RTX 6000 graphics card with 4608 CUDA cores. We calculated the Euclidean error and the number of tracked points on the map for each frame. This protocol classifies input (Number of points tracked in the map seen by the current frame) and output (Euclidean distance between a current pose and the one in the referenced trajectory) conditions of the algorithm in order to qualify its performance. Inspired from [38], [39], we proposed a parametric optimization based on the following confusion matrix Equ.1.

$$X = \begin{cases} TP & (E_i \leq s) \wedge (E_i \leq E_{i-1}) \wedge (M_i \geq M_{i-1}) \\ FP & (E_i \leq s) \wedge (E_i > E_{i-1}) \wedge (M_i < M_{i-1}) \wedge (M_i < \bar{M}) \\ TN & (E_i > s) \wedge (E_i > E_{i-1}) \wedge (M_i < M_{i-1}) \\ FN & (E_i > s) \wedge (E_i \leq E_{i-1}) \wedge (M_i \geq M_{i-1}) \wedge (M_i > \bar{M}) \end{cases} \tag{1}$$

X denotes the decision on a pose in the confusion matrix. TP stands for True Positive, FP for False Positive, TN for True Negative and FN for False Negative. E_i represents the Euclidean distance between a current pose and the one in the referenced trajectory. s is a given admissible positioning error. M_i represents the number of points tracked in the map seen by the current frame, and $\bar{M}$ represents the average number of points tracked in the map over the whole trajectory. After calculating the position in the confusion matrix of each point, the ROC curve is plotted for all parameter values. The optimum parameter value is the one with the highest True Positive Rate (TPR) and the lowest False Positive Rate (FPR).

E. Evaluation Metrics

Reported results are compared to a Dense Environment reconstruction using SfM-based mapping solution [28], [40], [41]. For this purpose, Evaluation metrics introduced by Sturm *et al.* [42] are used: The absolute trajectory error (ATE) provides a measure of the translational error, from a comparison of the absolute distances between the referenced trajectory and the estimated trajectory. The Relative Pose Error (RPE) is used to find the rotational error. These metrics allow an assessment of the estimated trajectory quality compared to the referenced trajectory. The referenced trajectory and the estimated one are aligned and time-synchronized. RPE metric is evaluated within a time interval of $\Delta = 1$s over 30 frames. For each experiment, the mean of the RPE values is estimated in order to reduce the influence to outliers.

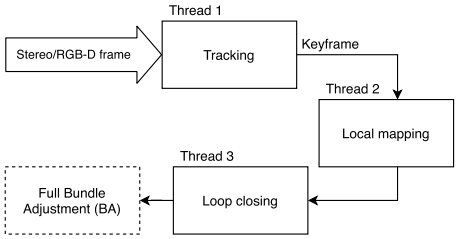


Fig. 2. Overview of the ORB stereo RGB-D thread system. We operate on the pre-processing input module inside the Tracking thread.

IV. Description of the Algorithms to Be Evaluated

A. ORB-SLAM2

ORB-SLAM2 is a feature-based method that computes the camera trajectory and a sparse 3D reconstruction [11]. It includes the three approaches: monocular, stereo and RGB-D. ORB-SLAM is recognized for its ability to reuse the map, to close the loop and to perform re-localization.

It is structured in three main threads: The tracking thread, localizes the camera every frame, by finding feature matches in the local map and minimizes the re-projection errors by applying motion-only Bundle Adjustment (BA). The local mapping thread manages the local map and optimizes it by performing local BA. Moreover, the loop closing thread detects large loops and corrects the accumulated drifts by performing graph-pose optimization. A general overview of the system is shown in Fig.2. In this study, we focus on the pre-processing input module, Fig.3, in the Tracking thread.

The stereo ORB-SLAM is based on ORB extractor, which is a binary descriptor based on BRIEF [43]. It generates the stereo keypoints with ORB coordinates on the left and the horizontal coordinate of the right match, which are defined as follows: (u_l, v_l, u_r) where (u_l, v_l) are the coordinates in the left image, and u_r is the horizontal coordinate in the right image.

While for an RGB-D input, the Tracking thread extracts the features from the RGB image and for each feature with the coordinates (u_L, v_L), it transforms the depth value d into a virtual right coordinate u_r, as shown by Equ.2. The depth threshold is a coefficient which is multiplied by the baseline to establish a threshold distance over which classification of far and near keypoints is performed. Keypoints are classified as near if their associated depths are 40 times less than the stereo/RGB-D baseline. Otherwise, they are considered far. Near keypoints are devoted to compute scale, translation and rotation since their depth is accurately estimated. On the other hand, far points provide accurate information on rotation, but uncertain information on scale and translation. Far points are triangulated when multiple views support them.

$$u_r = u_l - f_x \cdot \frac{b}{d} \tag{2}$$

Finally, the monocular keypoints are defined by two coordinates $\mathbf{x}_m = (u_L, v_L)$ in the image on the left. These are the points for which a stereo match could not be

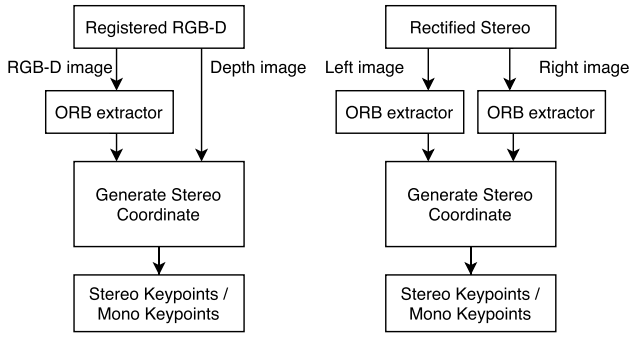


Fig. 3. Input pre-processing module.

found or which have an invalid depth value in the RGB-D case. These points are only triangulated from multiple views and used only to contribute to the estimation of rotation and translation [11].

B. RTAB-Map

RTAB-Map is a graph-based algorithm, fed with RGB-D or stereo input, odometry and extrinsics defining the position of the sensor in relation to the base of the robot. The inputs are then synchronized, and the Short-Term Memory (STM) creates a node that stores the odometry pose, raw sensor data. RTAB-Map has a memory management approach that limits the size of the graph in order to operate in large scale environments. RTAB-Map's memory consists of Working Memory (WM) and Long-Term Memory (LTM). When a node is transferred to LTM, it is no longer available for modules in WM. When RTAB-Map's update time exceeds the fixed time threshold, some nodes in WM are transferred to LTM to limit the size of WM and reduce the update time. The nodes that remain in WM are determined by a weighting mechanism to identify which locations are more reliable than the others. The outputs provided are Map Data which includes the latest added nodes with sensor data and graph, Map Graph, Corrected Odometry, 3D occupancy grid, Dense Point Cloud and 2D occupancy Grid. Fig.4 summarizes the main blocks of RTAB-Map [24]. In this paper, visual odometry was chosen as the input of the RTAB-Map which uses Stereo or RGB-D images as shown in Fig.5 [24]. The Frame-To-Frame (F2F) approach is adopted which registers each new frame against the last keyframe. For feature detection, GoodFeaturesToTrack (GFTT) + ORB are used. For Stereo images, stereo matching is performed using the optical flow based on Lucas-Kanade's iterative method. Feature matching, applies the optical flow directly on the features without computing the descriptors allowing a faster matching. Motion prediction is a model for predicting the location of features in the current frame, based on previous transformations. This limits the search window when matching. This is useful in dynamic environments or with repetitive textures. After the matches are computed, the transformation is calculated by the RANSAC Perspective-n-Point (PnP) method. The resulting transformation is refined by the local bundle adjustment on the features of the last keyframe. Finally, if the number of

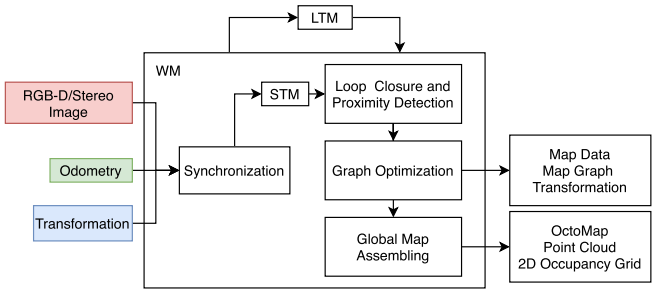


Fig. 4. Block diagram of RTAB-Map.

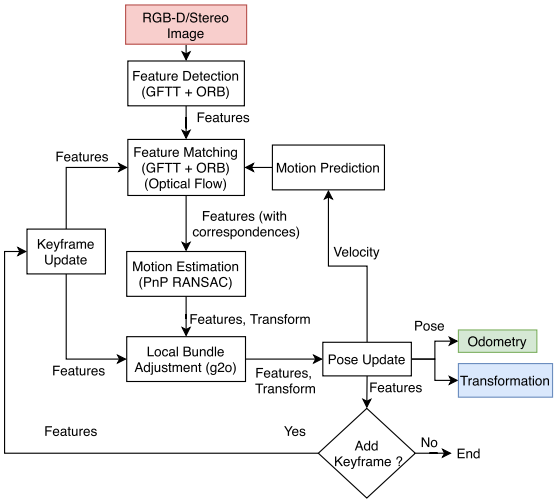


Fig. 5. RTAB-Map's Visual Odometry.

inliers calculated during the motion estimation is below a fixed threshold, the keyframe is replaced by the current frame.

V. Experimental Results

This study was carried out using the Intel RealSense D435i camera. The ORB-SLAM2 and RTAB-Map are run on different datasets. Identification of the best-suited acquisition mode is first investigated. Next, a sensor-algorithm parameters coupling is carried out through a parametric optimization protocol. Then, the depth-based method is compared to stereo based method. The translation and rotation errors are evaluated for each dataset compared to the referenced trajectory. Finally, the effect of the projector on trajectory quality is investigated.

A. Comparison of Depth-Based SLAM and Stereo-Based SLAM Algorithms

In this section, we will compare Depth-based SLAM algorithms against their Stereo-based SLAM version. First, the depth-based SLAM using RGB images and IR images (with the projector off) is tested to find the optimal mode in each environment.

1) Passive IR-D SLAM Vs Passive RGB-D SLAM: The D435i camera is equipped with an RGB camera and two IR cameras. The RGB camera is a rolling shutter type and has a Field of View (FOV) of 69.4° × 42.5°, while the IR camera is a global shutter type and has a Field of View (FOV) of 86° × 57°. We examine the distinction between the RGB

TABLE III
PASSIVE RGB-D SLAM AND PASSIVE IR-D SLAM FRONT-END PARAMETERS FOR ORB-SLAM2

Parameter	RGB-D	IR-D
Number of features	1000/2000	1000/2000
Depth threshold	**20**	**50**
FAST initial threshold	20	20
FAST min threshold	7	7

TABLE IV
PASSIVE RGB-D SLAM AND PASSIVE IR-D SLAM FRONT-END PARAMETERS FOR RTAB-MAP

Parameter	RGB-D & IR-D
Odometry strategy	Frame to Frame (F2F)
Feature detector	GFTT + ORB
Number of features	2000
Motion Prediction (Matching window size)	0 (Global matching)
Motion Estimation (Minimum inliers)	20

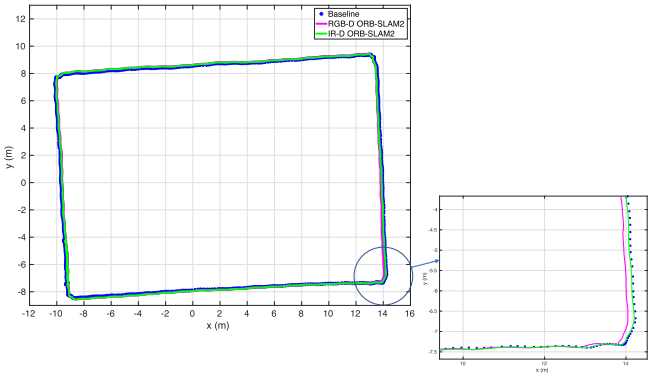


Fig. 6. Passive RGB-D ORB-SLAM2 vs Passive IR-D ORB-SLAM2 in Digiteo_seq1.

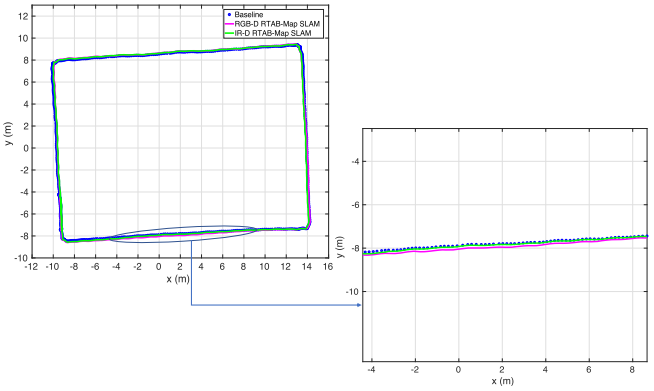


Fig. 7. Passive RGB-D RTAB-Map SLAM vs Passive IR-D RTAB-Map SLAM in Digiteo_seq1.

and IR images regarding the quality of the trajectory. Settings required to ensure proper operation of the algorithm on the dataset are shown in the Table III and IV. The number of features has been set to 1000 for a narrow environment (Digiteo_seq1 & Digiteo_seq2) and 2000 for a wide environment (Digiteo_seq3).

Passive RGB-D ORB-SLAM2 and Passive IR-D ORB-SLAM2 trajectories are plotted on the same Fig.6. Same for the RTAB-Map in the Fig.7. The translational error and the rotational error for Passive RGB-D SLAM and Passive IR-D SLAM are computed with respect to the referenced trajectory. Table V summarizes the results.

TABLE V
TRANSLATIONAL AND ROTATIONAL ERROR OF ORB-SLAM2 AND RTAB-MAP SLAM WITH PASSIVE IR-D AND PASSIVE RGB-D

		Passive RGB-D		Passive IR-D		RTAB error %	ORB error %
		RTAB	ORB	RTAB	ORB		
Digiteo_seq1	Tr (m)	0.13	0.09	0.1	0.08	23.08 ↓	11.11 ↓
	Rot (°)	29.6	15.31	27.98	11.10		
Digiteo_seq3	Tr (m)	0.21	0.28	0.21	0.16	0	42.86 ↓
	Rot (°)	19.46	9.48	19.91	9.39		

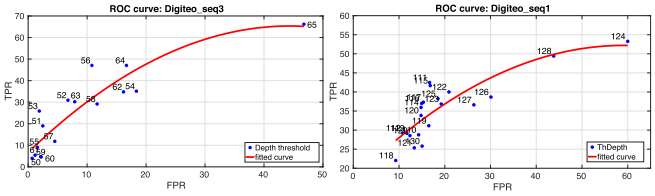


Fig. 8. The ROC curve for different ThDepth values.

TABLE VI
IR-D ERRORS IN THE ORB-SLAM2 AFTER OPTIMIZATION

	Optimal parameter value	ORB IR-D (Tr/Rot Error)	ORB IR-D optimized (Tr/Rot Error)	Error %
Digiteo_seq1	128	0.08m / 11.10°	0.07m / 11.11°	12.5 ↓
Digiteo_seq3	65	0.16m / 9.39°	0.05m / 9.4°	68.75 ↓

According to the results of Table V, it is worth noting that with the IR-D mode, for the ORB-SLAM2, the error is minimized by 12cm for a large-scale scene and 1cm for a narrow environment. For the RTAB-Map, we can see no difference between the two modes in the basement parking environment, whereas the error is reduced by 3cm in laboratory corridors. These results lead us to conclude that the IR-D mode gives better results since it has a wide field of view, allowing a higher parallax and more detection of close features used in the computation of translation, rotation and scaling. Also, it is essential to highlight that both passive IR-D and passive RGB-D modes are correlated by the depth map calculated from the IR images.

2) Depth Threshold Optimization: Since the IR-D mode gave better results in terms of localization, it was retained for an in-depth study according to the depth threshold of the ORB-SLAM2. To this end, we performed an automated test of the different values from 5 to 250, which corresponds by multiplying by the baseline (0.05m) at an interval of 0.5m to 15m. For each sequence, a ROC curve was established, as shown in Fig.8. The curves have been limited to a reduced scope containing the global minimum. The values with the highest TPR and the lowest FPR were selected. The results are shown in Table VI.

It should be noted that the depth threshold varies according to the environment. Also, the optimization has reduced the error by 68.75% in the Digiteo_seq3 dataset and 12.5% in the case of the Digiteo_seq1 dataset. This enhancement clearly shows how sensor-algorithm parameters coupling can significantly affect accuracy.

3) IR-D SLAM Vs Stereo Vision SLAM: Based on the parameters found in Table III, IV, we compare IR-D ORB-SLAM2 and IR-D RTAB-Map to their stereo versions. The goal is to find the rate of improvement in terms of

TABLE VII
TRANSLATIONAL AND ROTATIONAL ERROR OF ORB-SLAM2 AND RTAB-MAP SLAM WITH PASSIVE IR-D AND STEREO

		IR-Stereo		Passive IR-D		RTAB error %	ORB error %
		RTAB	ORB	RTAB	ORB (Optim)		
Digiteo_seq1	Tr (m)	0.14	0.14	0.1	0.07	28.57 ↓	50 ↓
	Rot (°)	29.11	11.08	27.98	11.11		
Digiteo_seq3	Tr (m)	0.21	0.23	0.21	0.05	0	78.26 ↓
	Rot (°)	20.23	9.39	19.91	9.4		

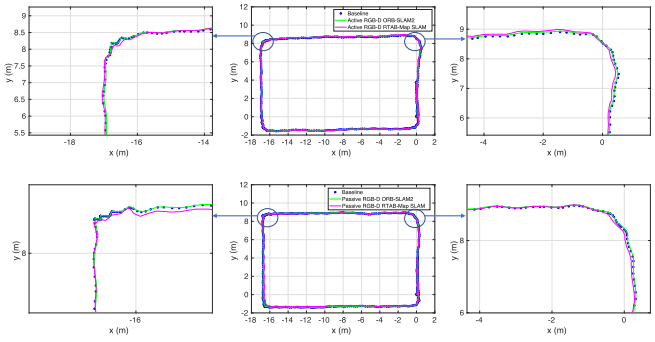


Fig. 9. RGB-D ORB-SLAM2 and RGB-D RTAB-Map SLAM: Active vs Passive mode in Digiteo_seq2.

TABLE VIII
TRANSLATIONAL AND ROTATIONAL ERROR OF ORB-SLAM2 AND RTAB-MAP SLAM WITH PASSIVE AND ACTIVE RGB-D

		Passive RGB-D		Active RGB-D		RTAB error %	ORB error %
		RTAB	ORB	RTAB	ORB		
Digiteo_seq2	Tr (m)	0.11	0.06	0.07	0.04	36.36 ↓	33.33 ↓
	Rot (°)	37.76	13.32	36.87	15.49		
Digiteo_seq3	Tr (m)	0.21	0.28	0.43	0.25	104.76 ↑	10.71 ↓
	Rot (°)	19.46	9.48	33.81	8.46		

accuracy of an RGB-D based algorithm versus a stereo based algorithm. The IR-D SLAM and IR-Stereo Vision SLAM algorithms are compared with the referenced trajectory, where the translational and rotational errors are computed in Table VII.

Table VII shows the translationnal and rotational errors for both modes. For the optimized ORB-SLAM2 IR-D, the error is reduced by 78.26% in the Digiteo_seq3 dataset and by 50% in the Digiteo_seq1 dataset. For RTAB-Map, we have a similar result between IR-D and stereo for the Digiteo_seq3 sequence and a minimal improvement in the Digiteo_seq1 sequence. These results confirm that the sensor's choice is not enough to improve the accuracy, but the algorithm parameters must also be optimized.

B. RGB-D SLAM: Active Vs Passive

In this section, we compare the impact of the IR projector turned on and turned off on the trajectory for two sequences. The results were compared to the referenced trajectory, as shown in Fig.9. The ATE and RPE are calculated in the Table VIII.

By analysing the errors, we can see that the trajectory's accuracy based on active depth is slightly better than that based on passive depth. This slight difference is due to the depth maps, which are denser in the active case allowing more features with a valid depth value. Also, we can see that RTAB-Map, in the case of Digiteo_seq3, requires a study of its parameters to take advantage of the projector.

VI. CONCLUSION AND FUTURE WORKS

We presented the evaluation of various trajectories based on the camera's different modes of acquisition. Passive IR-D SLAM vs Passive RGB-D SLAM was compared, and it was found that the depth threshold parameter for ORB-SLAM2 varies for each camera to get the best results. A method based on the ROC curve was established to find an optimal depth threshold value for the IR sensor. Using an IR camera compared to the RGB camera resulted in a decrease of ATE error by 23.08% for RTAB-Map in Digiteo_seq1 and 82.14% for optimized ORB-SLAM2 in Digiteo_seq3. The use of the D435's IR camera offers a significant advantage as it has a larger field of view for viewing blind spots, and the fact that the depth maps are also aligned with the left IR camera means that no further alignment processing is required. IR images would have been more efficient if they could have been combined with the active depth. Filtering out the IR images patterns is an essential feature to be implemented, especially for SLAM applications. Based on the parameters found in the IR-D vs Passive RGB-D SLAM comparison, the IR-D SLAM algorithm was compared with Stereo Vision SLAM, and we found a decrease in the translational error of 78.26% when using IR-D data for ORB-SLAM2 in Digiteo_seq3 and 28.57% for RTAB-Map in Digiteo_seq1. Finally, we compared the active and passive modes. We found that the active mode gives a more dense depth map; therefore, we get more accurate results since more features are used to calculate translation, rotation and scaling. An RGBD-based SLAM system design must set up a strong coupling of the sensor's algorithm's parameters, especially those related to the field of view, depth threshold, and IR projector. This increases the localization accuracy for robotics applications in indoor environments. In perspective, an in-depth study will focus on the application of the same approach for outdoor environments, in particular to SLAM algorithms based vehicle localization implementing RGB-D sensors.

REFERENCES

[1] H. Lategahn, A. Geiger, and B. Kitt, "Visual SLAM for autonomous ground vehicles," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 1732–1737.

[2] H. Yu, Q. Fu, Z. Yang, L. Tan, W. Sun, and M. Sun, "Robust robot pose estimation for challenging scenes with an RGB-D camera," *IEEE Sensors J.*, vol. 19, no. 6, pp. 2217–2229, Mar. 2019.

[3] J.-C. Trujillo, R. Munguia, S. Urzua, E. Guerra, and A. Grau, "Monocular visual SLAM based on a cooperative UAV–target system," *Sensors*, vol. 20, no. 12, p. 3531, Jun. 2020.

[4] D. Chekhlov, A. P. Gee, A. Calway, and W. Mayol-Cuevas, "Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual SLAM," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality (ISMAR)*, Nov. 2007, pp. 153–156.

[5] M.-F. R. Lee and T.-W. Chien, "Intelligent robot for worker safety surveillance: Deep learning perception and visual navigation," in *Proc. Int. Conf. Adv. Robot. Intell. Syst. (ARIS)*, Sep. 2020, pp. 1–6.

[6] A. Kim and R. M. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 719–733, Jun. 2013.

[7] T. T. O. Takleh, N. A. Bakar, S. A. Rahman, R. Hamzah, and Z. A. Aziz, "A brief survey on SLAM methods in autonomous vehicle," *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 38–43, Nov. 2018.

[8] C.-C. Sun, Y.-H. Wang, and M.-H. Sheu, "Fast motion object detection algorithm using complementary depth image on an RGB-D camera," *IEEE Sensors J.*, vol. 17, no. 17, pp. 5728–5734, Sep. 2017.

[9] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. J. Berlles, "S-PTAM: Stereo parallel tracking and mapping," *Robot. Auton. Syst.*, vol. 93, pp. 27–42, Jul. 2017.

[10] D.-D. Nguyen, A. Elouardi, S. A. R. Florez, and S. Bouaziz, "HOOFR SLAM system: An embedded vision SLAM algorithm and its hardware-software mapping-based intelligent vehicles applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4103–4118, Nov. 2019.
[11] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[12] R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2011, pp. 127–136.
[13] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2014.
[14] C. Sheng, S. Pan, W. Gao, Y. Tan, and T. Zhao, "Dynamic-DSO: Direct sparse odometry using objects semantic information for dynamic environments," *Appl. Sci.*, vol. 10, no. 4, p. 1467, Feb. 2020.
[15] J. Zhao *et al.*, "Visual semantic landmark-based robust mapping and localization for autonomous indoor parking," *Sensors*, vol. 19, no. 1, p. 161, Jan. 2019.
[16] Q. Fu *et al.*, "A robust RGB-D SLAM system with points and lines for low texture indoor environments," *IEEE Sensors J.*, vol. 19, no. 21, pp. 9908–9920, Nov. 2019.
[17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[18] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.
[19] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A system for large-scale mapping in constant-time using stereo," *Int. J. Comput. Vis.*, vol. 94, no. 2, pp. 198–214, Sep. 2011.
[20] D.-D. Nguyen, A. E. Ouardi, E. Aldea, and S. Bouaziz, "HOOFR: An enhanced bio-inspired feature extractor," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 2977–2982.
[21] Y. Liu *et al.*, "Real-time robust stereo visual SLAM system based on bionic eyes," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 3, pp. 391–398, Aug. 2020.
[22] V. Angladon *et al.*, "An evaluation of real-time RGB-D visual odometry algorithms on mobile devices," *J. Real-Time Image Process.*, vol. 16, no. 5, pp. 1643–1660, Oct. 2019.
[23] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 177–187, Feb. 2014.
[24] M. Labbé and F. Michaud, "RTAB-map as an open-source Lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, Mar. 2019.
[25] M. Carfagni, R. Furferi, L. Governi, M. Servi, F. Uccheddu, and Y. Volpe, "On the performance of the intel SR300 depth camera: Metrological and critical characterization," *IEEE Sensors J.*, vol. 17, no. 14, pp. 4508–4519, Jul. 2017.
[26] M. Carfagni *et al.*, "Metrological and critical characterization of the Intel D415 stereo depth camera," *Sensors*, vol. 19, no. 3, p. 489, Jan. 2019.
[27] E. Lachat, H. Macher, T. Landes, and P. Grussenmeyer, "Assessment and calibration of a RGB-D camera (Kinect v2 sensor) towards a potential use for close-range 3D modeling," *Remote Sens.*, vol. 7, no. 10, pp. 13070–13097, Oct. 2015.
[28] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4104–4113.
[29] I. EL BOUAZZAOUI *et al.* (2021). *Digiteo_Seq1*. [Online]. Available: https://data.mendeley.com/datasets/7swv73drgr/3
[30] I. EL BOUAZZAOUI *et al.* (2021). *Digiteo_seq2*. [Online]. Available: https://data.mendeley.com/datasets/tb9g7th9yz/2
[31] I. EL BOUAZZAOUI *et al.* (2021). *Digiteo_seq3_IR-D*. [Online]. Available: https://data.mendeley.com/datasets/2n7j5pg2xj/2
[32] I. EL BOUAZZAOUI *et al.* (2021). *Digiteo_seq3_Stereo*. [Online]. Available: https://data.mendeley.com/datasets/c2gtvyxyt7/2
[33] I. EL BOUAZZAOUI *et al.* (2021). *Digiteo_seq3_Active-Stereo_RGB-D*. [Online]. Available: https://data.mendeley.com/datasets/5xmzkgcgg7/2
[34] I. EL BOUAZZAOUI *et al.* (2021). *Digiteo_seq3_Passive-Stereo_RGB-D*. [Online]. Available: https://data.mendeley.com/datasets/kpps3854xm/2
[35] T. Peng, D. Zhang, R. Liu, V. K. Asari, and J. S. Loomis, "Evaluating the power efficiency of visual SLAM on embedded GPU systems," in *Proc. IEEE Nat. Aerosp. Electron. Conf. (NAECON)*, Jul. 2019, pp. 117–121.
[36] G. Florentz and E. Aldea, "SuperFAST: Model-based adaptive corner detection for scalable robotic vision," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 1003–1010.
[37] L. M. Paz, P. Pinies, J. D. Tardos, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 946–957, Oct. 2008.
[38] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2012, pp. 1643–1649.
[39] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *Int. J. Robot. Res.*, vol. 30, no. 9, pp. 1100–1123, Aug. 2011.
[40] J. L. Schönberger, E. Zheng, J. M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2016.
[41] J. L. Schönberger, T. Price, T. Sattler, J. M. Frahm, and M. Pollefeys, "A vote-and-verify strategy for fast spatial verification in image retrieval," in *Computer Vision* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2017.
[42] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.
[43] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

Imad El Bouazzaoui received the M.E. degree in embedded systems engineering from Cadi Ayyad University, Marrakech, in 2018, the M.S. degree in embedded systems and information processing from Paris-Saclay University, France, in 2019, where he is currently pursuing the Ph.D. degree with the SATIE Laboratory. His research is about sensors characterization, sensors parameters coupling with simultaneous localization and mapping algorithms, and the hardware–software co-design applied to vision-SLAM for autonomous robots and autonomous vehicles.

Sergio A. Rodríguez Florez received the M.S. and Ph.D. degrees from the University of Technology of Compiègne, France, in 2007 and 2010, respectively. Since 2011, he has been an Associate Professor with the University of Paris-Saclay. His research activities are focused on dynamic scene analysis through multimodal perception intended to enhance intelligent transportation systems applications.

Abdelhafid El Ouardi received the M.S. degree from Pierre and Marie Curie University in 2001, and the Ph.D. degree in electronics from Paris-Sud University, France, in 2005. He was with Henri Poincaré University, Nancy, as a Researcher, from 2005 to 2006. He is currently an Associate Professor with Paris-Sud University. With the Embedded Systems Team of the SATIE Laboratory, his research interests include hardware-software co-design, evaluation and instrumentation of embedded systems, design of smart architectures for image and signal processing, and SLAM applications.

Journal of Real-Time Image Processing
https://doi.org/10.1007/s11554-020-00986-9

ORIGINAL RESEARCH PAPER

FPGA implementation of HOOFR bucketing extractor-based real-time embedded SLAM applications

Dai Duong Nguyen[1] · Abdelhafid El Ouardi[2] · Sergio Rodriguez[2] · Samir Bouaziz[2]

Received: 18 September 2019 / Accepted: 12 May 2020

Abstract
Feature extraction is an important vision task in many applications like simultaneous localization and mapping (SLAM). In the recent computing systems, FPGA-based acceleration have presented a strong competition to GPU-based acceleration due to its high computation capabilities and lower energy consumption. In this paper, we present a high-level synthesis implementation on a SoC-FPGA of a feature extraction algorithm dedicated for SLAM applications. We choose HOOFR extraction algorithm which provides a robust performance but requires a significant computation on embedded CPU. Our system is dedicated for SLAM applications so that we also integrated bucketing detection method in order to have a homogeneous distribution of keypoints in the image. Moreover, instead of optimizing performance by simplifying the original algorithm as in many other researches, we respected the complexity of HOOFR extractor and have parallelized the processing operations. The design has been validated on an Intel Arria 10 SoC-FPGA with a throughput of 54 fps at 1226×370 pixels (handling 1750 features) or 14 fps at 1920×1080 pixels (handling 6929 features).

Keywords Features extraction · Parallel image processing · FPGA implementation · Embedded systems

1 Introduction

Keypoints extraction has an essential role in several systems based on cameras, particularly in visual SLAM [1, 2]. There are many different extractors used in various SLAM systems such as: SIFT [3], SURF [4] or ORB [5, 6]. Recently, in our previous research, we proposed an enhanced bio-inspired extractor denoted as HOOFR having a considerable precision with low computational cost [7].

Nowadays, implementing keypoints extraction on modern embedded electronic devices becomes a novel trend attracting many researches. This aim is the first part of implementing a complete SLAM system on an embedded device and requires to take into account the hardware architectures and its specifications. This results in the development of heterogeneous system architectures integrating modern system-on-chip (SoC) designs. Such systems allow us to avoid the issues encountered by multi-core scaling (using several homogeneous cores), stemming mainly from memory wall and von Neumann bottleneck [8]. Designs with such heterogeneous architectures essentially consist of a combination of multi-core processors and a variety of hardware accelerators to speed up the execution of intensive tasks.

A well-known accelerator is a graphical processing unit (GPU). GPUs allow designing architectures for parallel data processing with higher floating point throughput and higher memory bandwidth than processors. These properties make them good candidates to be used in modern computing systems [9]. However, using GPU-based accelerators is inefficient in terms of power consumption [10]. An alternative is a field programmable gate array (FPGA). Modern FPGA devices can provide better compromise between processing speed and energy consumption [11]. Moreover, there is a continuously widening performance gap favoring FPGAs from one generation to the next, especially with regards to high performance computing or data flow constraints. The enhanced performance combined with a superior power efficiency allows increased performance to power efficiency of FPGAs in comparison to both GPUs and CPUs [12].

✉ Abdelhafid El Ouardi
abdelhafid.elouardi@universite-paris-saclay.fr

[1] School of Electrical Engineering, Hanoi University of Science and Technology, Hanoï, Vietnam

[2] Systèmes et Applications des Technologies de l'Information et de l'Energie, CNRS, ENS Paris-Saclay, Université Paris-Saclay, 91405 Orsay, France

Published online: 12 June 2020

One of the main challenge in FPGAs implementations is the complexity of programming these circuits [13]. FPGAs are generally programmed using one of the hardware description languages (HDL) such as Verilog or VHDL used by hardware designers. In practice, these programming languages are complex, hard to analyze and debug so that designers usually spend much time to develop an application [14]. However, this limitation can be tackled by high-level synthesis (HLS), e.g., Vivado HLS and LegUp HLS. HLS enables designers to program an FPGA using high-level languages e.g., C, C++, SystemC or OpenCL. This, in turn, reduces both verification and design time in comparison with HDL.

In this paper, we propose to design a feature extraction system for a SLAM application on an FPGA-based heterogeneous architecture. The implementation is conducted by means of OpenCL programming language [15].

Our contributions are the following:

- Design of a whole feature extraction system, based on HOOFR extractor published in [7], dedicated for SLAM application taking into account the bucketing method.
- A hardware–software codesign approach using OpenCL programming to implement the system on FPGA-based heterogeneous architecture.
- A performance evaluation of FPGA-based implementation versus embedded GPU-based implementation using a publicly dataset.

The rest of the paper is organized as follows: Sect. 2 provides some background of feature extraction system and OpenCL-related work on FPGA. Section 2 describes the optical flow of HOOFR algorithm. Section 3 presents the architecture design on FPGA using OpenCL. Section 4 shows the experiment result. Finally, Sect. 5 concludes this paper and describes future work.

2 Related work and contribution

In the state-of-the-art, several works have previously investigated the acceleration of feature extraction using FPGA [16, 17]. In 2009, Yao et al. [18] proposed an optimized architecture for SIFT feature detection running at 31 ms per frame (640×480) on Xilinx ML507 FPGA. In 2010, Bouris et al. [19] implemented SURF detector on Xilinx Virtex 5 XC5VFX130T FPGA that processed images at 56 fps (~18 ms per frame) with the same resolution. The limitation of these works is that authors implemented only the detection task on hardware while the description task was out of the scope. In 2013, Chiu et al. [20] proposed a parallel hardware design for the whole SIFT extraction. The algorithm is modified to reduce computational amount by 90% and memory usage by 95%, running at 30 frames per second with VGA resolution.

Due to the fact that SIFT and SURF require floating computation, the hardware design of these algorithms performs a slower speed than binary descriptors such as FAST and ORB. Lee [21] presented an ORB extraction system in 2014 operating at 108 fps for 640×480 images. This system, however, did not consider the whole ORB algorithm when missing Harris filtering step. Another ORB system is proposed by Weberruss et al. [22] in 2017, running on Altera Arria V with throughput equivalent to 72 fps at 1920×1080 pixels or 488 fps at 640×480 pixels. Despite mentioning ORB, they employed Harris for detecting keypoints. It is worth noting that the original idea of ORB uses Harris score to filter keypoints only after FAST detection. An alternative of ORB implementation on FPGA was presented by Sun [23] where the performance is 42 fps with 1000 features in full-HD images. The proposed architecture is tested on Zynq-family FPGA.

There are many researches investigating the FPGA acceleration by OpenCL on various algorithms. As an example, Pu et al. [24] experiment KNN algorithm on FPGA-based heterogeneous architecture. OpenCL is used to program Stratix IV 4SGX530 FPGA from Altera. The performance was compared to Intel Core i7-3770 processor and an AMD Radeon HD7950 graphics card where the authors found that FPGA-based implementation was more power efficient. In 2017, Muslim et al. [25] evaluated the OpenCL implementation on Xilinx Virtex-7-series FPGA of three well-known algorithms: KNN, Monte Carlo for financial models and Bitonic sorting. A comparison in terms of execution time, energy and power consumption with some high-end GPUs was done as well. The authors also concluded that FPGAs are much more energy efficient in all the test cases and can sometimes be faster than GPUs. In 2018, Lou et al. [26] presented a research using OpenCL programming to design an accelerator for convolutional neural network on heterogeneous computing CPU-FPGA. The performance of the whole CNN task has been hence improved 2 times in comparison with CPU-only version. In 2019, Zhang et al. [27] also employed an FPGA based OpenCL programming to optimize the time-consuming convolution layer in deep learning. The authors got a former with 8–40 times higher in terms of performance than the corresponding optimization program provided by Xilinx.

Nevertheless, to the best of our knowledge, there is not a total system of feature extraction on FPGA using HLS with OpenCL programming until the present. Moreover, all extraction systems above are developed for a naive implementation, and it is not enough to have a high precision for SLAM applications. In practice, often image-based SLAM systems rely on bucketing method in order to extract keypoints [1, 28, 29]. None of the researches above considered

this method into the optical flow.HOOFR bucketing extraction

2.1 FAST detection filtered by Hessian matrix

FAST [30] considers the points on the circular ring around one pixel. In case of enough consecutive pixels on the ring which are brighter or darker than the central pixel with a threshold t, this latter pixel is considered as a corner. The number of consecutive pixels is generally set between 9 and 12 depending on the application. FAST-9 is employed in HOOFR. Due to the fact that FAST provides a significant number of features, the Hessian response is used to filter the result.

$$H = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \quad (1)$$

The Hessian matrix illustrated in Eq. 1 consists of the second-order partial derivatives of the image. The eigenvectors of this matrix form an orthogonal basis highlighting the local direction of the gradient. If the product of eigenvalues of the Hessian matrix is positive, a local maxima is present. For any square matrix, the product of eigenvalues is the determinant of the matrix. HOOFR proposes to use this determinant as the score of the feature point. In practice, each element of this matrix is generated by applying a square filter with the dimension of n^2 ($n = 7$ in our implementation) as shown in Fig. 1 corresponding to the second-order derivative of the Gaussian smoothing function. If there are more than K points detected by FAST, we only keep the Kth points which exhibit the highest score.

2.2 HOOFR descriptor

HOOFR descriptor is inspired by FREAK which was originally proposed in [31] by considering human retina topology and neuroscience observations. It is believed that human retina extracts information from the visual field by using the Gaussian comparison (Difference of Gaussians) of various sizes and by encoding these differences in binary mode as a neural network.

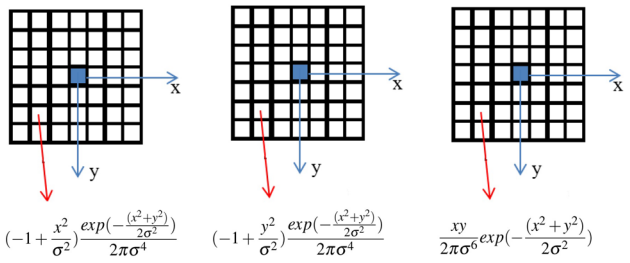


Fig. 1 Square filters for calculating Hessian matrix in HOOFR

2.2.1 Sampling pattern

HOOFR proposes a sampling pattern composed of 6 concentric circles as illustrated in Fig. 2. Each circle has 8 points representing 8 receptive fields distributed as the 8-segment method in DAISY [32]. Therefore, including the keypoint at the center, this pattern contains 49 receptive fields in total. The pixel intensity of each receptive field is smoothed by a Gaussian filter which radius is decreased exponentially with respect to the distance between the receptive field and the keypoint. Comparing to original proposition of FREAK, HOOFR configuration increases, in addition to the radial overlap, the amount of circumferential overlap among the fields. The justification for the proposed configuration is that for complex image processing tasks, various descriptors exploit, either in the image space [33] or in the frequency domain [34], a certain degree of overlapping in order to be able to grasp more effectively complex correlations. Due to the fact that the comparisons between these receptive fields are used to build the binary descriptor, with 49 fields, HOOFR has more pairs (1176 pairs) to choose than that of original FREAK (903 pairs).

2.2.2 Keypoint orientation

In order to estimate the keypoint orientation, HOOFR uses the summing of local gradients over selected pairs. In detail, HOOFR uses 40 pairs with symmetric receptive fields with respect to the center as shown in Fig. 3. The orientation is then obtained by Eq. 2 where S is the set of all 40 pairs used to compute local gradients, N is the number of pairs in S and $P_0^{r_1}$ is the 2D vector of coordinates of the receptive field center.

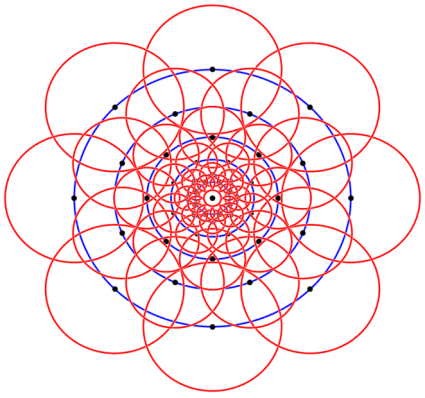


Fig. 2 Sampling pattern of HOOFR Gaussian filter for each receptive field is represented by red circles

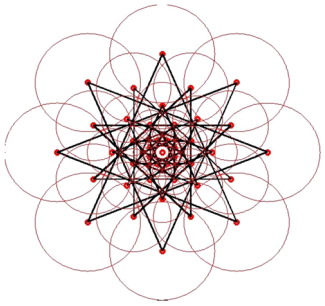


Fig. 3 Illustration of selected pairs to estimate the orientation in HOOFR

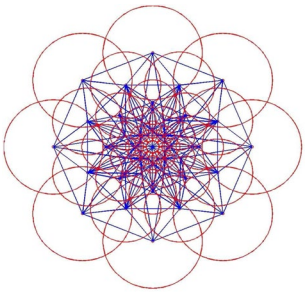


Fig. 4 Illustration of 256 selected pairs used to construct the descriptor in HOOFR

Similar to FREAK, the space of orientation in HOOFR is also discretized in order to accelerate the computation. In practice, keypoint orientation takes only 256 discrete values between 0 and 2π. This discretization allows us to compute beforehand the coordinates of receptive fields for all orientations and save them into a lookup table. Hence, these coordinates will not be re-computed while building the description for each keypoint.

$$O = \frac{1}{N} \sum_{P_0 \epsilon S} (I(P_0^{r_1}) - I(P_0^{r_2})) \frac{P_0^{r_1} - P_0^{r_2}}{\left\| P_0^{r_1} - P_0^{r_2} \right\|} \tag{2}$$

2.2.3 Descriptor

Binary descriptor F is constructed based on the comparison between receptive fields with their corresponding Gaussian kernel.

$$F = \sum_{0 \le n < N} 2^n T(P_n) \tag{3}$$

$$T(P_n) = \begin{cases} 1 & \text{if } (I(P_n^{r_1}) - I(P_n^{r_2})) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where P_n is the pair of receptive fields, N the size of binary descriptor, $I(P_n^{r_1})$ and $I(P_n^{r_2})$ are, respectively, the Gaussian smoothed intensities of the first and the second receptive field of the pair n. HOOFR builds a descriptor of size 256 bits. The 256 most relevant pairs (Fig. 4) are selected among the total of 1176 pairs through offline-analysis algorithm.

2.2.4 HOOFR pattern approximation

In HOOFR, as described above, the pixel intensity of each point in sampling pattern is smoothed by a Gaussian filter which is time-consuming. In practice, the smoothing is inspired of the approximation used in scale-space representation in SURF when we use mean intensity that approximates Gaussian derivatives. During the experiments, this approximation provides a competitive precision with a high computing acceleration due to the use of integral image. Figure 5 shows the shape of the fields used to compute mean intensity for each pattern point.

2.3 Bucketing extraction

Our design is intended for SLAM applications in which bucketing extraction is an essential factor. The reason is that almost SLAM systems require a homogeneous distribution of features to have a high precision of localization result [1, 28, 29]. Image is hence divided into a grid as showed in Fig. 6 where the number of cells (NUMBER_OF_CELLS) depends on the image resolution.

On each image cell, HOOFR is employed to extract features and the optical flow of HOOFR algorithm as illustrated in Fig. 7. The maximum number of keypoints returned from one cell is limited to POINTS_PER_CELL. In the case that FAST returns more keypoints on a cell, only POINTS_PER_CELL points having the highest Hessian score are maintained. Each block in HOOFR algorithm is suitable to be implemented on FPGA. FAST detection simply uses the comparison between pixel intensities to determine a relevant

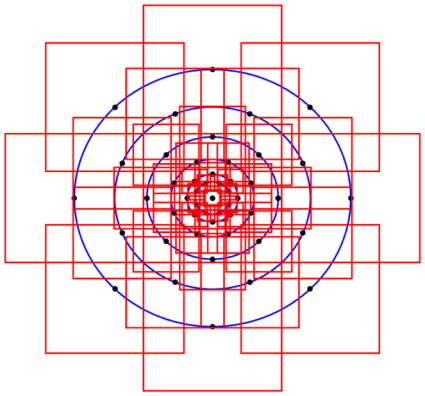


Fig. 5 Approximation of HOOFR sampling pattern

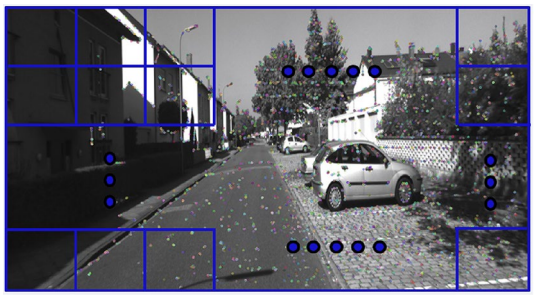


Fig. 6 Image divided into grid

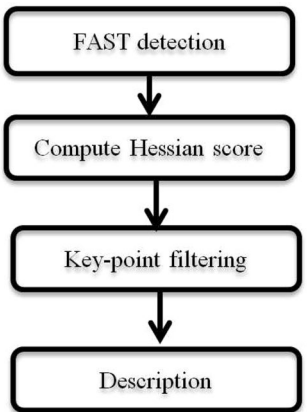


Fig. 7 HOOFR extractor optical flow

point. In Hessian score computation, the coefficient of Hessian matrix is floating, but it can be converted to integer by multiplying a factor of 10^k. In HOOFR, k is set typically to value of 3. Keypoints filtering also involves the comparison between Hessian score. Finally, the computation inside description block is between integer values and keypoint descriptors are composed of binary values after the comparison between neighbor pixel intensities around the central point. Another important remark is that the processing of each cell is independent to other cells so that the parallelization could be done at the image cell level.

3 High level synthesis of HOOFR extraction on FPGA-SoC architecture

3.1 OpenCL programming overview and design flow

OpenCL is based on C99 and provides both data-parallel and task-parallel programming models. It nowadays becomes a well-known parallel language for programming various computing architectures such as CPUs, GPUs, FPGAs as well as DSPs. OpenCL adopts the processor-accelerator concept. The OpenCL platform model includes both a (possibly multi-core) CPU (called host) and massively parallel accelerators (called devices) that can take the form of GPUs, FPGAs or DSPs. The host could set up the environment to enable the OpenCL kernels to execute on one or more devices. When the applications starts, the host executes the host program and passes high computational workload to the device. The device is programmed using kernel sources, executes this workload and passes results back to the host.

Figure 8 shows device architecture and its memory model seen from host in OpenCL. A device consists of compute units (CU), each further divided into processing elements (PE). Several concurrent executions of the kernel body (called work-items) take place on multiple processing elements. The work-items are grouped into work-groups, which are being executed by compute units. The memory is broadly divided into host memory and device memory. The device memory is further structured into private memory (specific to each work-item), local memory (shared by all the work-items in the same work-group) and a global memory (shared by all the work-groups). There are two types of global memory: Off-chip and on-chip. Off-chip global memory typically resides in external DRAM. It is the largest in size but the slowest to access. Data transfer from host to device always take place on off-chip global memory. Otherwise, on-chip global memory is not seen by host processor. Its capacity is much smaller than off-chip memory but provides the data communication cross-kernel without host intervention. Private memory (typically allocated to register files) is the fastest to access but the smallest in size. Local memory also resides in on-chip SRAM, having a compromise between size and accessing time.

OpenCL program is implemented on ALTERA FPGA SDK using AOCL synthesizer. Our design flow is shown in Fig. 9. In the first step, host and kernel codes are developed in parallel to warrant the conjunction between kernel interface and kernel calling of the host. Then, the functional verification is done using FPGA SDK for OpenCL emulator.

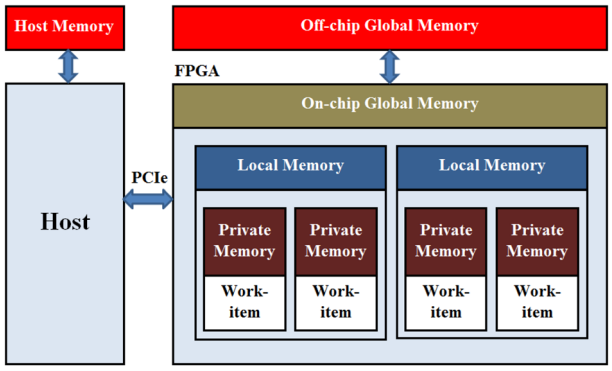


Fig. 8 OpenCL memory model

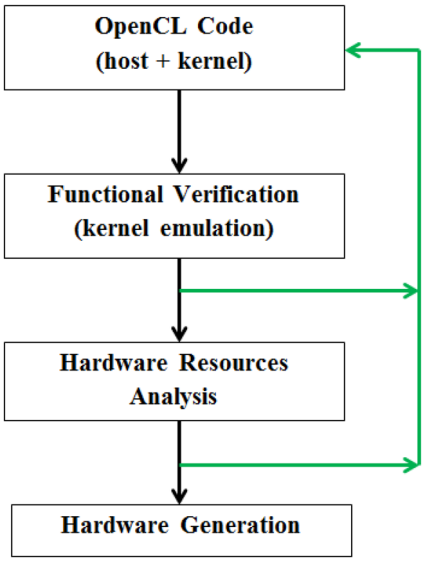


Fig. 9 Design flow

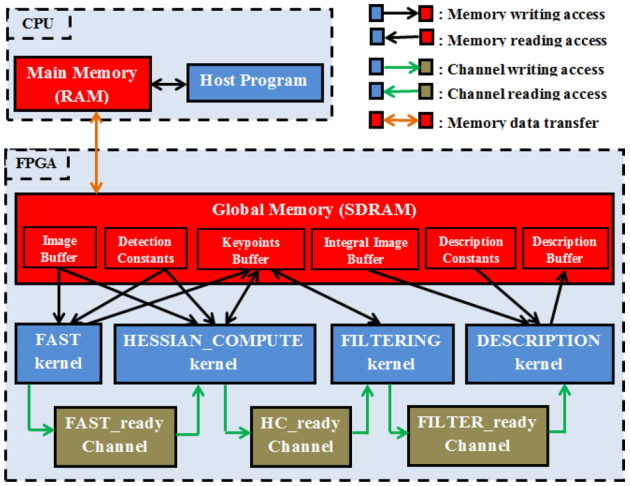


Fig. 10 HOOFR extractor: algorithm-architecture mapping

This feature allows us to test the functionality and to iterate on the design without executing it on FPGA material each time. The emulation of our design is run on x86-64 Ubuntu 14.04. Once the functionality is verified, the hardware resource usage for all kernels is estimated. This step requires a specific FPGA architecture to be designed and an Arria 10 GX is selected for the analysis. After the resources estimation, in the case that kernels take too much resources or the design is not suitable to be implemented on the target platform, we return to the first step to modify and optimize the design. Finally, hardware implementation is generated and is loaded to the target board to evaluate the performance.

3.2 HOOFR extractor partitioning on CPU-FPGA architecture

HOOFR extractor is divided into 4 functional blocks (FAST, HESSIAN_COMPUTE, FILTERING and DESCRIPTION) with respect to the algorithm process. This decomposition is based on an analysis of the data flow to achieve a compromise between consuming resources (memory, logic elements) and processing speed. For details, FAST block is to detect FAST features in the images. HESSIAN_COMPUTE is to compute Hessian score for all keypoints returned by FAST detector. Keypoints are then filtered in the FILTERING block to keep the relevant ones based on their Hessian scores. Finally, the DESCRIPTION block builds 256-bit HOOFR descriptor for all relevant keypoints after the filtering task.

CPU-FPGA system for HOOFR extractor is shown in Fig. 10. Four pipelined blocks are implemented on FPGA. Each functional block is programmed as one kernel, and all kernels are launched concurrently. CPU plays a role of a controller and computes integral image required in description block. Noting that the computation of integral image is irrelevant to be executed on FPGA device in OpenCL design as it could be realized rapidly by one query pixel-to-pixel. Otherwise, this operation is suitable on CPU side. Hence, a partitioning is proposed as demonstrated in Fig. 11 in order to make use of the computing resources. As we can see, for each input image, CPU firstly transfers the image to FPGA off-chip global memory. Then, CPU launches consecutively the three detection kernels. The DESCRIPTION kernel will be launched only when the integral image has already been computed and transferred to FPGA from CPU. This partitioning allows us to employ CPU and FPGA resources in parallel. There is no interruption on CPU after launching kernel. A synchronization occurs only when all kernels are active and CPU awaits until FPGA finishes the extraction. This synchronization is present to ensure that valid results are ready to be reloaded to CPU from FPGA.

To have a high precision in SLAM applications, bucketing detection is always employed to warrant the homogeneous keypoints distribution. It means that image is divided into a grid and a specific number of keypoints is aimed to be extracted for each image cell. Hence, the pipeline is achieved at the level of image cells. When a kernel finishes its task for one image cell, the next kernel starts to work immediately on this image cell as shown in Fig. 12. The communication control between kernels is done using Altera channel extension for passing data and for synchronizing kernels with low latency. The implementation of a channel resides on on-chip global memory and allows kernels to communicate directly with each other via FIFO buffers. Data movement across kernels is coordinated without host intervention.

OpenCL does not warrant the execution order of work-items. Therefore, the execution order of image cells is

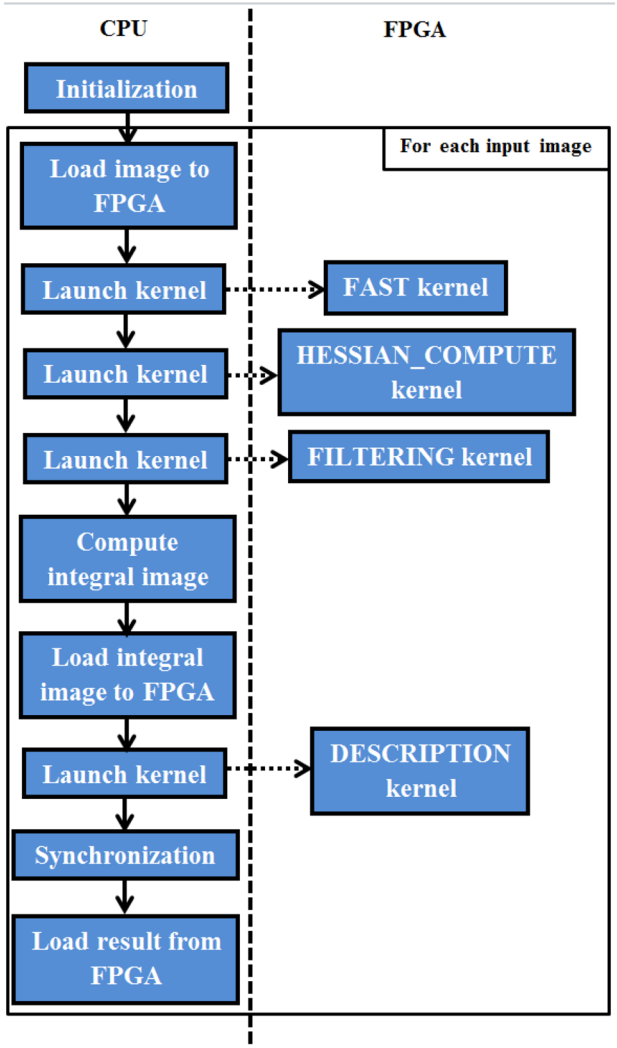


Fig. 11 CPU-FPGA execution flowchart

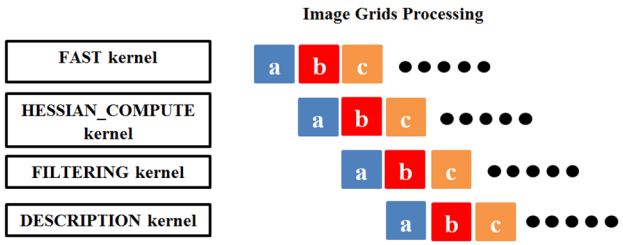


Fig. 12 Pipeline kernel processing

undefined. In Fig. 12, the three cells a, b, c correspond to cells 0, 1, 2 in the image. In practice, when FAST kernel is finished on cell "a", it writes the identification of cell "a" to FAST_ready channel. HESSIAN_COMPUTE kernel reads this identification and launches the processing for cell "a". The procedure continues by a similar way for the other kernels. We denote this design as "pipeline of pipeline" due to the fact that inside each kernel, work-items are also parallelized following the pipeline natural characteristic of FPGAs.

3.3 FAST kernel

The intensity of 16 pixels are compared to the intensity of the reference pixel. Each comparison takes one of four states: darker, brighter, not darker or not brighter. In practice, the smallest data type supported in OpenCL programming is 8-bits (char or unsigned char). Hence, we put the comparison result of 16 pixels into 4 elements (fast8_d1, fast8_d2, fast8_b1, fast8_b2) of the 8-bits type. Each bit of fast8_d1 and fast8_d2 indicates that the pixels are darker (value = 1) or not (value = 0) while each bit of fast8_b1 and fast8_b2 indicates that the pixels are brighter (value = 1) or not (value = 0).

The advantage of FAST detection is that the segmentation test could be employed to accelerate the processing. It means that the feature verification could proceed to some tests to ignore rapidly a pixel. HOOFR extraction used FAST-9 where a central pixel is considered as a feature when it is darker (dark feature) or brighter (bright feature) than at least 9 consecutive points in Bresenham circle. The segmentation test can be done with 8 symmetric pairs. In fact, if a central pixel is a dark feature in FAST-9, the central point must be darker than at least one of two pixels in a symmetric pair. It is applied for all 8 symmetric pairs in Bresenham circle. The condition is similar to the case of bright feature.

OpenCL implementation of FAST detection on FPGA is shown in Algorithm 1. After the segmentation test, central pixel is ignored in case of negative sign (dark = 0 and bright = 0). Otherwise, when positive sign is found (dark = 1 or bright = 1), two 8-bit variables will be concatenated to form a 16-bit variable. The function Verify_FAST_corner takes this 16-bit variable to check the feature condition. The central pixel is added to features_list if the presence of 9 consecutive darker or brighter pixels is valid.

The number of work-items launched for the kernel is equal to the number of cells in the image. Each work-item works on one image cell where the coordinates are determined by one top-left (tl) pixel and one bottom-right (br) pixel. The boundaries for image cells are fixed. They are precomputed and are saved to grid_coors array in the initialization step. At the end of the kernel, the work-item writes the identification of grid to FAST_ready_channel. From that, the next step knows which cell is ready for processing.

After FAST kernel, FAST features are added to features_list. However, the features_list is a global array used for all image cells, and the issue is that the number of features in each cell is different from those of others cells. To avoid a memory conflict, separated zones are created for each image

cell in features_list. Noting that the maximum number of features in one cell is equal to the number of pixels, features_list array is hence created with *N_CELLS* × *RES* elements where *N_CELLS* is the number of image cells and *RES* is the number of pixels (resolution) in the biggest cell. Each element is composed of three factors (x, y, score) corresponding to 2-D coordinates of the feature in the image and its Hessian score. The Hessian score will be computed in the next kernel. Each image cell with an identification (id) will work on the memory zone from the position at (id × RES) to the position at (id+1)*RES in features_list.

Algorithm 1 FAST kernel

```
declare global arrays: img, features_list, grid_coors;
function KERNEL: FAST
   declare 8-bit private variables : fast8_d1,
fast8_d2,fast8_b1, fast8_b2;
   declare 16-bit private variables : fast16;
   ptidx ← get_global_id(0);
   image_cell ← get_Image_Cell(grid_coors, ptidx);
   num_ktps ← 0;
   For each pixel in image_cell do
      /////segmentation test//////////
      p ← Get_intensity(pixel);
      dark ← 1;
      bright ← 1;
      For i from 0 to 7 do // for each pair of 8 symmetric
pairs
         p1 ← Get_intensity(bresenham_circle[i]);
         p2 ← Get_intensity(bresenham_circle[i+8]);
         if(dark ==1)
            fast8_d1 ← set_bit(p,p1,i);
            fast8_d2 ← set_bit(p,p2,i+8);
            dark ← Segmentation_test(fast8_d1,fast8_d2);
         end if
         if(bright==1)
            fast8_b1 ← set_bit(p,p1,i);
            fast8_b2 ← set_bit(p,p2,i+8);
            bright ← segmentation_test(fast8_b1,fast8_b2);
         end if
         if ((dark ==0) && (bright==0)) break; end if
      end for
      if((dark ==0) && (bright==0)) go_to_next_pixel; end
if
      ////////////////////////////////////
      ///////////verify corner/////////
      if(dark || bright)
         if(dark)
            fast16 ← Concatenation (fast8_d1, fast8_d2);
         else
            fast16 ← Concatenation (fast8_b1, fast8_b2);
         end if
         test_corner ← Verify_FAST_corner(fast16);
         if(test_corner)
            features_list ← Add_to_list (pixel_coordinates);
            num_ktps++;
         end if
      end if
      ////////////////////////////////////
   end for
   write_channel_intel(FAST_ready_channel,{ ptidx,
num_ktps}); end function
```

3.4 HESSIAN_COMPUTE kernel

As shown in Algorithm 2, before computing the Hessian score, a work-item of HESSIAN_COMPUTE kernel must call read_channel_intel function to get from FAST_ready_channel an identification (ptidx) of an image cell and its

number of FAST features (num_ktps). The oldest identification in the channel will be returned since AOCL channel is in type of FIFO array. The implementation of read_channel_intel function is blocking so that the processing will wait until an identification is successfully read. Following HOOFR algorithm, Hessian computation is simply applying three 7×7 Gaussian square filters on the feature, and it is performed for all FAST features in the image cell. The features_list will be updated with the computed Hessian score.

Algorithm 2 HESSIAN_COMPUTE kernel

```
declare global arrays: img, features_list;
function KERNEL: Hessian_Compute
                {ptidx,          num_ktps}          ←
read_channel_intel(FAST_ready_channel);
   For i from 0 to num_ktps do
      feature ← Get_pixel_from_list(features_list, i);
      hessian_score ← Compute_Hessian_score(feature);
features_list ← Update_features_list(hessian_score);
   end for
   write_channel_intel (HC_ready_channel, {ptidx,
num_ktps}); end function
```

Similarly to FAST kernel, the work-item writes the identification of image cell to HC_ready_channel at the end of function to communicate with FILTERING kernel.

3.5 Module duplication

During experiments, we found that FAST kernel and HESSIAN_COMPUTE kernel are bottle-necks of the algorithm flow. These two kernels do not consume much logic resources but take much time to operate. Despite the advantage of the FAST segmentation test allowing to reject rapidly the non-valid features, the test of the whole image (for example: 453,620 pixels with the dimension of 1226×370 pixels) makes FAST kernel become costly. HESSIAN_COMPUTE kernel works only on pixels considered as FAST keypoints. However, FAST detection returns many keypoints and Hessian score computation for each keypoint is costly so that HESSIAN_COMPUTE kernel is also time consuming. To accelerate the processing, we duplicate these two blocks.

There are two ways for the duplication: using num_compute_units attribute or physical duplication. For the first method, the value of num_compute_units is set to 2 in the declaration of the kernel function. The work-items are scheduled automatically to execute on 2 compute_units with uncontrolled ordering. However, AOCL tools only support the channel implementation with single compute_unit kernel. Hence, physical method is used in our design.

As shown in Fig. 13, two identical kernel functions are created for each duplicated block with exactly the same interface except the function name. To avoid the memory conflict, each

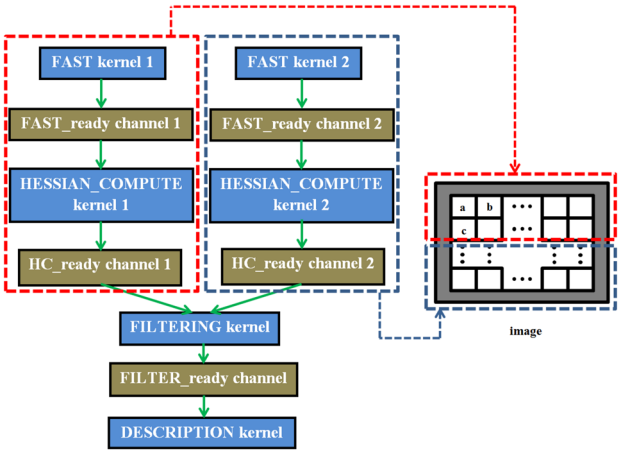


Fig. 13 Kernels duplication schema

function is called from the host to work on separated image zone: one for the first half and one for second half of the image. Following the instruction of AOCL tool consisting that one kernel can read and write to multiple channels, but one channel can be read and written from only one kernel, the FAST_ready_channel and HC_ready_channel are also duplicated for kernel communication on each image zone.

3.6 FILTERING kernel

This kernel is the last step of detection phase, it uses FILTERING_ready_channel to communicate with DESCRIPTION kernel. To read from HC_ready_channel, due to the fact that this channel is duplicated, we must use non-blocking channel reads as shown in Algorithm 3 to get one image cell identification from two separated FIFO chains.

This kernel is aimed to keep a limited relevant features in one image cell. The maximum number of keypoints is defined by POINTS_PER_CELL. With the same objective of avoiding the memory accessing conflict, we declare an array called filtered_features_list. An image cell occupies POINTS_PER_CELL individual positions in this array. In total, filtered_features_list is the size of POINTS_PER_CELL*N_CELLS. Each element is in the same form with the elements of features_list containing information about the coordinates (x, y) and Hessian score of a feature.

Algorithm 3 FILTERING kernel

```
declare global arrays: features_list, filtered_features_list;
function KERNEL: Filtering
    declare private variables : hessian_min,
filtered_num_elements;
    valid ← false;
    while(!valid) do
                        {ptidx,          num_ktps}          ←
read_channel_nb_intel(HC_ready_channel, &valid);
                  if(!valid)    {ptidx,    num_ktps}    ←
read_channel_nb_intel(HC_ready_channel_2,        &valid);
end if;
    end while
    filtered_num_elements ← 0;
    For i from 0 to num_ktps do
        hessian_score ← get_score(features_list, i);
        if ( (filtered_num_elements < POINTS_PER_CELL) ||
(hessian_score > hessian_min) )
                        {filtered_features_list,     hessian_min,
filtered_num_elements} ← Update_filtered_list (features_list,
i);
        end if
    end for
    write_channel_intel (FILTERING_ready_channel,
{ptidx, filtered_num_elements}); end function
```

The number of relevant keypoints (filtered_num_elements) is initialized to zero. For every FAST feature detected in the image cell, the filtering procedure in Update_filtered_list function is described as follows:

- If *filtered_num_elements* is smaller than POINTS_PER_CELL, the feature is added to *filtered_features_list* and *filtered_num_elements* increments by one.
- When *filtered_num_elements* attains the value of POINTS_PER_CELL, filtered_features_list is queried to find the position which contains keypoint having the smallest Hessian score (*hess_min*).
- Then, for each new feature, its Hessian score is first compared to hessian_min. If its score is smaller than hessian_min, it is discarded rapidly without changing the *filtered_features_list*. In contrast, when its score is bigger, it is added to the list and a new *hessian_min* is determined by query *filtered_features_list* once again.

3.7 DESCRIPTION kernel

The DESCRIPTION kernel is shown in Algorithm 4. The processing task of each feature consists of two parts: orientation estimation and binary descriptor construction. The variable pattern_points is an array which contains the intensity of surrounding pixels used to describe the central pixel. In each part, the intensities of surrounding points are firstly smoothed by a Gaussian. In HOOFR, to have a high efficiency between precision and timing, this smoothing is approximated by the mean intensity requiring the use of the integral image of the original input image. This integral image is computed by CPU and is loaded to global array *imgintegral* before this kernel is launched.

Features description is saved to a global array denoted as *descriptors*. The structure of *descriptors* is a 2-D array of 32-bit elements where the number of rows is equal to the number of elements in filtered_features_list and the number of columns is 8. In practice, each row is a 256-bit descriptor of one feature. Each image cell will describe its own features and save result to the rows from the position ptidx*POINTS_PER_CELL to the position (ptidx+1)*POINTS_PER_CELL. Due to the fact that the number of features in each image cell could be varied (from 0 to POINTS_PER_CELL), some unused rows could exist. Hence, the quantity of features must be saved to a global array called num_kpts_list to determine the useful rows in each memory zone.

After DESCRIPTION kernel, three global arrays (*filtered_features_list*, *descriptors* and *num_kpts_list*) are uploaded back to CPU to regroup the information.

Algorithm 4 DESCRIPTION kernel

```
declare global arrays: imgintegral, filtered_features_list,
descriptors, num_kpts_list;
function KERNEL: Description
    declare private variables: keypoint, keypoint_angle,
pattern_points, keypoint_descriptor;
                {ptidx,          num_ktps}          ←
read_channel_intel(FILTERING_ready_channel);
    For i from 0 to num_ktps do
        keypoint ← Get_keypoint(filtered_features_list, i)
        //// compute orientation ////
        pattern_points ← Gaussian_smooth(imgintegral, key-
point, 0);
                        keypoint_angle      ←      Com-
pute_keypoint_angle(pattern_points);
        ////////////////////////////////////
        ////compute descriptor/////
        pattern_points ←
Update_Gaussian_smooth(imgintegral, keypoint,
keypoint_angle);        keypoint_descriptor ←
Make_description(pattern_points);
        descriptors ← Add_to_descriptors_list(keypoint_descriptor);
        ////////////////////////////////////
    end for
    num_kpts_list ← Add_to_num_ktps_list(num_ktps); end
function
```

Table 1 FPGA resource usage. *ALUTSs* Adaptive Look-Up Tables, *FFs* Flip Flops, *RAMs* Random Access Memory blocks, *DSPs* Digital Signal Processing blocks

Kernel name	ALUTs	FFs	RAMs	DSPs
FAST	21,021	29,762	242	4
Hessian_Compute	11,736	18,193	122	9
Filtering	11,485	24,003	180	1
Description	59,820	73,948	376	40
Channel resources	230	1094	5	0
Total	**104,292**	**147,000**	**925**	**54**
(**no duplication**)	(**23%**)	(**17%**)	(**51%**)	(**3%**)
FAST_2	21,021	29,762	242	4
Hessian_Compute_2	11736	18,193	122	9
Total	**137,049**	**194,955**	**1289**	**67**
(**duplication**)	(**31%**)	(**22%**)	(**71%**)	(**4%**)
Available	**448,160**	**896,320**	**1805**	**1633**

4 Experiment results

4.1 Resource usage

Our design was synthesized on Arria 10 SoC SX660 architecture including a dual-core ARM Cortex-A9 processor (1.5 GHz) and an FPGA (operating at 100 MHz) with 660K LEs. The version of AOCL tool is 17.0. As shown in Table 1, availability of the resources in Arria 10 SX660 does not constrain any design model (with or without duplication). Our description kernel consumes the most resources, and it is much more costly comparing to the description module in [35] or [22]. The reason is that processing complexity of the HOOFR algorithm was respected in our design where the keypoint orientation and keypoint descriptor are generated in description module. Moreover, noting that instead of using raw value as in BRIEF, pixel intensity in HOOFR flow is filtered to be robust to image noise. As a result, description kernel takes more resources to handle its task.

4.2 Timings

The hard processor system (HPS) and the FPGA are coupled via a high-bandwidth interface built on high-performance ARM AMBA AXI bus bridges. IP bus masters in the FPGA have access to HPS bus slaves via the FPGA-to-HPS interface. In the same way, HPS bus masters have access to bus slaves in the FPGA via the HPS-to-FPGA bridge. So, HPS–FPGA bridges allow the FPGA to achieve communications with the HPS slaves, and vice versa. The hard processor system integrates an ARM Cortex-A9 MPCore processor. The ARM AMBA AXI 32/64/128 bus bridges supports simultaneous read and write operations. The ARM-FPGA interface supports over 128 Gbps peak bandwidth between the processor and the FPGA. The results of the hardware–software partitioning and performance evaluation presented in this paper are produced according to this bandwidth.

We tested our design on the images of the 16th sequence of KITTI dataset. The image size is of 1226×370 pixels. However, to evaluate the effect of image resolution on processing time, we re-scale the original images to the different resolutions. The scaling was done using resize function in OpenCV. Then, for each resolution, extraction time was achieved by mean value after launching 100 times. As shown in Table 2, our design attains a frequency of 54 frame per second (fps) at the original scale (1226×370), generating an average of 1750 keypoints per image. At full-HD scale (1920×1080 pixels), we obtain a frequency of 14 fps with 6929 keypoints per frame.

In the reference [23], the authors demonstrated that their design achieves 42 fps with ORB extractor. However, their proposed design dealt with only 1000 keypoints, and the algorithm was extremely simplified by changing Harris score to SAD score or changing Gaussian smooth to Binominal smooth. Besides, authors used score only for 3×3 non-maximal suppression, but this is not the original idea of ORB inventor. In the original invention, Harris score is aimed to filter keypoints in an image zone. If the number of features returned after FAST detection is more than a value K in a zone, only Kth relevant ones having the highest Harris score are kept as done in our design.

Another advantage of our system is that the performance is stable across frames when the maximum number of keypoints in each image zone is limited. In contrast to other systems in the state-of-the-art where the only way to manage the number of keypoints is changing FAST threshold. Given a random image, the FAST threshold is not determined so that the number of keypoints for these systems is unbounded. Otherwise, if designers of these systems control the number of keypoints by stopping the frame processing when N keypoints are found, they could not warrant the homogeneous

Table 2 Timing performance (FAST_threshold = 12, POINTS_PER_CELL = 15)

Resolution	N keypoints	NX × NY	Time (ms)	fps
352×240	360	6×4	4.031	248
350×480	682	6×8	8.210	121
580×480	1074	10×8	11.993	83
720×480	1462	14×8	15.362	65
720×576	1780	14×10	18.590	53
1226×370	1750	24×6	18.247	54
1280×720	3661	24×14	38.262	26
1920×1080	6929	38×20	68.684	14

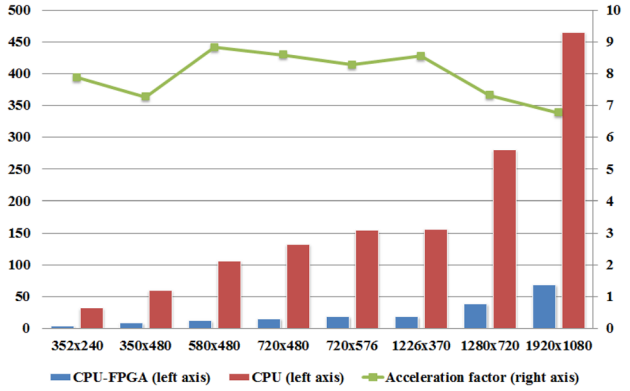


Fig. 14 Acceleration factor on Arria 10 SoC (Right axis: execution time in ms. Left axis: acceleration factor)

keypoints distribution which is very important for a SLAM application.

Figure 14 demonstrates the acceleration on Arria 10 board of our design in comparison with the C++ version running completely on the embedded ARM CPU. The C++ version running on ARM CPU had taken into account the optimization of dual-core CPU using OpenMP programming. Furthermore, the compilation used the flag –O3 which allows loop vectorization. It is obvious that the higher image resolution is, the higher computation cost is. By offloading the processing to FPGA, we could obtain a speedup from 7 to 9 times faster. Execution time of OpenCL version took into account the data transfer between CPU and FPGA which is time-consuming.

It is worth noting that there is a fluctuation of acceleration factor. The reason of the speedup fluctuation is that FAST detection uses segmentation test. It allows FAST detector to discard rapidly a pixel after one or two tests. Moreover, after having relevant points returned by FAST detector, HOOFR algorithm filters these points to obtain keypoints. It means that detection complexity is not constant for each pixel. In the image region where pixels are discarded rapidly (few relevant points), computation complexity is low and the detection on FPGA is not much effective than on CPU. Otherwise, in the region containing many relevant points, computation complexity is high and the detection on FPGA is much effective than on CPU. Therefore, acceleration factor depends highly on the distribution of relevant points in the image; it is not proportional to the resolution.

4.3 Comparison with GPU implementation

Our design is realized using OpenCL which allows an implementation not only on an FPGA but also on various alternative hardware such as a GPU. Here, for comparison, we used a powerful GPU Nvidia Geforce GT 740 containing 384 CUDA cores clocked at 1.0 GHz. The essential difference between FPGA and GPU implementations is that GPUs do not support channel communication so that kernel blocks must be launched sequentially. Table 3 shows a timing comparison between the FPGA and the GPU. As can be seen, FPGA is faster than GPU at low resolution but at higher resolution, GPU becomes faster. The reason is that the GT 740 GPU includes a huge number of CUDA cores. At low resolutions, the number of threads is small so that it does not make use of all computation resources. Otherwise, when the resolution increases, the number of thread increases. All GPU resources are hence employed in the processing, and the GPU becomes faster in our benchmark.

Power efficiency factor is defined as the processing speed given a power energy supply. The FPGA power efficiency $\mathrm{FPGA_{PE}}$ compared to that of the GPU is defined by Eq. 5:

$$\mathrm{FPGA_{PE}} = \frac{\mathrm{GPU_{Power}} * \mathrm{GPU_{Time}}}{\mathrm{FPGA_{Power}} * \mathrm{FPGA_{Time}}} \tag{5}$$

Considering the worst case, with a full-HD resolution (1920×1080), and given the energy consumption for the Arria 10 SoC (21 Watts) and that of the Nvidia GT 740 GPU (64 Watts), it is expected a higher power efficiency of the FPGA design when it is compared to the GPU.

5 Conclusions and future work

In this work, an OpenCL-based FPGA SoC architecture for HOOFR feature extraction has been designed. The complexity of HOOFR algorithm was respected to ensure the robustness. The functional blocks were optimized so that the detection result on hardware is exactly similar to that on software. This feature extraction system integrates bucketing method to warrant the homogeneous distribution of keypoints because it is aimed to use in SLAM applications. Our design was implemented on Arria 10 SoC-FPGA where the version OpenCL is 7× to 9× faster than the C++ version running completely on the embedded ARM CPU. The throughput was 54 fps at 1226×370 pixels or 14 fps at

Table 3 FPGA and GPU processing times comparison

Resolution	GPU time (ms)	FPGA time (ms)
352×240	15.362	4.031
350×480	19.779	8.210
580×480	20.075	11.993
720×480	20.191	15.362
720×576	21.195	18.590
1226×370	23.441	18.247
1280×720	27.127	38.262
1920×1080	49.023	68.684

1920 × 1080 pixels with the values of FAST_threshold and POINTS_PER_CELL set to 12 and 15, respectively. Moreover, through the experiments, we found that FPGA offers a better power efficiency comparing to GPU implementation.

In the near future, we would like to continue optimizing our design to reduce resource usage and execution time. Then, we intend to integrate in our system a feature matching block which finds correspondences between two consecutive frames. This block is a basic part for estimating relative poses in visual odometry or SLAM applications. Furthermore, we would like to investigate the embeddability of the whole SLAM system on a SoC ARM-FPGA system for mobile applications.

References

1. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras (2016). arXiv preprint arXiv:1610.06475
2. Mei, C., Sibley, G., Cummins, M., Newman, P.M., Reid, I.D.: A constant-time efficient stereo slam system. In: BMVC 2009, pp. 1–11 (2009)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
4. Bay, H., Tuytelaars, T., Van Gool, L., Surf: Speeded up robust features. In: Computer Vision–ECCV 2006. Springer, Berlin, pp. 404–417 (2006)
5. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to sift or surf. In 2011 IEEE International Conference on Computer Vision (ICCV). IEEE, pp. 2564–2571 (2011)
6. Aldegheri, S., Bombieri, N., Bloisi, D.D., Farinelli, A.: Data flow orb-slam for real-time performance on embedded gpu boards. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5370–5375 (2019)
7. Nguyen, D.-D., El Ouardi, A., Aldea, E., Bouaziz, S.: Hoofr: an enhanced bio-inspired feature extractor. In 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, pp. 2977–2982 (2016)
8. Pereira, K., Athanas, P., Lin, H., Feng, W.: Spectral method characterization on FPGA and GPU accelerators. In: 2011 International Conference on Reconfigurable Computing and FPGAs (ReConFig). IEEE, pp. 487–492 (2011)
9. Weber, R., Gothandaraman, A., Hinde, R.J., Peterson, G.D.: Comparing hardware accelerators in scientific applications: a case study. IEEE Trans. Parallel Distrib. Syst. **22**(1), 58–68 (2011)
10. De Schryver, C., Shcherbakov, I., Kienle, F., Wehn, N., Marxen, H., Kostiuk, A., Korn, R.: An energy efficient fpga accelerator for monte carlo option pricing with the Beston model. In: 2011 International Conference on Reconfigurable Computing and FPGAs (ReConFig). IEEE, pp. 468–474 (2011)
11. Pauwels, K., Tomasi, M., Alonso, J.D., Ros, E., Van Hulle, M.M.: A comparison of fpga and gpu for real-time phase-based optical flow, stereo, and local image features. IEEE Trans. Comput **61**(7), 999–1012 (2012)
12. Morales, V.M., Horrein, P.-H., Baghdadi, A., Hochapfel, E., Vaton, S.: Energy-efficient fpga implementation for binomial option pricing using openCL. In: Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014. IEEE, pp. 1–6 (2014)
13. Helali, A., Ameur, H., Górriz, J., Ramírez, J., Maaref, H.: Hardware implementation of real-time pedestrian detection system. Neural Comput. Appl. (2020). https://doi.org/10.1007/s00521-020-04731-y
14. Mami, S., Lahbib, Y., Mami, A.: A new HLS allocation algorithm for efficient DSP utilization in FPGAs. J. Signal Process. Syst. **92**, 153–171 (2019)
15. Intel FPGA SDK for OpenCL Standard Edition: Programming Guide. INTEL (2014)
16. Jelodari, P.T., Kordasiabi, M.P., Sheikhaei, S., Forouzandeh, B.: Fpga implementation of an adaptive window size image impulse noise suppression system. J. Real-Time Image Process. **16**, 2015–2026 (2017)
17. Marin, Y., Mitéran, J., Dubois, J., Heyrman, B., Ginhac, D.: An FPGA-based design for real-time super resolution reconstruction. In: Proceedings of the 12th International Conference on Distributed Smart Cameras, pp. 1–2 (2018)
18. Yao, L., Feng, H., Zhu, Y., Jiang, Z., Zhao, D., Feng, W.: An architecture of optimised sift feature detection for an FPGA implementation of an image matcher. In: International Conference on Field-Programmable Technology, 2009. FPT 2009. IEEE, pp. 30–37 (2009)
19. Bouris, D., Nikitakis, A., Papaefstathiou, I.: Fast and efficient FPGA-based feature detection employing the surf algorithm. In: 2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). IEEE, pp. 3–10 (2010)
20. Chiu, L.-C., Chang, T.-S., Chen, J.-Y., Chang, N.Y.-C.: Fast sift design for real-time visual feature extraction. IEEE Trans. Image Process. **22**(8), 3158–3167 (2013)
21. Lee, K.: A design of an optimized orb accelerator for real-time feature detection. Int. J. Control Autom. **7**(3), 213–218 (2014)
22. Weberruss, J., Kleeman, L., Boland, D., Drummond, T.: FPGA acceleration of multilevel orb feature extraction for computer vision. In: 2017 27th International Conference on Field Programmable Logic and Applications (FPL). IEEE, pp 1–8 (2017)
23. Sun, R., Liu, P., Wang, J., Accetti, C., Naqvi, A.A.: A 42fps full-hd ORB feature extraction accelerator with reduced memory overhead. In: 2017 International Conference on Field Programmable Technology (ICFPT). IEEE, 2017, pp. 183–190 (2017)
24. Pu, Y., Peng, J., Huang, L., Chen, J.: An efficient knn algorithm implemented on FPGA based heterogeneous computing system using opencl. In: 2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). IEEE, pp. 167–170 (2015)
25. Muslim, F.B., Ma, L., Roozmeh, M., Lavagno, L.: Efficient fpga implementation of opencl high-performance computing applications via high-level synthesis. IEEE Access **5**, 2747–2762 (2017)
26. Luo, L., Wu, Y., Qiao, F., Yang, Y., Wei, Q., Zhou, X., Fan, Y., Xu, S., Liu, X., Yang, H.: Design of FPGA-based accelerator for convolutional neural network under heterogeneous computing framework with openCL. Int. J. Reconfig. Comput. (2018). https://doi.org/10.1155/2018/1785892
27. Zhang, S., Wu, Y., Men, C., He, H., Liang, K.: Research on opencl optimization for fpga deep learning application. PloS ONE (2019). https://doi.org/10.1371/journal.pone.0222984
28. Pire, T., Fischer, T., Civera, J., De Cristóforis, P., Berlles, J.J.: Stereo parallel tracking and mapping for robot localization. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) IEEE, pp. 1373–1378 (2015)
29. Konolige, K., Agrawal, M.: Frameslam: from bundle adjustment to real-time visual mapping. IEEE Trans. Robot. **24**(5), 1066–1077 (2008)
30. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Computer Vision—ECCV 2006. Springer, Berlin, pp. 430–443 (2006)

31. Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: fast retina keypoint. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2012, pp. 510–517 (2012)
32. Tola, E., Lepetit, V., Fua, P.: Daisy: an efficient dense descriptor applied to wide-baseline stereo. IEEE Trans. Pattern Anal. Mach. Intell. **32**(5), 815–830 (2010)
33. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
34. Bianconi, F., Fernández, A.: Evaluation of the effects of gabor filter parameters on texture classification. Pattern Recognit. **40**(12), 3325–3335 (2007)
35. Fularz, M., Kraft, M., Schmidt, A., Kasinski, A.: A high-performance fpga-based image feature detector and matcher based on the fast and brief algorithms. Int. J. Adv. Robot. Syst. **12**(10), 141 (2015)

Dai Duong Nguyen received the Electrical Engineering degree in 2014 (mention in Industrial Information) from Hanoi University of Science and Technology, Hanoi, Vietnam and M.S. degree in Information, System and Technology from Paris-Sud University, Orsay, in 2015. He received the PhD. degree with MOSS group in SATIE laboratory, Paris-Sud University in 2018. His research activities are focused on vision systems for SLAM applications.

Abdelhafid El Ouardi received the M.S. degrees from Pierre & Marie Curie University in 2001 and the Ph.D. degree in Electronics from Paris-Sud University in 2005. He worked at Henri Poincaré University, Nancy, as a researcher in 2005–2006. He is currently an Associate Prof at Paris-Sud University, France. In Embedded Systems team of SATIE lab, his research interests include Hardware-Software co-design, evaluation and instrumentation of embedded systems, design of smart architectures for image and signal processing, SLAM applications.

Sergio Rodriguez received his M.S. and Ph.D. degrees from University of Technology of Compiègne, France, in 2007 and 2009, respectively. Since 2011, he is an Associate Professor of the Paris-Sud University. His research activities are focused on dynamic scene analysis through multi-modal perception intended to enhance Intelligent Transportation Systems applications.

Samir Bouaziz received the Ph.D. degree in Electronics, in Paris-Sud University, Orsay, France 1992. He is a full Professor at Paris-Sud University. His research focuses on Hardware-software designs of embedded systems for autonomous vehicles and robots. His research is led by time constraints and complexity consideration for a good fit between hardware and algorithms, instrumentation and benchmark systems to understand Human–Vehicle interactions.

sensors

MDPI

Article

Integrity Monitoring of Multimodal Perception System for Vehicle Localization

Arjun Balakrishnan, Sergio Rodriguez Florez * and Roger Reynaud

CNRS, ENS Paris-Saclay, Université Paris-Saclay, 91190 Gif-sur-Yvette, France;
arjun.balakrishnan@universite-paris-saclay.fr (A.B.); roger.reynaud@universite-paris-saclay.fr (R.R.)
* Correspondence: sergio.rodriguez@universite-paris-saclay.fr

Received: 22 July 2020; Accepted: 15 August 2020; Published: 18 August 2020

Abstract: Autonomous driving systems tightly rely on the quality of the data from sensors for tasks such as localization and navigation. In this work, we present an integrity monitoring framework that can assess the quality of multimodal data from exteroceptive sensors. The proposed multisource coherence-based integrity assessment framework is capable of handling highway as well as complex semi-urban and urban scenarios. To achieve such generalization and scalability, we employ a semantic-grid data representation, which can efficiently represent the surroundings of the vehicle. The proposed method is used to evaluate the integrity of sources in several scenarios, and the integrity markers generated are used for identifying and quantifying unreliable data. A particular focus is given to real-world complex scenarios obtained from publicly available datasets where integrity localization requirements are of high importance. Those scenarios are examined to evaluate the performance of the framework and to provide proof-of-concept. We also establish the importance of the proposed integrity assessment framework in context-based localization applications for autonomous vehicles. The proposed method applies the integrity assessment concepts in the field of aviation to ground vehicles and provides the Protection Level markers (Horizontal, Lateral, Longitudinal) for perception systems used for vehicle localization.

Keywords: multimodal data source; integrity assessment; intelligent vehicles; localization; Protection Level markers

1. Introduction

The second half of the last decade has seen a significant emergence of commercially available vehicles with autonomous driving capabilities. We can confidently say that the status of autonomy in vehicles is well into the realm of Society of Automotive Engineers (SAE) level 2 [1]. While the researchers and industries are rapidly moving towards SAE level 3 systems that can dramatically improve driving safety and efficiency, monitoring the integrity of sources and process used in such systems can often pose challenges [2]. In [3], the classical integrity concepts used in aviation are transposed to integrity requirements for ground vehicle localization. Using road-safety-related statistics and geometry of roads and vehicles, [3] derived bounds for localization error in both highway and urban scenarios. They further distributed the derived total integrity risk to allocate integrity levels to every subsystem present in autonomous vehicles. In this work, we focus on the integrity assessment of perception data sources such as vision, LiDAR, map, etc. Most advances in this area explicitly address the task of integrity monitoring of data sources by introducing redundancy in sensors [4,5], using sensors with advanced features [2,6], monitoring repetitive journeys [7], or assuming one source (often high-quality digital maps) as reliable ground truth [8,9]. While adding data redundancy (often different GPS receivers for map-matching and sensor fusion [5]) can monitor the integrity of processes, the integrity of data sources has to be largely assumed. Only a small number of works like [10] and [7] consider digital maps as a source with probabilities of error. However, to achieve context-aware

autonomous navigation, perception sensors such as camera and LiDAR are used along with digital maps, GPS, and proprioceptive sensors. In [11], facades of buildings at intersections are detected using vision and are fused with building footprints extracted from the digital map to provide better localization. They further extended their work in [12] to achieve localization at intersections using road structures instead of building facades and map data. A map-matching-based localization involving lane detection from vision is used in [13] and [14]. Similar strategies are employed combining digital maps with features detected from LiDAR data such as curb detection [15], intersection structure detection [16], lane detection [17], etc. However, to the best of our knowledge, integrity monitoring of data from such spatial perception sensors used in the aforementioned works is largely overlooked. Considering the multimodality of data provided by this wide variety of sensors, finding a common framework to evaluate integrity is a challenging yet crucial task. In [18], we made an effort to address this task using a cross-consistency-based integrity monitoring framework for highway scenarios. In this paper, we address limitations of [18] and improve the framework to apply it to complex semi-urban and urban scenarios in a generalized way, thus providing context awareness to a multimodal vehicle localization system.

2. Problem Statement

Semi-urban and urban environments often contain a multitude of intersections, roundabouts, road-splits, and merges compared to highway scenarios. As discussed in Section 1, multimodal data from different sources are used to achieve accurate localization in such scenarios. Developing upon the framework presented in [18], finding a generalized common model for the representation of data from all sources is the primary objective of this work. Even though works like [12] and [17] propose geometrical models for several types of intersections, they are limited to a single perception data source and digital maps. They also require prior classification of intersections to reliably fit the predefined models to the data. On the other hand, sensors used in intelligent vehicles have considerably different behavior and output in such scenarios. Hence, the rest of this section is focused on how data from different sources are used in complex scenarios. We also examine the possible errors associated with these use cases and discuss the applicability issues of a simple common geometrical model (e.g., the polynomial model in [18]) in these situations.

Traditionally, vision data is used to detect ego lane markings and/or lanes parallel to the ego lane using a curvature-based model. In urban scenarios, such lane detection models fail due to different types of lane markings (e.g., stop lines, road separation markings, etc.), orientation (e.g., lane markings from other road sections in the junctions) and complex curvatures (e.g., splitting and merging lane markings). Another approach using visual data is to detect the drivable road region in front of vehicle. But due to the unforeseeable shapes of possible road segment detections, modeling of such output with a geometrical model is difficult. Intersections with multilane branch roads can have a large common region at the center, which can limit the observability of other road branches through visual inputs.

It is reasonable to assume that vehicles travel slowly and stop more often in semi-urban and urban scenarios than highways. GPS receivers are proven to have poor performance in slow-moving vehicles [19]. Combined with the fact that the presence of buildings and other obstructions can cause multi-path effects or even outages of signals [2], GPS receivers experience classical localization problems in urban environments.

With the exception of a few advanced and proprietary Geographic Information Systems (GISs, e.g., Google maps), publicly available GIS sources lack accurate road properties (lane or road widths, locations of lane splits and merges at junctions, etc.) and strongly depend on rule-based rendering to display maps. The discrepancies observed while overlapping the satellite view and rendered map structures from different GISs as shown in Figure 1 are examples of the limitation of this approach. GPS tracking of the vehicle is accurate in satellite view of the junction, which includes a lane change to the leftmost lane of the highway for a left turn and a smooth turn through the left side of link road. However, from the rendered road structure view of all the map sources, the track

section corresponding to lane change appears to be wrong as it is outside the boundary of the road structure. It is also worth noticing that none of the GIS represent roads with their actual width, but with rule-based dimensions. It is evident from the same width of two highway sections despite different number of lanes in each of them. Likewise, modeling of junctions is also considerably different in each map source, particularly between Google Maps and OpenStreetMap. Hence, inclusion of map data in localization process is suboptimal in urban and semi-urban scenarios and forces us to consider it as a data source with associated instantaneous integrity rather than ground truth.

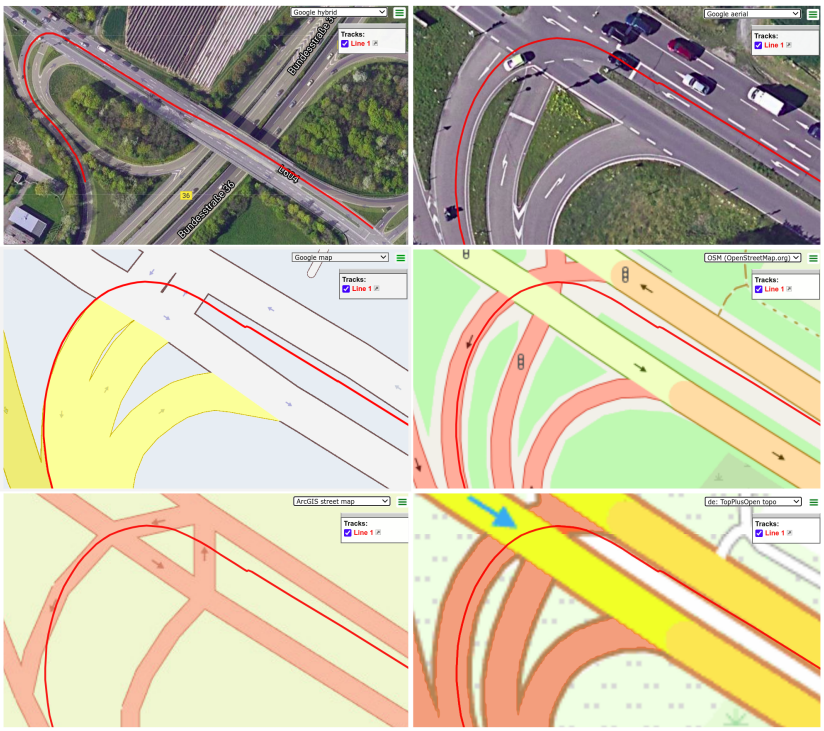


Figure 1. Integrity issues in map sources. Top-left: An example of a GPS track (in red) from the KITTI dataset projected on a satellite map from Google. Top-right: Zoomed aerial view of the track at an intersection. Middle-left: The intersection in a street map from Google. Middle-right: The intersection in a street map from OpenStreetMap. Bottom-left: The intersection in a street map from ArcGIS. Bottom-right: The intersection in a street map from the Federal Agency for Cartography and Geodesy (BKG) of Germany.

While data from vision, GPS, and maps add complexities and impose limitations, LiDAR, on the other hand, can provide useful data in urban and semi-urban environments. It can observe the ego road and other road branches efficiently. By using the reflectivity information available in LiDAR data, we can detect bright surfaces like lane markings and curbs [15]. Though LiDAR poses challenges in the detection and modeling of features as in the case of vision, the accurate 3D information available makes it an important source for representing the structure of a large urban scenario.

The integrity monitoring method in [18] provides a weighting scheme for data sources that infers the cause of inconsistencies observed in the data-fusion method at a given time. For any data source combination that can be represented in a common frame and with a common model in that chosen frame, the cross-consistency analysis proposed in [18] can be applied. However, the discussion presented in this section shows that developing a common model is difficult when different sensor modalities and diverse features are introduced to the system in order to accommodate urban scenarios. To this extent, we could not find any integrity assessment solution in the literature that can handle more than two perception data sources and a wide variety of scenarios.

2.1. Contributions

The paper presents the following contributions based on the problem statement outlined above.

1. Defining a common reference frame and formalizing a common model to represent all data sources in all scenarios.
2. Prototyping an integrity assessment framework using the common model and providing proof of concept.
3. Analyzing the performance of the proposed framework using publicly available datasets and comparison with other state-of-the-art integrity monitoring solutions from the literature.

3. Methodology

The framework proposed for the integrity assessment developed in this work is given in Figure 2. The Detection Block includes sensor-specific routines to detect features that are relevant to different data fusion algorithms described in Section 2. The Rendering Block uses GPS position to extract data from surrounding map regions and applies rule-based rendering to reconstruct the geometrical structure of the area. The obtained information is represented in a common frame using a common model. In this work, the common reference frame is chosen as the ego frame of the vehicle as the transformations between ego frame, camera frame, LiDAR frame, and GPS frame can be determined by calibration procedures [20]. A decision algorithm is used to decide whether the optimization of localization is required in case of unknown transformations between frames of data and the common reference frame (in our case, map frame to ego frame). Once the required optimization is achieved, coherence between data representations is evaluated and integrity is assessed for each source. In this section, we outline the specific techniques and concepts used in the framework presented in Figure 2.

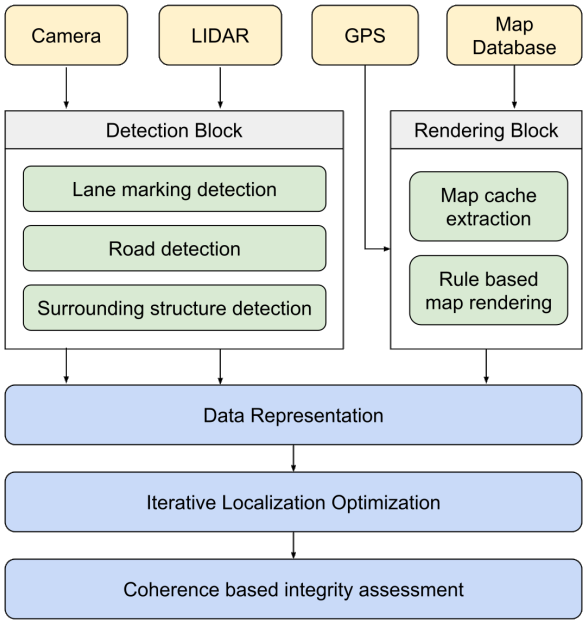


Figure 2. Framework for integrity assessment of multimodal data sources.

3.1. Detection

The purpose of the Detection Block is to extract the same information (features) from each data source. From the literature review, we identify three features that are most commonly used in state-of-the-art localization methods in urban scenarios—lane markings, drivable roads, and the structure of the surroundings of the vehicle. Here, we limit the surrounding structures to grass patches/vegetation and curbs and avoid building facades and other objects due to the complexities

of their detection. Indeed, any feature can be used in this process if it is detectable from every data source considered. The methods used to detect these features from each source are explained here.

3.1.1. Vision

To accommodate varieties of lane markings present in different scenarios, all possible markings are detected. Images from cameras are transformed to bird's-eye view (BEV) using camera calibration. Intensity-based segmentation is used to detect all possible white lane markings. After detection of all the candidate lane markings, blob analysis is used to reject poor detections [14]. Seed-based wavefront segmentation is used to detect dark road regions with asphalt and regions with grass patches. For road segmentation, seeds are selected in front of the vehicle and using propagating waves from these seeds, connected road regions are segmented. Seeds for grass patch detection are selected by color-based keypoint detectors. After these detections, every pixel in the BEV can be classified into lane markings, roads, other surfaces, or unclassified.

3.1.2. LiDAR

A subset of LiDAR data containing points that lie inside a 3D region of interest (ROI) is selected. Points on the road and on the edges of the road are classified using 3D gradients. The ROI is divided into smaller patches in the XY plane, and the points belonging to each patch are examined for their Z values. This helps to differentiate between road segments, curbs, dividers, vegetation, etc. using the technique presented in [21]. Points with high reflectivity are also selected as they correspond to the bright surfaces such as lane markings and railings. These are further classified into reliable lane marking detections by combining their position with road regions. As a result of these detection steps, every point in the ROI is classified with lane markings, roads, other surfaces, or unclassified.

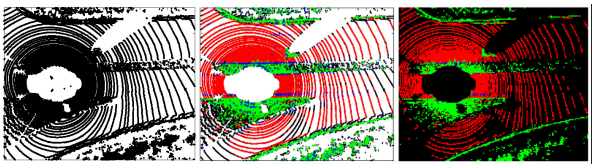
3.2. Map Handling

OpenStreetMap (OSM) is used in this work as a GIS source. OSM provides nodes corresponding to ways, grass patches, and railings, etc. However, finding relevant geometrical information in a vehicle's surroundings from maps involves two key components: location and orientation of the vehicle [10]. Using location measurements, all of the relevant map nodes in the ROI are selected and the map data is transformed into the vehicle's ego-frame using the orientation of the vehicle. The location estimate is provided by the GPS sensor, whereas the orientation estimate is given by the on-board Inertial Measurement Unit (IMU). Once the map nodes are represented in ego frame, a rule-based rendering algorithm is used to create a geometrical sub-map for the ROI. The number of lanes, lane width (when available), location of road boundaries, boundaries of curbs, dividers, and vegetation, etc. are used in the rendering process, producing an enriched geometrical model of the environment from OSM. In works like [13,14], custom-made high-definition maps (HD maps) that contain lane marking information and accurate road structure information are used. Even though the exact location or type of lane markings are unavailable in OSM, assuming continuous lane markings on the left side of the leftmost lane, right side of the rightmost lane, and dashed lane markings for the lanes in the middle, approximate lane level information can be produced. In case of missing lane width information, the standardized road construction guidelines of the country are used to render the map. However, it is evident that errors in GPS positioning or orientation estimation can greatly affect the accuracy of map data extraction and cause uncertainties in map rendering [10], especially for the exact locations of lane markings.

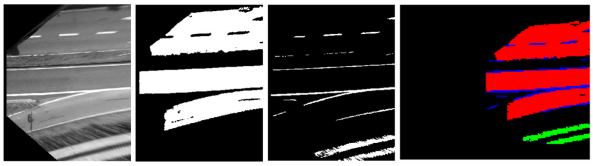
3.3. Representation

To be able to deal with the features and geometries of different types and shapes, a 2D feature grid (FG) is proposed as the model. FG consists of an array of cells where each cell c_i represents a 20×20×100 cm block in the real world. Four feature labels (LBs) are assigned to cells in the FG according to the type of feature: (1) LB_r—road, (2) LB_l—lane marking, (3) LB_o—other surfaces,

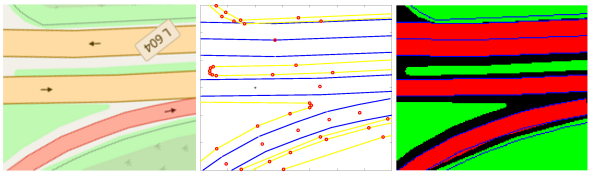
(4) LB_u—unclassified/unidentifiable. The blocks corresponding to each of the cells are examined for the information they contain. The type of feature with the highest ratio inside a block is used to assign the respective label to the cell. Each data source produces an FG following this criterion, as shown in Figure 3.



(a) Left: LiDAR data in ROI. Middle: Detection of road (red), lane markings (blue), other surfaces (green), and unclassified points (black). Right: Feature Grid from LiDAR



(b) From Left, 1—bird's-eye view (BEV) image. 2—Output of road detection. 3—Output of lane detection. 4—Feature grid from vision



(c) Left—Rendered OpenStreetMap (OSM) map from official website. Middle—Available map data in ROI. Right—Feature Grid from map

Figure 3. Example of modeling data from different sources using feature grid representation: cells with road labels (red), cells with lane marking labels (blue), cells with other surface labels (green), cells with unclassified labels (black).

Along with labels, it is important to model the confidence of data provided by each sensor. The accuracy of LiDAR data decreases as the distance from the sensor to the measurement location increases [22]. On the other hand, the Inverse Perspective Mapping (IPM) transformation used to create the bird's-eye view images from actual images increasingly introduces deformation as the distance from the camera increases due to camera calibration errors. To account for these facts, a confidence function is proposed, drawing inspiration from [18] for all relevant FGs. Using the concept of an Inverse Distance Weighting (IDW) function presented in [23], the weights are computed as

$$w_{ij} = 1 - \left\langle \sqrt{(x_{ij}^2 + y_{ij}^2 + h_s^2)} \right\rangle_{min-max}, \quad (1)$$

where w_{ij} is the weight associated with the cell C_{ij}, x_{ij} and y_{ij} are the distances to the center of C_{ij} from the sensor position, and h_s is the height of the sensor. The min–max normalization operator $\langle \hat{x} \rangle_{min-max}$ is defined as

$$\hat{x}_{norm} = \frac{\hat{x} - \hat{x}_{min}}{\hat{x}_{max} - \hat{x}_{min}}.$$

Hence, the total representation of data from sources will have two components: the labels and their importance, which is denoted by the weights. Other source-specific weighting functions using homography of image transformations and LiDAR data acquisition models can also be used for this purpose. However, data sources like maps use uniform weights for all the cells in their FGs due to the fact that they are not measured but just extracted.

3.4. Integrity Analysis

The treatment of different sensors as multimodal data sources with a common frame of representation and the same dimensionality allows us to use the definitions of integrity presented in

the domain of data sciences. Integrity measures overall accuracy and consistency of data sources [24]. While accuracy is defined as the correctness of validated data, consistency refers to the measure of coherence between them. Data sources with high consistency can be treated as reliable, and their integrity can be expressed as a function of coherence with respect to other data sources.

Let $S = \{s_1, s_2, s_3, \cdots, s_N\}$ be the set of N sensors and s_iFG be the feature grid provided by each sensor. One cell c_k with feature label LB_x from s_iFG is defined as consistent if there is at least one matching cell with LB_x in a 3×3 neighborhood around the cell c_k in s_jFG. By extension, a matching operation f_m between FGs is defined as

$$f_m(s_iFG, s_jFG) = N_m^{s_iFG} / N_T^{s_iFG}, \tag{2}$$

where $N_m^{s_iFG}$is the number of matching cells in s_iFG and $N_T^{s_iFG}$ is the total number of applicable cells in s_iFG, i.e., cells with labels except LB_u. After computing the matches between all of the possible combinations, the integrity associated with a source is computed as

$$W_i = \frac{\sum\limits_{\forall j, i \neq j} f_m(s_iFG, s_jFG)}{\sum\limits_{\forall i, j\, i \neq j} f_m(s_iFG, s_jFG)}. \tag{3}$$

3.5. Localization Optimization

The integrity analysis mentioned in Section 3.4 assumes that the localization of a vehicle is accurately known, i.e., the localization information used in map extraction is reliable. But in real-world applications, GPS positioning—even from an inertial/dead reckoning coupled GPS receiver—can have errors due to multipath effects, outages, or drifts. Inherently, error in localization affects the consistency of map data to the other sources, hence impacting the integrity of the whole system. Hence, we developed a localization optimization procedure that uses semantic-level information from data representations of sources. It can efficiently allow integrity assessment and also identify particular defaults such as map offsets or inconsistent map sections.

In this work, a particle filter [25] is developed for map-matching to improve localization. The block diagram for the localization optimization in the ego frame of the vehicle with decision criteria is given in Algorithm 1. In the first step, new position and orientation measurements from GPS and IMU are compared with the current best localization estimate. If the new measurements ($x_m : [x_m, y_m, \theta_m]$) are not within the non-holonomic constraints of the current state ($X_{state} : [x_{state}, y_{state}, \theta_{state}]$) of the vehicle, they are detected as an outlier [26]. Conversely, consistent position and orientation measurements are used to render a map from the database and the coherence between FGs of the map and other sources is computed. If sufficient coherence is observed (greater matching than the empirically-derived threshold for $f_m(s_iFG, s_jFG)$ considering different sensors and scenarios), localization optimization is not performed and the data representations from each source are used for integrity assessment. In case of poor coherence between the combinations, a sequential localization optimization using particle filters is performed. The transformation function t on map FG (MFG) used to maximize the coherence between sources is defined as

$$t(MFG, x, y, \theta) = R(\theta) * s_iFG + T(x, y), \tag{4}$$

where $R(\theta)$ is the 2D rotation matrix constructed using θ and T is the 2D translation vector constructed using x and y translations.

In the sequential localization optimization, coherence between the map (MFG) and each of the other sources (s_iFG) is maximized in ego frame along the y direction (lateral) at first by iteratively distributing particles around the best match localizations. The lateral offset estimation y^* and the final distribution of particles from this step is used for initializing the second particle filter, which maximizes the match along the x (longitudinal) and θ (heading) dimensions. The resulting optimized localization

$(x^*, y^*, \theta^*)_{s_iFG}$ for each s_iFG is checked for consistency by thresholding the distance between them. If they are not consistent, the coherence between all s_iFG is computed. An issue with the map structure is identified if the coherence between other sources (other s_iFG combinations, e.g., LiDAR–vision) is good but] the localization optimization of these sources cannot produce consistent localizations (within 2σ uncertainty bounds). If the localization estimations for each sensor combination are consistent, the estimation that gives the best coherence is chosen and integrity assessment is carried out. This estimation is also used to update the current localization estimation for the next time step.

Algorithm 1 Algorithm for localization optimization

Inputs: Localization: X_{state}, GPS+IMU localization measurement: x_m, FG of LiDAR: L, FG of Vision: C, FG of Map: M, Minimum coherence limit: $limit$
if X_{state} and x_m are consistent **then**
  if $f_m(L, M) > limit$ and $f_m(C, M) > limit$ **then**
    Output: Integrity markers
    Update X_{state}
  else
    Compute:
    $y_L^* = arg\,max\,(f_m(L, t(M, 0, y, 0)))$
    $y_C^* = \overset{y}{arg\,max}\,(f_m(C, t(M, 0, y, 0)))$
    $(x_L^*, \theta_L^*) \overset{y}{=} arg\,max\,(f_m(L, t(M, x, y_L^*, \theta)))$
    $(x_C^*, \theta_C^*) = \overset{x,\theta}{arg\,max}\,(f_m(C, t(M, x, y_C^*, \theta)))$
    if $(x, y, \theta)_L^*$ and $\overset{x,\theta}{(x, y, \theta)_C^*}$ are consistent **then**
      Output Integrity values
      Update X_{state}
    else
      if $f_m(L, C) > limit$ **then**
        Output: Integrity markers
      else
        Output: Error in map
      end if
    end if
  end if
else
  Output: Error in GPS
end if

4. Experiments and Discussions

Experiments are conducted with scenarios available in the KITTI benchmark suite [27] to establish proof of concept. Real-Time Kinematic (RTK) GPS fixes in these datasets are added with noise generated using the GPS-noise simulation model proposed by [28] to simulate poor GPS localization fixes. Outliers that are higher than the 2σ variance of the GPS-noise simulation model are used to replace RTK GPS fixes at random sections of the trajectory. Finally, 5% of the RTK GPS fixes are randomly removed from the trajectory to emulate GPS outages as they may occur in generic GPS receivers. Since different data sources have different spatial ranges, a 3D region of interest (ROI) in the vehicle's ego frame is established. its limits in XY plane are chosen as 25 m in front of the vehicle (positive X axis), 15 meters behind (negative X axis), and 15 m at each side (Y axis). Since vision cannot provide data in the back of the vehicle as well as to the front bumper of the vehicle, the ROI of vision is limited from 3.5 m to 25 m along the positive X axis.

Even though the vision data used in this work does not cover the back view of the vehicle, the other two major sources—LiDAR and map—can provide information in the back of vehicle, hence justifying the choice of the limit in negative X axis.

The discussion on the results has three parts. Firstly, comparing the performance of the proposed method to the method in [18]. This includes a comparison of integrity markers in the datasets presented in [18] and showcasing the improvements provided by the new method in handling fault and feasibility predictors (FPs) produced by the previous method. FPs are the markers generated when the fitting of the common model to the data sources is not possible or feasible. These markers suggest the limitations of the method in [18], which mainly arise when the method is applied on non-highway scenarios. The set of five FP markers defined is

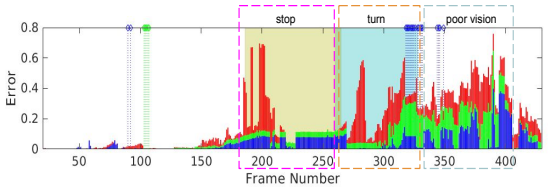
- FP_m: Not enough nodes in the map for model fitting;
- FP_v: Not enough lane markings for model fitting;
- FP_g: GPS measurement is not available or an outlier;
- FP_s: Vehicle not moving or moving very slowly;
- FP_t: Vehicle performing a hard turn.

The second part of this discussion considers more datasets in semi-urban and urban scenarios to evaluate the integrity estimation of sources in complex situations such as junctions, road splits, and merges, etc. In the final part, we compute classical integrity markers from our framework and compare them with values presented in [3].

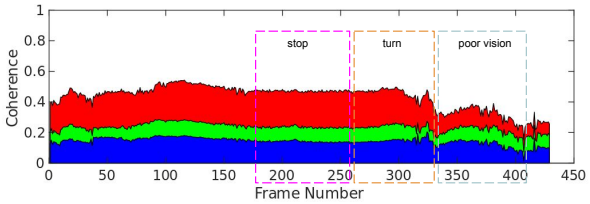
4.1. Integrity Marker Comparison

In this section, we compare the results presented in [18] with the results obtained from the new method. The key difference between these two methods is the parameter they use for the integrity computation. The former uses the error observed in model fitting to evaluate integrity, whereas the latter uses coherence between data representations to achieve the same. Hence, the contribution of error by each sensor and the contribution of coherence by each sensor are used for this analysis of the results of these methods, respectively. The same errors are introduced in the GPS for each algorithm, and the results obtained from the dataset 2011_09_26_drive_0029 are shown in Figure 4.

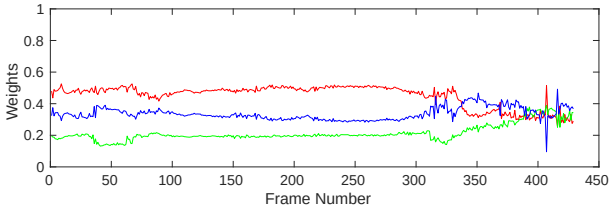
The primary advantage of the proposed method is the ability to evaluate the integrity in conditions where FPs are produced due to the limitations of the model-based integrity analysis employed in [18]. The stopping of the vehicle between frame numbers 187 and 265 and a hard left turn at the junction from 265 to 330 cause poor model extraction using the previous method, resulting in an unusable integrity evaluation. Consistent coherence is observed during the same scenario as shown in Figure 4b using the new method, providing meaningful integrity estimation. Figure 5a shows an example frame (207) in this section, where polynomial model estimation fails to represent data from sources. On the other hand, the FGs are able to represent the scenario well. After frame 330, the vehicle enters a curved link road with challenging light conditions such as shadows and oversaturated road sections, as shown in Figure 5b, causing large model-fitting errors in vision, shown in Figure 4a. Though a decrease in the coherence is observed, the addition of LiDAR and introduction of new features helps the new method provide more consistent integrity markers.



(a) Error from each sensor used in dataset 2011_09_26_drive_0029 using the previous method. Red: GPS error; Green: Map error; Blue: Vision error; Green dotted lines: FP_m; Blue dotted lines: FP_v; Light brown: FP_s; Light blue: FP_t

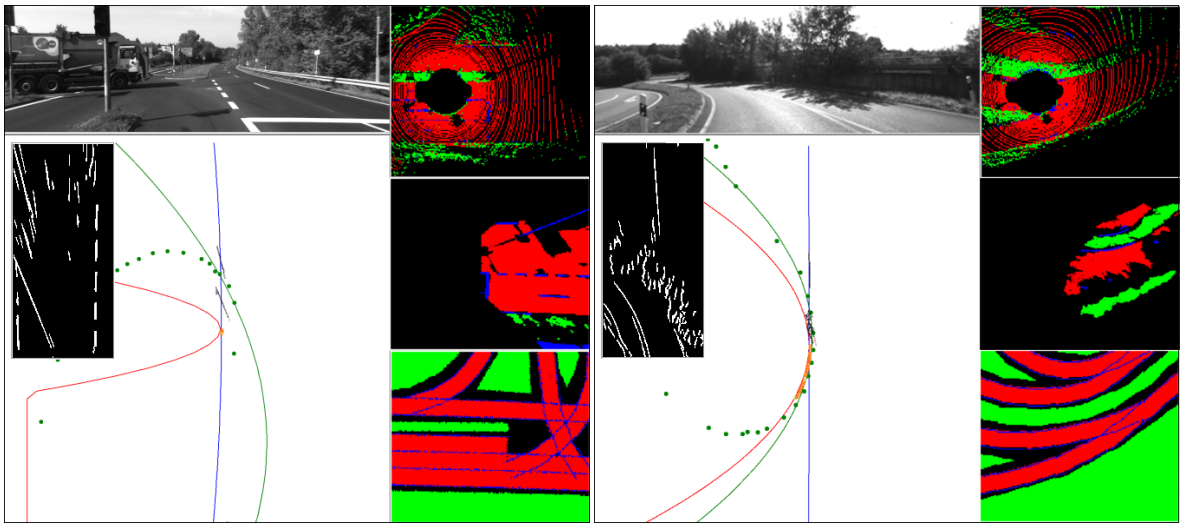


(b) Coherence observed for each sensor used in dataset 2011_09_26_drive_0029 using new method. Red: LiDAR; Green: Map; Blue: Vision



(c) Integrity Markers for dataset 2011_09_26_drive_0029 using the new method. Red: LiDAR; Green: Map; Blue: Vision

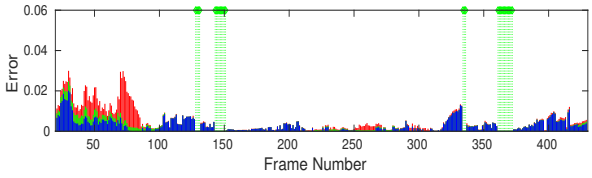
Figure 4. Comparison results of dataset 2011_09_26_drive _0029.



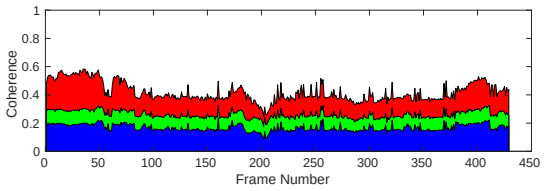
(a) Scenario at frame no. 207 **(b)** Scenario at frame no. 375

Figure 5. Specific scenarios from dataset 2011_09_26_drive_ 0029. Top left: view of the scenario; bottom left: model fitting; left inset: lane-marking detections; top-right: feature grid (FG) of LiDAR; middle right: FG of vision; bottom right: FG of map.

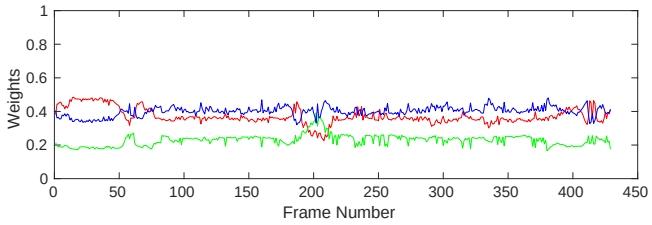
In Figure 6, the results of integrity assessment in a highway scenario are presented, where the old method reliably performed. The FP_m instances observed in this dataset are due to the lack of map nodes to reliably fit the polynomial model in straight line road sections. In the new method, the model fitting is replaced with FG data representation, which eliminates such errors in modeling.



(a) Error from each sensor used in dataset 2011_09_26_drive_0028 using the previous method. Red: GPS error; Green: Map error; Blue: Vision error; Green dotted lines: FP_m; Blue dotted lines: FP_v; Light brown: FP_s; Light blue: FP_t



(b) Coherence observed for each sensor used in dataset 2011_09_26_drive_0028 using the new method. Red: LiDAR; Green: Map; Blue: Vision



(c) Integrity Markers for dataset 2011_09_26 drive_0028 using the new method. Red: LiDAR; Green: Map; Blue: Vision

Figure 6. Comparison results of dataset 2011_09_26_drive _0028.

The comparison of integrity markers in specific cases presented in [18] with the integrity markers provided by the new method is given in Table 1. A general tendency of improved integrity values is observed across all datasets and scenarios. For example, in the second row of Table 1, the integrity weight of vision computed using the old method was lower due to the improper detection of curved lane markings as straight lane markings. This resulted in an inconsistent polynomial model compared to the other two data sources, causing a low integrity weight of 0.175. But using the new method, drivable road detection along with surrounding structure detection improved the consistency of vision data with other sources, resulting in a higher integrity value of 0.612. The proposed method is proven to be able to handle every situation where FP was provided by the old method. In the first row of Table 1, the lack of sufficient map nodes on a straight road segment made model-based integrity estimation impossible, as confirmed by the FP_m flag. The new approach enables integrity estimation and provides an integrity weight of 0.422. It is worth noting that a high integrity value is not observed because of poor map rendering due to lack of correct lane width information from the map. Incorrect road segment selection from the map does not affect the new method as it uses all of the neighborhood roads in integrity estimation.

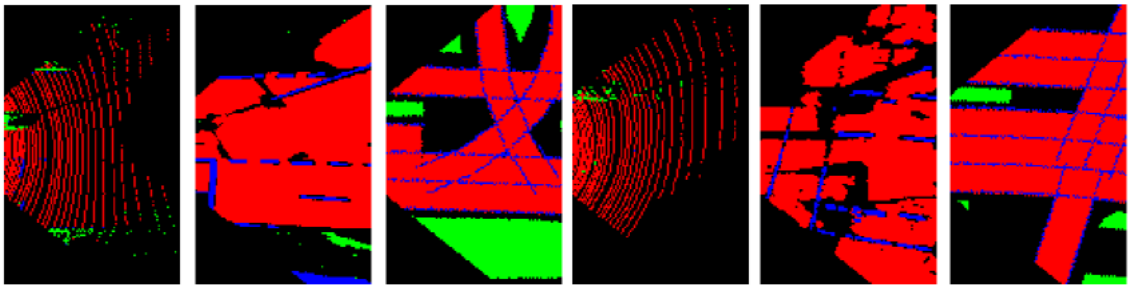
Table 1. Results obtained using the proposed method and the method presented in [18].

Dataset–Frames	Integrity [18]	Integrity (Ours)	Situation
Dataset 1–150	FP_m	map-0.422	not enough nodes from the map
Dataset 1–21	vision-0.175	vision-0.612	no good quality lane markings
Dataset 2–390	map-0.087	map-0.374	road with multiple curvatures
Dataset 3–562	FP_v	vision-0.573	partial occulusion in vision due to vehicles
Dataset 3–1117	map-0.006	map-0.381	wrong map extraction
Dataset 4-22	vision-0.214	vision-0.681	road with multiple curvatures
Dataset 4-260	vision-0.651	vision-0.629	highway road with single curvature

4.2. Complex Situations

This section is dedicated to analyzing the behavior of the integrity assessment system in some of the selected complex scenarios present in the KITTI dataset. In Figure 7a, an example of a semi-urban road junction is shown. Due to the lack of information from the map, the rendering process failed to reconstruct the continuity of lanes at the intersections. On the other hand, vision and LiDAR data detected all of the branch roads at the junction and managed to perceive the width of each of these road sections accurately. This results in a lower integrity value for the map at this junction (Frame numbers: 310–320) compared to other sources, as shown in Figure 4c.

One of the main reasons behind the proposed data representation is the fact that it is an improvement over other existing geometrical models for intersections, which fail to accommodate partially correct data. Figure 7b shows a partial road detection from LiDAR due to the difference in elevation of one of the road branches in the scenario. Even though data available from LiDAR is not complete, the part that is detected is coherent with both vision and map. In fact, LiDAR has more integrity than vision in the comparison, not only because of its coherence in road detections, but also, the available grass-patch detection compensates the partial road detection. The integrity values in this scenario (Frame numbers: 120–200, dataset 2011_09_26_drive_0011) are computed around 0.456, 0.349, and 0.165 for LiDAR, vision, and map, respectively.



(a) Multilane junction from dataset 2011_09_26_ drive_0029.

(b) Partially consistent data from a junction from dataset 2011_09_26_drive _0011.

Figure 7. Examples of complex scenarios—cells with road labels(red), lane marking labels (blue), other surface labels (green), unclassified labels(black).

4.3. Performance of Integrity Monitoring

To evaluate and compare the proposed integrity framework to the integrity concepts transposed from civil aviation concepts, the Horizontal Protection Level (HPL) is computed. According to [29], the HPL is the radius of a circle in the horizontal plane that describes the region assured to contain the indicated horizontal position. It is the statistical bound for horizontal position error with a confidence level derived from the integrity risk requirement of an application. We also compute the Lateral Protection Level ($LatPL$) and Longitudinal Protection Level ($LonPL$), as proposed in [3]. The illustration given in Figure 8 shows the geometrical interpretations of these protection levels with respect to the ego frame of the vehicle and feature grids. Extending these concepts, we use the final distribution of the particles from the localization optimization particle filter described in Section 3.5 to compute $LatPL$, $LonPL$, and HPL. The lateral and longitudinal positions of all the particles that belong to the 95th percentile of the coherence matching scores are modeled using a Gaussian distribution. $LatPL$, $LonPL$, and HPL are then computed using the average standard deviation of particle distributions from each sensor combination used to optimize localization as

$$LatPL = 2\sqrt{\left(\sigma_{CY}^2 + \sigma_{LY}^2\right)/2}, \tag{5}$$

$$LonPL = 2\sqrt{\left(\sigma_{CX}^2 + \sigma_{LX}^2\right)/2}, \tag{6}$$

$$HPL = 2\sqrt{\left(\sigma_{CX}^2 + \sigma_{CY}^2 + \sigma_{LX}^2 + \sigma_{LY}^2\right)/4}, \tag{7}$$

where σ_{CX}^2 and σ_{CY}^2 are the lateral and longitudinal variances of particles from the vision-map optimization result and σ_{LX}^2 and σ_{LY}^2 are the lateral and longitudinal variances of particles from the LiDAR-map optimization result.

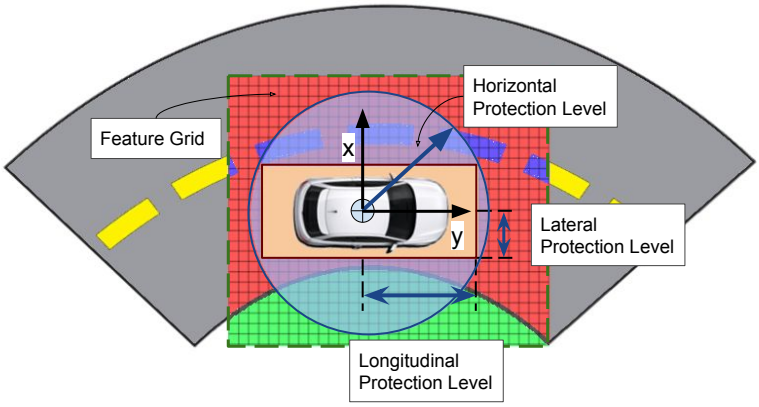
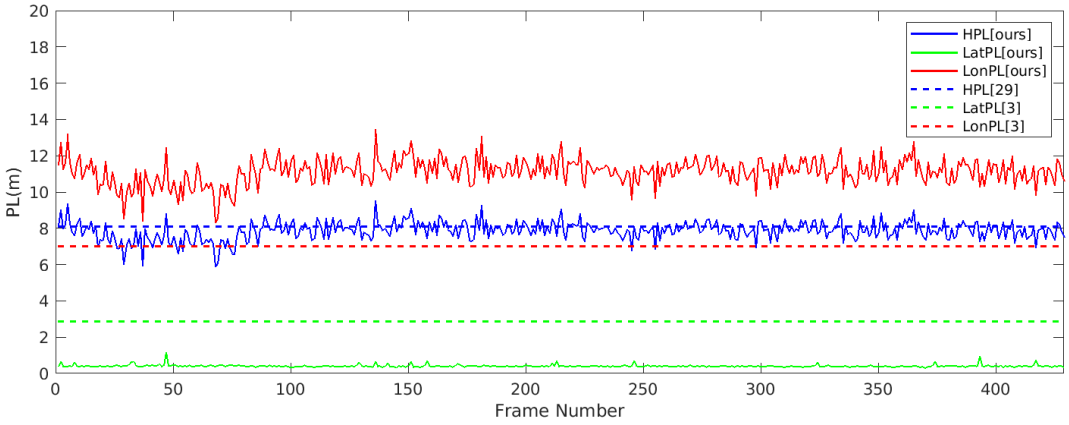
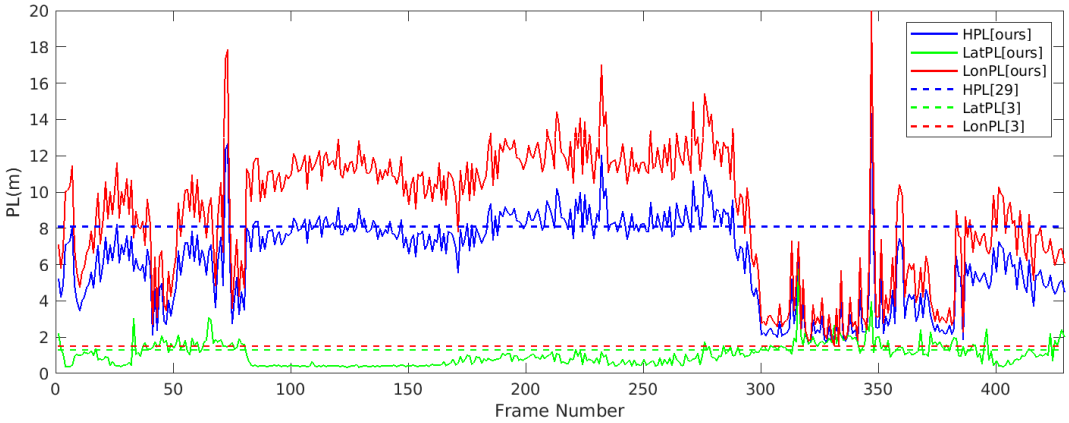


Figure 8. Illustration of protection levels for the localization of ground vehicles.

The results obtained from the HPL evaluation of two of the datasets presented in Section 4.1 are shown in Figure 9. Using historical HPL data available from hte European Global Navigation Satellite System Agency [30], the average value of the HPL over the last 5 years (from 01-2015 to 07-2020) for the nearest zone (Zurich) to the dataset location (Karslruhe) is calculated as 8.1 m. According to the total integrity levels and allocation of integrity risks derived in [3], $[LatPL, LonPL]$ of the perception block is computed as [2.85 m, 7 m] for highway scenarios and [1.45 m, 1.45 m] for non-highway scenarios. The results of this comparison are shown in Figure 9. In highway scenarios, the $LatPL$ computed using our method is completely within the $LatPL$ limit derived by [3], whereas in urban scenarios, 91% of the time, $LatPL$ from our method is under the limit. On the other hand, the HPL computed using our method shows good coherence with the historical HPL calculated using [30]. However, the $LonPL$ computations are, most of the time and in both scenarios, outside the limit of $LonPL$ derived by [3]. This is due to the fact that the sensors considered in this work are better at providing lateral information ([3]) than longitudinal information. This is evident from the highway scenario in Figure 9a, where the road is straight without any other significant information to bound the sensor data in the longitudinal direction. In Figure 9b, sections where the $LonPL$ computed from our method is closer to the $LonPL$ limit of 1.45 m contain curved road sections or other distinguishable surfaces, which helps to reduce $LonPL$ considerably. Hence, the results presented in this section demonstrate the capability of the proposed method to assess the integrity of perception sensors in localizing vehicles with the accuracy required for urban and highway navigation.



(a) Horizontal Protection Level (HPL) evaluation result of dataset 2011_09_26_drive_0028 (highway scenario)



(b) HPL evaluation result of dataset 2011_09_26_drive_0029 (urban scenario)

Figure 9. Horizontal Protection Level (HPL) comparison.

5. Conclusions

This work presents a framework for integrity monitoring of sources used in the localization of autonomous vehicles. The limitations of common geometrical models in representing multimodal data sources are identified in this work. To overcome these issues, a semantic feature grid model is proposed that can geometrically represent different features using labels. A function for coherence evaluation between feature grids is formalized to iteratively optimize the localization as well as to assess the integrity of data sources. The framework is tested using different scenarios from datasets, and the results show the versatility of the proposed model, which is able to provide reliable and consistent integrity estimation in highway as well as semi-urban and urban environments. This method is proven robust against inconsistencies in feature detections such as partial detections, occlusions, and poor map rendering. The method presented claims scalability since it can be implemented with any number of sensors and digital map sources. The only requirement for the applicability of this framework is the ability to detect common features from all of the data sources and represent them geometrically in the proposed feature grid representations. This work also illustrates how classical integrity markers like protection levels can be transposed for perception data sources used in autonomous vehicles.

6. Future Works

The rule-based map rendering technique used in this method is observed to be contributing several inconsistencies, which makes it difficult to isolate map rendering errors from GPS positioning errors. We propose the use of high-definition maps, which are enriched with globally localized lane-level information, to address this issue. Accurate maps will improve the coherence estimation between

features detected from other data sources such as stop lines, pedestrian crossings, lane merging information, road structure information, etc. It will also be important to study map-rendering techniques that improve integrity multi-source perception analysis by including precise building footprints, road width information, lane markings, and traffic sign localization.

Author Contributions: Conceptualization, A.B., S.R.F. and R.R.; methodology, A.B., S.R.F. and R.R.; software, A.B., S.R.F. and R.R.; validation, A.B., S.R.F. and R.R.; formal analysis, A.B., S.R.F. and R.R.; investigation, A.B., S.R.F. and R.R.; resources, A.B., S.R.F. and R.R.; data curation, A.B., S.R.F. and R.R.; writing–original draft preparation, A.B., S.R.F. and R.R.; writing–review and editing, A.B., S.R.F. and R.R.; visualization, A.B., S.R.F. and R.R.; supervision, A.B., S.R.F. and R.R.; project administration, A.B., S.R.F. and R.R.; funding acquisition, A.B., S.R.F. and R.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministère de l'Education Nationale, de l'Enseignement Superieur et de la Recherche, France.

Acknowledgments: We thank Flavien Delgehier for the technical support during this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. International SAE Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2018. Available online: https://www.sae.org/standards/content/j3016_201806/ (accessed on 16 July 2020)
2. Velaga, N.R.; Quddus, M.A.; Bristow, A.L.; Zheng, Y. Map-aided integrity monitoring of a land vehicle navigation system. *IEEE Trans. Intelligent Transp. Syst.* **2012**, *13*, 848–858. [CrossRef]
3. Reid, T.G.; Houts, S.E.; Cammarata, R.; Mills, G.; Agarwal, S.; Vora, A.; Pandey, G. Localization Requirements for Autonomous Vehicles. *SAE Int. J. Connect. Autom. Veh.* **2019**, *2*, 2574–2590. [CrossRef]
4. Palmqvist, J. Integrity Monitoring of Integrated Satellite/Inertial Navigation Systems Using the Likelihood Ratio. In Proceedings of the 9th International Technical Meeting of the Satellite Division of The Institute of Navigation, Kansas City, MO, USA, 17–20 September 1996; pp. 1687–1696.
5. Zinoune, C.; Bonnifait, P.; Ibañez-Guzmán, J. Integrity monitoring of navigation systems using repetitive journeys. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Ypsilanti, MI, USA, 8–11 Jun 2014 ; pp. 274–280.
6. Yang, Y.; Xu, J. GNSS receiver autonomous integrity monitoring (RAIM) algorithm based on robust estimation. *Geod. Geodyn.* **2016**, *7*, 117–123. [CrossRef]
7. Zinoune, C.; Bonnifait, P.; Ibañez-Guzmán, J. Sequential FDIA for autonomous integrity monitoring of navigation maps on board vehicles. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 143–155. [CrossRef]
8. Worner, M.; Schuster, F.; Dolitzscher, F.; Keller, C.G.; Haueis, M.; Dietmayer, K. Integrity for autonomous driving: A survey. In Proceedings of the 2016 IEEE/ION Position, Location and Navigation Symposium, Savannah, GA, USA, 11–14 April 2016; pp. 666–671.
9. Le Marchand, O.; Bonnifait, P.; Ibañez-Guzmán, J.; Betaille, D. Automotive localization integrity using proprioceptive and pseudo-ranges measurements. In Proceedings of the Accurate Localization for Land Transportation, Paris, France, 16 June 2009.
10. Li, L.; Quddus, M.; Zhao, L. High accuracy tightly-coupled integrity monitoring algorithm for map-matching. *Transp. Res. Part C Emerg. Technol.* **2013**, *36*, 13–26. [CrossRef]
11. Ballardini, A.L.; Cattaneo, D.; Fontana, S.; Sorrenti, D.G. Leveraging the OSM building data to enhance the localization of an urban vehicle. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems, Rio de Janeiro, Brazil, 1–4 November 2016; pp. 622–628.
12. Ballardini, A.L.; Cattaneo, D.; Sorrenti, D.G. Visual Localization at Intersections with Digital Maps. In Proceedings of the 2019 International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019; pp. 6651–6657.
13. Kang, J.M.; Yoon, T.S.; Kim, E.; Park, J.B. Lane-Level Map-Matching Method for Vehicle Localization Using GPS and Camera on a High-Definition Map. *Sensors* **2020**, *20*, 2166–2188. [CrossRef] [PubMed]
14. Nedevschi, S.; Popescu, V.; Danescu, R.; Marita, T.; Oniga, F. Accurate Ego-Vehicle Global Localization at Intersections Through Alignment of Visual Data With Digital Map. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 673–687. [CrossRef]

15. Liu, H.; Ye, Q.; Wang, H.; Chen, L.; Yang, J. A Precise and Robust Segmentation-Based Lidar Localization System for Automated Urban Driving. *Remote. Sens.* **2019**, *11*, 1348–1366. [CrossRef]
16. Liang, W.; Zhang, Y.; Wang, J. Map-Based Localization Method for Autonomous Vehicles Using 3D-LIDAR. *IFAC Pap. Online* **2017**, *50*, 276–281.
17. Mueller, A.; Himmelsbach, M.; Luettel, T.; Hundelshausen, F.; Wuensche, H.J. GIS-based topological robot localization through LIDAR crossroad detection. In Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, USA, 5–7 October 2011; pp. 2001–2008.
18. Balakrishnan, A.; Rodríguez F., S.A.; Reynaud, R. An Integrity Assessment Framework for multimodal Vehicle Localization. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference, Auckland, New Zealand, 27–30 October 2019; pp. 2976–2983.
19. Toth, C.; Jozkow, G.; Koppanyi, Z.; Grejner-Brzezinska, D. Positioning Slow-Moving Platforms by UWB Technology in GPS-Challenged Areas. *J. Surv. Eng.* **2017**, *143*, 04017011. [CrossRef]
20. Pusztai, Z.; Eichhardt, I.; Hajder, L. Accurate Calibration of Multi-LiDAR-Multi-Camera Systems. *Sensors* **2018**, *18*, 2139–2161. [CrossRef] [PubMed]
21. Xu, F.; Chen, L.; Lou, J.; Ren, M. A real-time road detection method based on reorganized lidar data. *PLoS ONE* **2019**, *14*, 1–17. [CrossRef] [PubMed]
22. Zheng, S.; Ye, J.; Shi, W.; Yang, C. Robust smooth fitting method for LIDAR data using weighted adaptive mapping LS-SVM. *Proc SPIE* **2008**, *7144*, 5669–5683.
23. Feng, R.; Li, X.; Zou, W.; Shen, H. Registration of multitemporal GF-1 remote sensing images with weighting perspective transformation model. In Proceedings of the 2017 IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 2264–2268.
24. Boritz, J.E. IS practitioners' views on core concepts of information integrity. *Int. J. Account. Inf. Syst.* **2005**, *6*, 260 – 279. [CrossRef]
25. Sandhu, R.; Dambreville, S.; Tannenbaum, A. Particle filtering for registration of 2D and 3D point sets with stochastic dynamics. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
26. Roysdon, P.F.; Farrell, J.A. GPS-INS outlier detection elimination using a sliding window filter. In Proceedings of the 2017 American Control Conference, Seattle, WA, USA, 24–26 May 2017; pp. 1244–1249.
27. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
28. Backman, J.; Kaivosoja, J.; Oksanen, T.; Visala, A. Simulation Environment for Testing Guidance Algorithms with Realistic GPS Noise Model. *IFAC Proc. Vol.* **2010**, *43*, 139–144. [CrossRef]
29. Zhu, N.; Marais, J.; Betaille, D.; Berbineau, M. GNSS Position Integrity in Urban Environments: A Review of Literature. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2762–2778. [CrossRef]
30. PROTECTION LEVEL (specific location). Available online: https://egnos-user-support.essp-sas.eu/new_egnos_ops/protection_levels. (accessed on 16 July 2020).

2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

Towards a Reference Data Generation Framework for Performance Assessment of Perception Systems

Rémi Defraiteur*†‡, Sergio A. Rodríguez F.*†, Marie-Anne Mittet‡, Roger Reynaud*† and Nicole El Zoghby‡
*SATIE Laboratory CNRS Joint research unit - UMR 8029 - firstname.lastname@u-psud.fr
†ENS Paris-Saclay. Paris-Sud University. Paris-Saclay University, 91405, Orsay, France
‡Renault S.A.S - Guyancourt, France - (marie-anne.n.mittet, nicole.el-zoghby)@renault.com

***Abstract*—Sensors and their associated data fusion techniques, play a crucial role in Autonomous Vehicle (AV) decision-making applications. Accurately evaluate performance and reliability of the perception sources is an important task to be able to know the consistency of this data fusion. In this paper, a reference data generation framework for assessing perception sensors performances is proposed. Our approach relies on the complementary use of three data sources: a highly precise 3D map with semantic information, a High Density range finder sensor and a GNSS-RTK/INS localization unit. 3D map provides semantic knowledge of the environment and HD range finder precisely senses ego-vehicle surroundings. Finally, 3D map and HD scans are geometrically associated using positioning information in order to combine them and to infer reference data. Thorough experiments were conducted to evaluate and validate the proposed approach. As a proof of concept, performances of a LiDAR-based road plane detection method were evaluated, quantified and reported in terms of precision and recall.**

I. Introduction

Majority of ADAS (Advanced Driver Assistance Systems) functionalities rely on data transmitted by integrated perception sensors to control the vehicle. Precise performance evaluation of embedded perception solutions is then an important task for AV and ADAS. It is therefore crucial to be able to evaluate performances and reliability of embedded perception solutions. In the aim of evaluating and validating perception solutions, existing approaches that have been proposed in the literature can be structured in three categories: manual, semi–supervised and automatic reference data creation.

Manual labeling: First and less complex method consists in generating referenced datasets by human manual processing. LiDAR (Light Detection And Ranging) datasets close to ADAS functions conditions of use correspond to sequences of point clouds measured from moving sensor. Sequences are recorded to be reusable. Each cloud of the whole sequences is manually segmented and object-level labeling is carried out by hired annotators or crowd-sourcing means. This is the case of KITTI Dataset [1], where the use of both hired annotators and crowd-sourcing were exploited to assign label objects to 3D bounding boxes manually defined. Some approaches were developed to simplify object labeling. In [2], labeling task is facilitated by optimizing points clouds using voxels which are further hand-labeled. Similar techniques are used for images as in [3] where images are hand-labeled. In [4], a database composed of LiDAR and images is addressed. Point clouds are manually labeled and exploited further for referencing images. Finally, in [5], [6] software tools are created to facilitate and speed-up manual generation of reference data for 2D/3D object detection. The main advantage of manual labeling is the high accuracy of reference data created. However, manual labeling is time- and cost- consuming and is not a scalable solution. Indeed, hand-labeled data is specific to each dataset and cannot be applied on new datasets.

Semi-supervised approaches: In [7] a semi-supervised process for reference data generation method was developed in order to annotate scenes collected from KITTI optical flow dataset. This method uses the labeled 3D bounding boxes of KITTI to project them on 2D images and label the pixels of optical flow images. In [8], [9] stereo images are employed to reconstruct 3D point clouds. The created clouds are then classified by trained machine learning algorithm (e.g. random forest algorithm). Finally, the labeled data are re-projected in the 2D image plane. The disadvantage of these solutions is the changing of dimensions between 2D images and 3D point clouds that can bring errors in the data. Moreover, with machine learning algorithm trained on a single dataset, the risk of over-fitting can also bring classification errors in assigned labels.

Automatic methods: Finally, as far as we know, unsupervised approaches generating reference data are based on the use of synthetic data. Sensor simulation tools like Blensor [10], Helios [11] or Carla [12] are used to generate reference data. This is the case in [13], [14], [15] where labeled reference images are obtained in simulated environments. This solution allows the user to generate a big amount of reference data, sometimes very close to real data [16], [17], [18] and scenarios are easy to reproduce with new sensor solutions. However, since the environment is perfectly controlled, synthetic data are not impacted by external errors or other phenomena that only appear under real sensing conditions. To be closer to non-simulated data, reference data generated must take into considerations all the effects that could appear under real conditions.

978-1-5386-7024-8/19/$31.00 ©2019 IEEE

To generate reference data needed for perception algorithms evaluation, our approach aims to generate reference data in a semi-supervised way and combines some advantages of the three cited approaches. It is done by augmenting perception data recorded with a simulated controlled environment. The main strength is the reproducibility and compatibility of our methodology with new datasets recorded, without needing manual labeling for each of the new datasets.

This paper is structured as follows : Section II describes the proposed semi-supervised approach for reference data generation allowing performance evaluation on perception algorithms. This strategy also provides a common referencing support to quantify consistent uncertainties of the perception functions as stated in section IV. Experiments are presented to assess our methodology. An implementation is also deployed to evaluate a 3D road detection algorithm performances. Finally, section V concludes, supported by the presented results, and provides perspectives for future works.

II. Developed approach

A. Framework overview

In order to evaluate perception solutions, it is necessary to have access to reference data. When both reference data and perception data are available for a given scenario, the comparison between data allows us to evaluate the perception solution. The expressed need to be able to evaluate the perception algorithms relies on the availability of reference data.

The proposed framework provides support for offline evaluation of perception algorithms. Thus, perception data are recorded to be replayable and exploitable with our approach. Our aim is to generate reference data without the need of complex classification algorithms or repetitive manual labeling. As it is illustrated in Fig. 1, three data sources compose the proposed methodology : a perception system, a localization system and a 3D map of the environment. They are required for performing the semi-supervised generation of reference data.

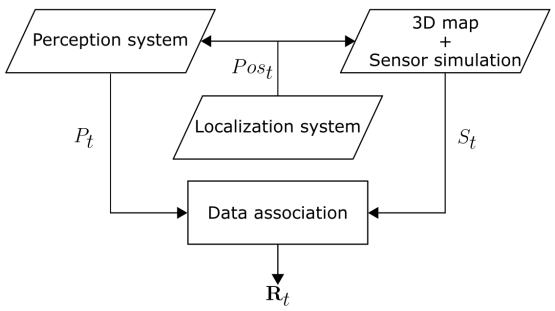


Fig. 1. Pipeline of the reference data generation framework

Perception system: is exploited to record raw perception data P_t acquired under real conditions (e.g. point clouds for LiDAR, images for cameras...), with all the phenomena and external errors. P_t denotes a precise representation of the ego-vehicle surroundings.

Localization system: provides precise localization positions. This information facilitates data association between perception data P_t and model data S_t. To do this, a common reference frame F is exploited for P_t and S_t. Thus, the pose of the perception system recorded at a time t: Pos_t, corresponds to the sensor position and attitude for P_t and is used on the 3D map with the simulator to generate S_t at the same position.

3D map: coupled with a simulation tool allows us to generate model data S_t. The model constituting the geo-referenced 3D map was produced using a precise Mobile Mapping System (MMS) Reigl VMX450. Point clouds obtained by MMS were then manually noise-filtered, post-processed, fused and meshed to create the model. Each object label (e.g. road, lane markings, traffic signs...) is associated with a specific id and every part of the mesh constituting the 3D model is labeled in conjunction with the object represented in the scene. This post-processing is performed only once. Subsequently, new datasets recorded in the locations represented by the 3D map will be able to exploit semantic information from this 3D map without requiring new manual labeling.

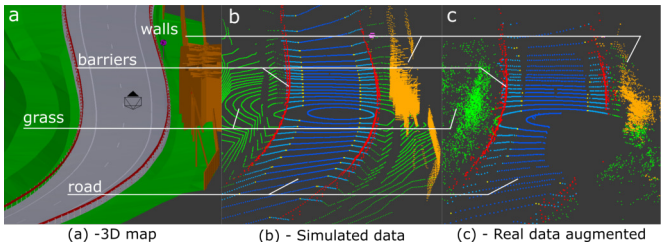


Fig. 2. Comparison between the (a) 3D map, (b) data from the simulated sensor and (c) the real data augmented

3D map is then imported in a settable simulation tool [10], [11], [12]. A perception sensor model with same characteristics of the system used to produce P_t is set and integrated in the simulator. That is, for a given pose Pos_t, data generated by simulation S_t is a model-based perception corresponding to P_t, but contains additional semantic information deduced from the 3D map environment knowledge.

In a last step, a data association method is applied between P_t and S_t in order to transfer semantic environment knowledge contained in S_t on P_t, and outputs reference data $\mathbf{R}_t$. Thus, the combination of these three sources allows us to transfer semantic environment knowledge into the perception data by the way of labeling. Augmented data resulting from this framework represents the reference data and can be exploited in a purpose of evaluation for perception algorithms.

B. System solution and implementation

1) Perception system: To record raw perception data P_t, we oriented our system on LiDAR solution. Several reasons motivate this choice. First, it provides range measurements with a centimeter accuracy and each point measured is

independent of other measurements. Second, it is an active sensor, it is not sensitive to light conditions and can be used to record data at any moment of the day without needing modification on the sensor or on its settings. Last, LiDAR data are represented in 3D. Any further treatment on data is necessary to get 3D information as it is the case with point cloud reconstructed from images. Possible errors introduced by the reconstruction step are thus avoided. These advantages grant access to a reliable and detailed representation of the sensed environment in P_t.

A Velodyne VLP-32C is integrated on the top of our vehicle offering the characteristics listed in Table I. P_t produced by this sensor corresponds to 3D point clouds recorded to accurately scan the environment in detail.

TABLE I
PERCEPTION SENSOR CHARACTERISTICS

Velodyne VLP-32C	
Rotation frequency	10 Hz
Points per rotation	20.000
Vertical angular resolution	0.4°
Azimuthal angular resolution	0.1°
Range	200 m
Range measurements accuracy	+/- 3 cm

2) Sensor simulator: Blensor [10] was used to generate S_t. This simulation tool is completely settable and provides support to new sensor models. LiDAR sensor model characteristics corresponding to the Velodyne VLP-32C were considered. To generate S_t, 3D map is first imported in the simulator. Once the pose of the sensor is defined, simulated point clouds S_t are obtained by ray tracing the 3D map. Ray-object intersection creates impact points stored in the cloud S_t. Each point is constituted by its 3D coordinates and a label value corresponding to impacted object id.

3) Localization system: Finally, the system that plays the link role between P_t and S_t is the localization unit. Positioning accuracy of GNSS solutions in automotive is no better than 2-3 m in good conditions (open sky without outages or multipath effects) [19]. This is not enough to grant access to an accurate data association between S_t and P_t. To perform this task, we oriented our choice on a highly accurate localization solution based on a GNSS-RTK/INS coupling. RTK corrections signals received grant access to centimeter accurate absolute positioning [19], [20], [21]. Coupling with INS (iXblue Atlans-C) gives complementary information of the system attitude. Moreover, high rate of recorded data provided by INS is employed to interpolate the positioning between GNSS-RTK measurements. This configuration outputs an accurate and absolute position of the system.

Positions provided by the localization system are in WGS84 coordinate system. If the 3D map is geo-referenced in a reference frame different from WGS84, transformation is applied on localization data in order to use a common coordinate system F. In our case, the model is geo-referenced in UTM and we define UTM as our common coordinate system F. By the transformation method described in [22], position coordinates from each pose Pos_t, expressed in WGS84, are projected into UTM coordinate system. Sensor pose Pos_t is used to geo-reference P_t and also to generate S_t. P_t and S_t are geo-referenced with respect to the same pose and overlap each-other allowing the data association.

C. Data augmentation

The last processing step allows us to generate reference data $\mathbf{R}_t$ by a data association between P_t and S_t. The purpose is to augment P_t with semantic environment knowledge included in S_t by transfer of information. In this study, five object classes are taken into account:

$$\mathcal{L} = \{road, signs, borders, lanes, N/A\}$$

To realize the data association, objects represented in both P_t and S_t are matched based on a geometrical criterion. Since we oriented our approach on LiDAR data, P_t and S_t correspond to point clouds. Given the big amount of data that point clouds represent, processing them without optimization can be memory consuming and costly in time. Data association is sped-up using an octree structure [23]. Points are then gathered and stored in voxels. Given a 3D query point q, it is then possible to know in which voxel q is situated, which points compose the same voxel and which ones are located in surroundings voxels. Thanks to this optimization, neighbors research method given a specific a research radius r is much sped-up and is used in Algorithm.1 to outputs $\mathbf{R}_t$.

Algorithm 1 Data association algorithm

INPUT: P_t - LiDAR point cloud / S_t - Model point cloud
OUTPUT: $\mathbf{R}_t$ - Reference cloud
q - query point
r - radius of research arround q
n - neighbor found for q
m - number of neighbors found
N - set of neighbors found
for all point $q \in P_t$ **do**
  Find the neighbors $n \in S_t$ situated in r
  if $m > 0$, neighbors n are stored in N **then**
    Label values occurrence in N are counted:
    Case 1 - a label value $\mathcal{L}_j$ is more represented than other labels values: $\mathcal{L}_j$ is assigned to q
    Case 2 - several label values have the same number of votes: Euclidian distance d_i is computed between q and each $n_{i_{1\to m}}$, label value of the closest neighbor $\mathcal{L}_j$ is assigned to q
  else
    No neighbor was found in r, q keeps its initial value N/A
  end if
  q is stored in $\mathbf{R}_t$
end for

According to our approach, S_t is firstly optimized using an octree structure. Each point q from P_t is processed, Algorithm.1 allows us to transfer environment knowledge on P_t and generate reference data $\mathbf{R}_t$.

Augmented LiDAR data $\mathbf{R}_t$ resulting from Algorithm.1 corresponds to reference data and can then be exploited to evaluate performances of perception solutions.

III. Experimental results

A. Methodology analysis

1) Simulation limits: As stated in Algorithm.1, the data augmentation of P_t depends on the radius of research r. The bigger r, the more neighbors will be found and the more P_t points are labeled. However, the larger the number of found neighbors, the more different label values will be found and important labeling errors can be introduced. Two criteria reflecting these aspects are studied.

The first criterion C_1 reflects the quantity of semantic information introduced by our approach on P_t. To quantify C_1, we use a sample of 100 reference clouds obtained by our approach. All points contained in P_t are not "labelisable" since objects non-referenced in the 3D map can be captured. This is the case for moving objects (e.g. passage of other vehicles during data recording) or, as we can see Fig.3.a with captured trees and buildings, irrelevant elements distant from the model and non-referenced in the 3D map Fig.3.b.

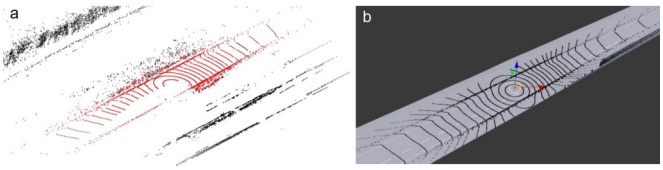


Fig. 3. Comparison between : (a) -P_t with labelisable points (red), non-labelisable points (black), (b) - S_t on the model in Blensor

Taking this aspect into account, we compute the number of "labelisable" points by estimating the fitted 3D bounding box of S_t. The number of P_t points contained in this box correspond to the number of "labelisable" points. Finally, C_1 represents the percentage of labeled points compared to the number of "labelisable" points.

The second criterion C_2 reflects errors of labeling caused by the radius of research r. To quantify C_2, we simulate 100 clouds X_i. X_i' are generated by using the same points stored in X_i and removing semantic labels. Our approach is applied to augment semantic labels on X_i' points. Thereafter, the label value of each point from X_i' is compared to the original value found in X_i. C_2 corresponds to the percentage of wrongly labeled points.

Table.II summarizes the mean values of C_1 and C_2 computed on 100 clouds for five values of r. As reported in Table.II if r is small, risks of wrong labeling is reduced. However, chances to find neighbors in r are also reduced because of geometrical differences between S_t and P_t points position. For further tests, r is set on 0.50m.

TABLE II
Computed criteria C_1 and C_2 according to r

r (m)	0.10	0.20	0.50	0.80	1.00
C_1	28.80 %	59.73 %	80.73 %	85.27 %	87.38 %
C_2	0.11 %	0.72 %	3.21 %	4.59 %	6.94 %

2) Role of the localization data: As explained in II-B3, localization data plays a major role in our approach since the labeling method is based on the geometrical overlap between S_t and P_t. If localization measurements are error impacted or does not reach a sufficient accuracy, offsets between P_t and S_t appear. Therefore, P_t and S_t are not correctly overlapping and labeling errors can appear during data association, affecting the relevance of $\mathbf{R}_t$. We can notice this aspect in the beginning of one of our datasets, where an irregular offset is visible between S_t and P_t (Fig.4.a), before decreasing until an accurate overlap of the data (Fig.4.b).

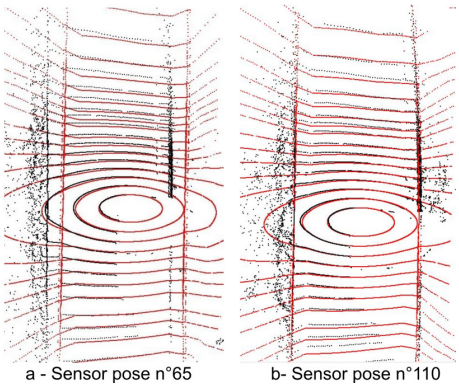


Fig. 4. Offsets between S_t (red) and P_t (black) with : (a) high localization standard deviation, (b) low localization standard deviation.

By studying the standard deviation (σ) of the localization data on position measurements, we can see Fig.5 that the evolution of σ matches perfectly with the noticed offsets. This phenomenon can be explained by an imperfect initialization on the INS causing the high values of σ during the first measurements. Once the INS is fully initialized, the localization measurements become more accurate, σ decreases. The overlapping of S_t and P_t becomes then accurate (Fig.4.b).

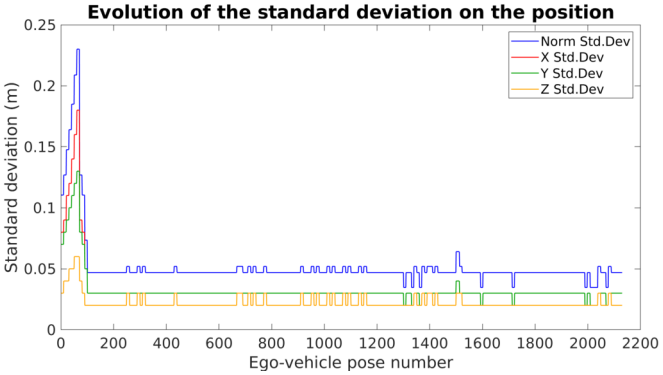


Fig. 5. Evolution of the localization standard deviation on the position

This experiment shows the significance of the localization data in the processing pipeline.

3) Impact of localization errors: Localization data accuracy can be affected by measurements errors as it is the case with the initialization of the INS in III-A2. With this experiment, we evaluate impacts of these errors on reference data created. Data used to lead the test are generated by simulation tool [10].

- First, we use a sequence of 100 sensor poses Pos_i and we simulate the 100 corresponding clouds S_i.
- Second, we introduce centered -additive white Gaussian noise on localization data. Noised poses are named Pos'_i. 100 clouds X_i are simulated with noisy poses Pos'_i but geo-referenced with Pos_i. This way, errors on localization data can be simulated and controlled according to the noise introduced. X_i generated by the simulator are correctly labeled.
- A copy X'_i is made and label value of each point is re-initialized with "N/A". X'_i is then labeled by our approach with S_i.
- Finally, we compare each point between X_i and X'_i and evaluate the percentages of points badly labeled.

Fig.6 represents effects of localization data noise on clouds labeled by our approach. The experiment is leaded several levels of noise introduced in Pos'_i from 0.10m to 5m.

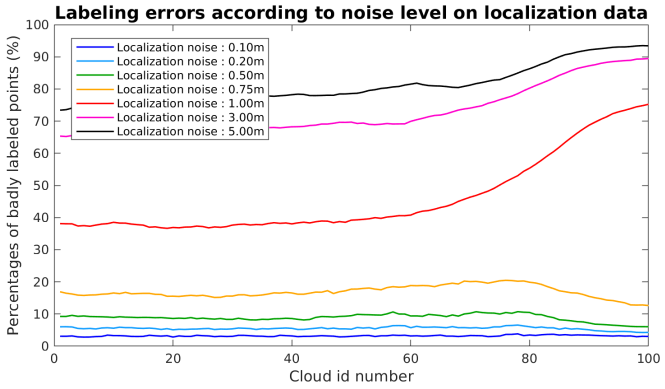


Fig. 6. Labeling errors according to noise level on localization data

The results shown in Fig.6 demonstrates that the accuracy of localization data has to be carefully taken into account in the process. The impact of localization noise is accentuated in turns as we see with increasing curves. Thus, if localization accuracy indicators show a low confidence level, data resulting from our approach is not exploitable as reference data.

B. Application of the approach

In this last experiment, performances of a 3D road detection algorithm [24] are evaluated thanks to reference data generated by our framework. It is worth noting that such an evaluation is suitable for algorithms and sensors performing in 3D and 2D. To operate this test, a sequence of 200 clouds is exploited. Fig.7 illustrates the processing pipeline including the road detection.

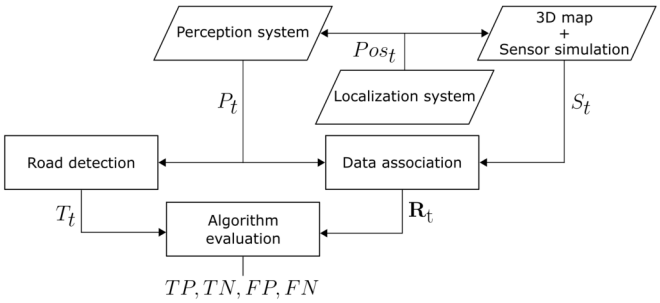


Fig. 7. Steps of the road detection algorithm evaluation

Road Detection : Road detection algorithm performs LiDAR point cloud P_t. All the points that are detected as road receive the label value "$road$". Remaining points keep their initial label value "N/A". The outputs labeled cloud tested is named T_t.

Algorithm Evaluation : The last step of our approach receives both T_t and $\mathbf{R}_t$ to evaluate the processing applied, in our case, the road detection algorithm. Both clouds are composed of the same points, only the label value can be different between peer points. Evaluation metrics : true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) are computed by comparing points label values $\mathcal{L}$ as detailed in Algorithm.2.

Algorithm 2 Evaluation metrics computing

INPUT: $\mathbf{R}_t$ - Reference cloud / T_t - Tested cloud
OUTPUT: TP, TN, FP, FN - Evaluation criteria
for all point $p_i \in T_t$ **do**
  Find the peer point $r_i \in \mathbf{R}_t$
  if $\mathcal{L}_{p_i}$ and $\mathcal{L}_{r_i} = road$ **then**
    TP = + 1
  else if $\mathcal{L}_{p_i}$ and $\mathcal{L}_{r_i} \neq road$ **then**
    TN = + 1
  else if $\mathcal{L}_{p_i} = road$ and $\mathcal{L}_{r_i} \neq road$ **then**
    FP = + 1
  else if $\mathcal{L}_{p_i} \neq road$ and $\mathcal{L}_{r_i} = road$ **then**
    FN = + 1
  end if
end for

Thanks to TP, FP and FN criteria, we can deduce standard metrics recall and precision for each cloud of the sequence. On Fig.8, curves corresponding to standard metrics for each cloud are represented. By analyzing curves, we are able to get a detection performances overview of the tested algorithm and its behavior overall the sequence. Peaks allows us to quickly identify parts of sequence where the performances decrease.

Investigations are then facilitated (Fig.9) to identify why the efficiency of the tested algorithm varies, what are the causes responsible of performance decreases and how the algorithm can be improved to overcome these causes.

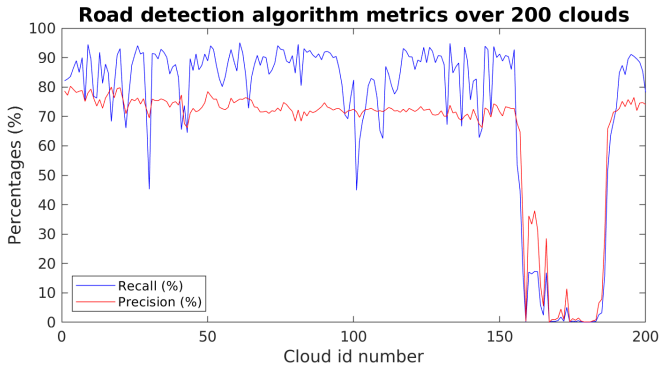


Fig. 8. Evaluation metrics curves

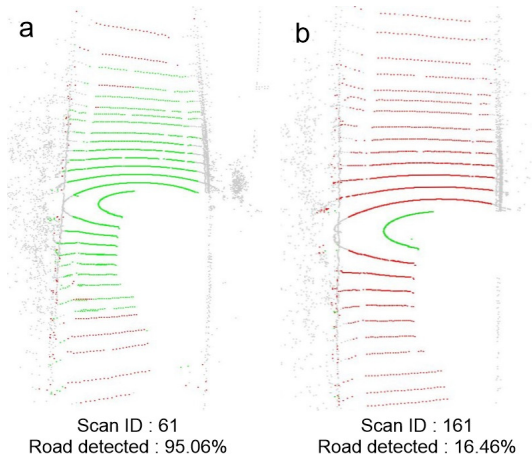


Fig. 9. Comparison of two cases among the algorithm results : (**a**) Accurate road detection rate. (**b**) Inaccurate road detection rate. (Green points = points well classified after comparison to our referencing approach, Red points = points missed after comparison to our referencing approach)

IV. Conclusions

In this paper, a semi-supervised reference data generation method is presented. The joint use of LiDAR data with synthetic data generated from a detailed 3D map, all linked by localization data is exploited to augment semantic information in LiDAR clouds and generate reference data. Transfer of environment knowledge stored in synthetic data on LiDAR data allows us to avoid the need of complex classification algorithms. Thus, labeling errors are greatly reduced. However, sensibilities on the localization accuracy and the neighbors search parameters must be carefully taken into account.

As a perspective, improving the simulated data to make them closer to LiDAR data will allows us to reduce radius of research. Risks of labeling errors will be reduced while keeping a high percentage of transferred semantic information. To deal with localization accuracy variations, clouds alignment methods could also be introduced to support low confident localization measurements. Efforts will be done to evaluate other sensor modalities (e.g. vision systems) and to efficiently manage dynamic objects on the evaluated scenarios.

References

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[2] Y. Xu, Z. Sun, R. Boerner, T. Koch, L. Hoegner, and U. Stilla, "Generation of ground truth datasets for the analysis of 3d point clouds in urban scenes acquired via different sensors," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2018.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[5] T. Tatschke, F. Farber, E. Fuchs, L. Walchshausl, and R. Lindl, "Semi-autonomous reference data generation for perception performance evaluation," in *International Conference on Information Fusion*, 2007.

[6] A. Borkar, M. Hayes, and M. T. Smith, "A novel lane detection system with efficient ground truth generation," *IEEE Transactions on Intelligent Transportation Systems*, 2012.

[7] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[8] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Neural Information Processing Systems*, 2015.

[9] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *European Conference on Computer Vision*, 2008.

[10] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "BlenSor: Blender sensor simulation toolbox," in *Advances in Visual Computing*, 2011.

[11] S. Bechtold and B. Hofle, "Helios : a multi-purpose lidar simulation framework for research, planning and trainig of laser scanning operations with airborne, ground-based mobile and stationary platforms," *Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2016.

[12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on Robot Learning*, 2017.

[13] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[14] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," *Computing Research Repository*, 2017.

[15] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" *Computing Research Repository*, 2016.

[16] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *International Conference on Computer Vision*, 2017.

[17] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox, "What makes good synthetic training data for learning disparity and optical flow estimation?" *International Journal of Computer Vision*, 2018.

[18] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A lidar point cloud generator: from a virtual world to autonomous driving," *International Conference on Multimedia Retrieval*, 2018.

[19] J.-M. Zogg, *GPS Essentials of Satellite Navigation Compendium*, 2009.

[20] Y. Feng and J. Wang, "GPS RTK performance characteristics and analysis," *Journal of Global Positioning Systems*, 2008.

[21] B. Scherzinger, "Precise robust positioning with inertial/gps rtk," *ION-GPS*, 2000.

[22] J. P. Snyder, *Map projections–A working manual*, 1987.

[23] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the Association for Computing Machinery*, 1975.

[24] S. Rodriguez, V. Fremont, P. Bonnifait, and V. Cherfaoui, "An embedded multi-modal system for object localization and tracking," in *IEEE Intelligent Vehicles Symposium*, 2010.

HOOFR SLAM System: An Embedded Vision SLAM Algorithm and Its Hardware-Software Mapping-Based Intelligent Vehicles Applications

Dai-Duong Nguyen, Abdelhafid Elouardi, Sergio A. Rodriguez Florez, and Samir Bouaziz

***Abstract*— This paper deals simultaneously with the trajectory estimation and map reconstruction by means of a stereo-calibrated vision system evolving in a large-scale unknown environment. This problem is widely known as Visual SLAM. Our proposal optimizes the execution time of the VSLAM framework while preserving its localization accuracy. The contributions of this paper are structured as follows. First, a novel VSLAM approach based on a "Weighted Mean" of multiple neighbor poses is detailed and is denoted as HOOFR SLAM. This approach provides a localization estimate after computing the camera poses (6-DOF rigid transformation) from the current image frame to previous neighbor frames. Taking advantage of the camera motion, we conjointly incorporate two types of stereo modes: "Static Stereo" mode (SS) through the fixed-baseline of left-right cameras setup along with the "Temporal Multi-view Stereo" mode (TMS). Moreover, instead of computing beforehand the disparity of SS mode for all key-points set, the disparity map in scale estimation step is limited to the inliers of the TMS mode so as to reduce the computational cost. This strategy is suitable to be parallelized on a multiprocessor architecture and exhibits a competitive performance with the other state-of-the-art strategies in many real datasets. Second, we report a hardware-software mapping of the proposed VSLAM approach. To this end, a heterogeneous CPU-GPU architecture-based vision system is considered. Third, a thorough and extensive experimental evaluation of our algorithm implemented on an automotive architecture (the NVIDIA Tegra TX1 system) is studied and analyzed. We report hence the localization and timing results through experiments on five well-known public stereo SLAM datasets: KITTI, Malaga, Oxford, MRT, and St_Lucia datasets.**

***Index Terms*— Visual SLAM, scene recognition, feature extraction, hardware-software mapping, embedded systems.**

I. Introduction

VISUAL Simultaneous Localization and Mapping (VSLAM) is a key issue in computer vision and robotics community. VSLAM aims at estimating the camera trajectory while reconstructing a consistent 3D model of the environment. In the literature, existing approaches are based in two predominant perception strategies: monocular and stereo. Stereo VSLAM is generally transposable to RGBD systems [1], [2]. The most versatile of VSLAM approaches is the monocular VSLAM [3]–[6] since its hardware requirement is only one camera to observe the environment. However, "scale drift" remains an open problem of this approach. This is due to the fact that frame-to-frame motion estimates are integrated over time up to an absolute global scale. On the other hand, stereo VSLAM uses two calibrated cameras to capture the scenes so the depth from camera to points can be computed for each frame using the disparity.

Over almost two decades, there have been many successful stereo VSLAM methodologies. Early stereo VSLAM frameworks were based on classical EKF approach enclosing a large Extended Kalman Filter for managing all landmarks [7]–[9]. This approach suffers from two main problems: firstly, the quadratic complexity of the EKF limits the number of processed landmarks; and secondly, the consistency of EKF is known to be poor causing the impossibility of re-linearizing the cost function. In order to tackle such drawbacks, Paz *et al.* [10] proposed to split a large EKF filter into sub-maps. However, this variant limits the application to environments with an area of 100m². An alternative variant is FastSLAM [11] which represents trajectories by means of a set of particles and small EKF filters are assigned to each landmark. FastSLAM framework is also afflicted by its complexity when the number of particles is not bounded for a given environment. It also suffers to achieve loop-closures due to the 6-DOF nature of visual SLAM, making this approach not well-suited for large-scale scenarios.

Since EKF and FastSLAM are filter-based frameworks, they marginalize all past poses and summarize the information gained over time with a probability distribution. In contrast, keyframe-based VSLAM approach performs a windowed optimization (e.g. bundle adjustment) on a small set of past frames to optimize current pose. Then, the global optimization in case of loop closure could be done using Graph-based SLAM method [12], [13]. Strasdat *et al.* [14] compared filter and keyframe-based VSLAM demonstrating that the latter achieves a better balance between computational cost and precision.

Manuscript received December 7, 2017; revised July 3, 2018 and September 17, 2018; accepted October 28, 2018. Date of publication December 3, 2018; date of current version November 6, 2019. The Associate Editor for this article was Z. Duric. *(Corresponding author: Dai-Duong Nguyen.)*

The authors are with SATIE - UMR CNRS 8029, University of Paris-Sud, Paris-Saclay University, 91400 Orsay cedex, France (e-mail: dai-duong.nguyen@u-psud.fr; abdelhafid.elouardi@u-psud.fr; sergio.rodriguez@u-psud.fr; samir.bouaziz@u-psud.fr).

Digital Object Identifier 10.1109/TITS.2018.2881556

The requirement for SLAM algorithms in terms of calculation, accuracy and embeddability is a critical factor limiting the use of existing approaches in embedded applications. Meanwhile, trends towards low cost implementations and low power processing require massive parallelism and implementation on heterogeneous architectures.

II. Related Works

Inside keyframe-based VSLAM methodologies, the use of two different data representations can be highlighted: feature and image based. Feature-based strategy comes out earlier and was inspired on several researches. For instance, a scalable stereo visual SLAM have been introduced in frameSLAM [15] and RSLAM [16]. The contribution of frameSLAM consists in reducing the complexity of large loop-closures by constructing sub-maps and simplifying feature constraints into frames constraints. In this way, the mapping task was optimized so as to maintain a subset of frames (skeleton). RSLAM implements a local bundle adjustment with a bounded complexity in order to provide an accurate map and trajectory. Even if RSLAM achieves constant time complexity, the global consistency is not warranted. S-PTAM proposed by Pire *et al.* [17] exploits, in parallel, the tracking and mapping in order to achieve real-time performances. However, it lacks large loop closing which is indispensable in an accurate SLAM system. Recently, ORB-SLAM appears to be one of the most actively developed VSLAM framework. After the monocular version, Mur-Artal *et al.* contributes with a stereo version of ORB-SLAM in [1] to handle the problem of scale drift. They inherit the main spirit of S-PTAM and complement it with a loop closing procedure. A first dense image-based approach is latterly presented in which LSD-SLAM [18] is named as candidate. This approach provides depth estimation and mapping by direct image alignment with affine lighting correction on a rich set of pixels having a high intensity gradient. LSD-SLAM provides good results under low image resolution and small camera motions. The use of high resolution images or video sequences with an important interframe camera motion with LSD-SLAM provides poor localization results and its computational cost becomes a severe issue. An alternative image-based approach was presented recently and named as DSO [19]. Such a method performs following a direct visual odometry framework and does not incorporate loop-closure detection, correction and re-localization. Besides, the authors state that DSO cannot handle dynamic scenes.

The implementation of SLAM algorithms in embedded architectures is often preceded by an algorithm-architecture-adequacy study, which allows formal verifications as soon as possible to warrant the feasibility of the design and to reformulate optimization problems so as to exploit at the best the target architecture. Rodriguez-Losada *et al.* [20] analyzed the acceleration of a laser SLAM on two desktop GPUs: GF8400M and GTX280. The speed-up factors achieved are respectively 8 and 57 in comparison to the execution on a T7250 CPU (@2GHz). More recently, Whealan *et al.* [21] evaluated his approach for dense visual SLAM based RGB-D camera on a powerful heterogeneous system consisting of an Intel CPU (i7 - 3.4GHz) and an Nvidia GeForce GTX 780 GPU. A fast execution is achieved where the average time ranges from 31ms to 45ms per frame. Heterogeneous architectures (CPU-GPU, CPU-DSP or CPU-FPGA) are a common trend nowadays on computing platforms, specially for embedded systems. Therefore, many researchers took advantage of these architectures to accelerate SLAM applications. Vincke *et al.* [22] proposed an efficient EKF-SLAM system based on a low-cost and heterogeneous architecture. The hardware contains an ARM processor, an SIMD coprocessor (NEON) and a DSP core. The system implements low-cost sensors: a camera and odometers. The emergence of embedded systems has lead to several works addressing the embeddability issue of SLAM algorithms. However, few works deal with hardware-software mapping of VSLAM algorithms on embedded architectures which recently offers a high potential to have a great improvement in designing VSLAM systems.

Our Contribution: Inspired from the works in the state-of-the-art where a hardware-software mapping of SLAM algorithms were proposed and evaluated on heterogeneous architectures, we present a novel end-to-end study of an original feature-based stereo VSLAM framework. Our system is designed to work with stereo calibrated cameras, integrating our previous HOOFR extractor [23] for extracting features and matching. The processing is structured to maximize the parallelization. The algorithm complexity is optimized to be suitable for an embedded device. Hence, instead of optimizing camera poses by high cost bundle adjustment on keyframes or saving the map-points history, this algorithm is intended to achieve accurate position for each input frame. In practice, relative poses of current input frame with a set of previous neighbor frames are estimated. The optimal pose is achieved as the mean of relative poses with weighted factors. We name this variant of visual SLAM as HOOFR SLAM. The second contributions of this work consists in presenting a detailed study related to mapping the HOOFR SLAM algorithm on a heterogeneous embedded architecture. So, thanks to a hardware-software mappnig process, the proposed algorithm was implemented on an automotive embedded platform (NVIDIA Tegra TX1) with a low-power CPU associated to a GPU for hardware acceleration. A thorough and extensive experimental evaluation of our algorithm is studied and analyzed. As a third contribution, we present localization and timing results through experiments on five well-known public stereo SLAM datasets (KITTI, Malaga, Oxford, MRT and St_Lucia datasets).

III. HOOFR SLAM Algorithm

A. Algorithm Overview

The transformation (translation and rotation) of a stereo camera can be computed in homogeneous coordinates (up to scale) by one image chain (left or right). Therefore, in our system, we employ only the left image for relative motion estimation between two camera positions. The right image is used in further step to calculate the real scale. For each input stereo frame, HOOFR features are extracted in the left image and HOOFR description is then computed for both motion

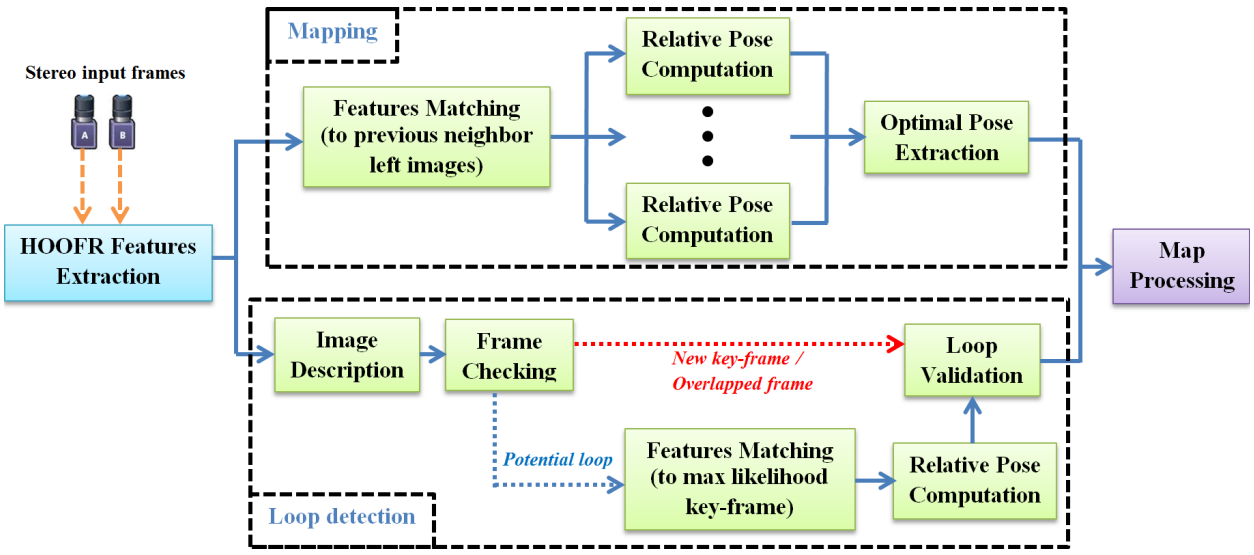


Fig. 1. Functional blocks of the algorithm flow at each input stereo frame.

estimation and loop detection. HOOFR features are matched with those of previous left image in order to estimate relative transformations between the current frame and the previous frame. We define the "previous neighbor frame" (PNF) as the frame that the camera transformation is successfully estimated from it to current frame through essential matrix.

Essential matrix (E) [24] corresponding to one relative transformation is computed from the matching sets using RANSAC [25]. The camera translation, camera rotation and landmarks positions are extracted from E by triangulation but they are in homogeneous coordinates (up to scale). In order to have a real scale, we use stereo matching to match the position of triangulated landmarks in the left image to those corresponding in the right image. The real landmark-camera distance is computed based on this stereo matching and the distribution of the left and right cameras. Scale factor is the ratio of real distance on the triangulated distance. Finally, the camera motion is estimated as the product of the homogeneous motion and the scale factor. To achieve the optimal camera pose, the main idea of our design is that we do not employ bundle adjustment which presents a high processing cost. Instead, we propose another method denoted as "windowed filtering" which estimates camera pose of current frame from a set of previous neighbor frames. For each previous neighbor frame, we apply the entire motion estimation to achieve one prediction of current pose. Each predicted pose is associated with a weight corresponding to its confidence in comparison to others predictions. The optimal current pose is then the mean of all predictions by their weights respectively.

In parallel with current-to-neighbors motion estimation, we perform a loop detection test for left image. HOOFR binary feature description is used one again to extract image description. The current left image is queried in key-frame set to find the max-likelihood. In the case of low matching score, current image is considered as a new key-frame, we add current position attached with its image description to pose graph. In contrast, potential loop closing is considered when high matching score is presented and the max-likelihood key-frame is far from current frame in pose graph. The motion estimation between current frame and max-likelihood key-frame is then computed to validate the loop closure. A real loop is taken into account only if motion estimation is successful.

Our algorithm pipeline is shown in detail in figure 1. After HOOFR features extraction, we launch at one time the Loop detection and Mapping threads. Inside Mapping thread, Features Matching block finds the correspondences for each key-point of the current frame in each PNF. We offload this block to GPU due to its computational cost. Then, a number of Relative Pose Computation tasks are executed, each of them computes one predicted camera pose from one PNF. The number of PNFs ($nframes$) hugely depends on the camera movement speed. However, in practice, due to the architecture constraints, $nframes$ is fixed to 3 or 4 for the maximum number of neighbor frames. The "Optimal Pose Extraction" block evaluates the predictions to get an optimal current pose. Otherwise, inside Loop detection thread, Image Description block describes the current frame by comparing the descriptions of relevant key-points to a bag of words (BoW). The image description is then passed to Frame Checking block to find the max-likelihood in key-frames set. We define an overlapped frame as a frame having a max-likelihood near to it in pose graph with high matching score. Normally, when we have a new key-frame, some of the following frames could be overlapped frames. Features Matching and Relative Pose Computation between current frame to max-likelihood key-frame in case of potential loop are processed by stricter condition than that of Mapping thread to warrant an accurate loop closure. "Map Processing" block gathers the result of two main threads (Loop detection and Mapping). It always updates current pose and points to the map if mapping is successful,

updates the key-frame set if loop detection determines that current frame is a new key-frame or corrects the map by distributing error along the pose graph when a real loop is presented.

B. HOOFR Features Extraction

1) Bucketing HOOFR Features: HOOFR [23] extracts key-points that are used for motion estimation. To ensure a precise estimation, many correspondences are required so that many points should be detected and described in an image. Due to this reason, we need a high speed extractor. HOOFR detector is the combination of ORB detector with Hessian score and provide better compromise between execution time and matching precision [26]. The main idea of HOOFR detector is that it detects features in an image by applying FAST detector over multiple scales of an image pyramid. Then, the detected features are filtered to keep the most relevant key-points based on their Hessian score (instead of Harris score in ORB). This filtering provides a good repeatability. It is eliminated in the processing flow of some works such as LSD-SLAM [27]. However, in our system, we maintain this filtering step to improve the matching result for an enhanced pose estimation.

In order to warrant a homogeneous distribution of features, we employ bucketing technique as used in all others systems. The input frame is divided into a grid where the number of cells depend on the image resolution. HOOFR features are then detected with adapting threshold trying best to extract enough points. We fix the maximum number of points retained in one cell to pts. In each cell, at the first detection, FAST threshold is set to a high value fa. Unless the number of key-point is higher than pts, the second detection is operated with lower FAST threshold value ($fa/2$). After detection, if the number of points is higher than pts, we compute Hessian score for each point and maintain only pts points having the highest score. The orientation and description are computed by HOOFR descriptor for the most relevant points retained by each cell. HOOFR descriptor (256 bits) is an enhanced version of FREAK (512 bits). It has a low sensitive to viewpoint and is fast to compute and match.

2) Binary Descriptor for Place Recognition: Scene recognition is the fundamental step for loop closure. Typically, this step uses SIFT or SURF full-featured descriptors due to their high matching score among the existing approaches. Nevertheless, their computational cost has degraded performances of SLAM systems. Recently, binary bag of words [28] is proposed with a competitive performance by an order of magnitude faster than floating approaches. this approach is widely used and has remarkable results in visual SLAM systems such as [4] and [29]. Thus, in our system, we integrate the idea of binary word so as to keep place recognition process light. Among the relevant key-points provided by HOOFR detector in the whole image, we select K points ($K = 150$ in our implementation) having the highest Hessian score to get corresponding words based on their HOOFR description. Image description is built from these binary words.

C. Mapping Thread

1) Features Matching: In mapping thread, features matching is carried out between the current frame and the previous neighbor frames. We note that the camera frequency is high (10-50 fps) leading to a little change between consecutive images. For this reason, the correspondence in neighbor frame is located not too far from key-point position in current frame, so we can limit the searching region instead of the whole image. For each key-point I in PNF, we perform "Brute-Force" matching with all key-points locating in same cell or "neighbor cell" in the current frame. We find the most and the second correspondences (J and J') by the smallest and the second smallest Hamming distance respectively. The result of feature matching has an important role in the precision of pose estimation so that it must be done carefully. Hence, we apply further three following conditions to select pairs among the most matching pairs $I - J$ (smallest Hamming distance):

Firstly, the matching pair must have a high distinction in comparison to its second matching. It means that the ratio of Hd_{I-J} to $Hd_{I-J'}$ must be lower than a threshold φ where Hd_{I-J} represents the Hamming distance of the pair $I-J$. The value of φ is 0.85 giving a good exhibition in our experiments.

Secondly, if the positions of I and J have a small difference in the images ($\|p_I - p_J\| < 2$), it means probably that the point is too far from camera or the camera does not move significantly compared to the previous pose. Such two cases do not provide a good estimation so that these matches should be also rejected. Furthermore, if too many matches have a small position change, the camera is considered staying nearby the previous pose.

Thirdly, in contrary to the second condition, if I and J have too big differences in positions ($\|p_I - p_J\| > 90$), it could be a false matching and also must be eliminated.

In some other researches like ORB-SLAM or LSD SLAM, people use guiding search to find correspondences. They rely on the last transformation of camera and point position in the previous frame to predict the point position in the current frame. This method has a good performance when the transformation is small and stable but it is easy to loose the tracking when the transformation becomes more critical. In our algorithm, we apply Brute-Force to find the best candidate in a large set of local features. After 3 test conditions above, we get a reliable matching set for the following step.

2) Relative Pose Computation: The goal of Relative Pose Computation (RPC) block is to compute the relative pose between two frames (always from left images of a stereo camera) and to triangulate a set of map points. There are many RPC blocks executed in parallel. We defined these execution as sub-threads inside mapping thread. Each of them estimates one relative motion from current frame to one previous neighbor frame. RPC block consists of 3 principal steps: rotation and translation extraction from essential matrix, solution determination and scale estimation. We assume the image domain to be given in stereo-rectified coordinates, the intrinsic (focal length, center points) and extrinsic (baseline, relative angle) camera parameters are calibrated a-priori.

- Rotation (R) and Translation (t) Extraction From Essential Matrix (E): Essential matrix is estimated from HOOFR matching set returned by the previous step. The epipolar geometry is described by equation (1):

$$[p_I; 1]^T K^T E K [p_J; 1] = 0 \tag{1}$$

where K is the intrinsic camera matrix, p_I (x_I, y_I) and $p_J(x_J, y_J)$ are respectively positions in PNF and current frame of a correspondence $I - J$. Each matching pair gives a constraint to solve E. In others works such as ORB-SLAM, people use 5-point algorithms [30] inside a RANSAC scheme to extract an optimized model E_{op} from matching set. They assume a standard deviation of 1 pixel in the measurement error. Then, they consider R and t extracted from E_{op} as the initial state for the optimization of bundle adjustment (BA).

In our proposal, we intend to avoid BA which has a high computational cost. Hence, we focus on the method to improve the precision of estimating E, which makes the most difference of our system with others in state of art. Before computing E, number of matching pairs (np) in each RPC block is checked. If np is under a threshold λ, the corresponding RPC block is considered as an invalid estimation and its sub-thread will be stopped immediately. In contrariwise, E estimation is processed when np is bigger than λ. Through experiments, we found that a high precise localization is presented when the measurement error (me) of inlier in RANSAC scheme is smaller than 0.4. However, when we apply RANSAC with $me = 0.4$ to the initial matching set, the execution time severely increases. The reason is that the number of iterations in RANSAC is updated during the estimating process. After each iteration, the remaining number of iterations is computed by equation (2):

$$N_I^i = max(N_I^{i-1} - 1, log\frac{1-c}{1-(n_e/N_e)^5}) \tag{2}$$

where N_I^i is the remaining number of iterations at time i, c is the parameter of confidence (normally between 0.95 and 0.99), $n_e$is the number of inliers in the best model at time t and N_e is the total number of elements in the whole set. When the measurement error is smaller than 0.4, it is obvious that n_e decreases leading to the increase of remaining number of iterations. Therefore, in order to accelerate the processing, we propose the Algorithm 1 for estimating E_{op}:

We mark the inliers of E_{op}, while outliers are rejected. Given that E_{op} has been determined; our method for estimating rotation R, translation t and 3D points triangulation is based on performing single value decomposition (SVD) of E (mentioned in Hartley & Zisserman's book [24]). Due to the fact that E is "up to scale" so that SVD provides the solution of $[t]_m$ in homogeneous coordinates (scale is not defined). Furthermore, we have 2 opposite directions which are possible for translation (t) and two different rotations (R) which are compatible with an essential matrix. This gives four classes of solutions in total for the relation between two camera coordinates. However, there is only one correction solution where the triangulated point is in front of camera at both positions (current and reference positions).

Algorithm 1 Essential Matrix Estimation From Matching Set

1) Apply 5-points algorithm inside RANSAC scheme to the initial matching set with the measurement error equal to 1.0.
2) Select the inliers corresponding to the optimal model of step 1 to form another set (refined matching set).
3) Apply 5-points algorithm inside RANSAC scheme to the refined matching set with the measurement error equal to 0.4 to get a final optimal model (E_{op}).
4) Test the final optimal model of step 3 on the whole initial matching set to select the inliers (measurement error reclaims the value of 1.0).
5) Compute the mean of measurement errors returned by inliers from step 4. The inverse of this value represents the score of the estimated model.

- Solution Determination: In order to select the correct solution among the four possibilities, for each inlier matching pair, we compute 3D triangulated position in the 4 solutions. The point is arranged to the solution in which it is in front of camera at both reference and current positions. The chosen solution is the one containing the most points seen in comparison to others. In theory, if the estimation of E is noiseless, one solution will contain all triangulated points. In that case, we can check only one matching pair to find the valid solution. However, matching is affected by noise in practice, so checking all matching pairs provides a more robust method. In particular, if image is too degraded by noise leading to no clear winner solution, the relative pose estimation of the corresponding sub-thread is stopped immediately and will be marked to be invalid.

- Scale Estimation: As the essential matrix is "up to scale", the translation and 3-D triangulated points computed above are in unit coordinates. Therefore, the residual problem after selecting the valid solution is determining the "real scale" of map and camera motion.

The most advantage of a stereo camera is providing two images from different physical cameras, taken at the same time. Hence, the depth from 3D point to camera can be estimated without scale ambiguity using stereo-disparity (static stereo). Assuming that we have a point in the left image, the correspondence of this point is searched along epipolar line in the right image. In our case of rectified-stereo, this search is performed along the horizontal lines.

As stereo correspondence measure, we use 5 pixels-SSD method [31] along the scan-line. In our system, we obtain *a-priori* a point in the left image. Hence, if we consider the same position in the right image, the correspondence is located undoubtedly on the left side of this position. In practice, the disparity range in the right image is constrained to $[(x_J - \sigma, y_J), (x_J, y_J)]$ where σ is the limited search region (σ=30 in our experiment). Once the correspondence is defined, the real 3D point $\bar{P}_J(\bar{X}_J,\bar{Y}_J,\bar{Z}_J)$ with reference to camera will be extracted by well-known static-stereo triangulation (using disparity, baseline and camera focal length)

as mentioned in [24]. We compute the real distance for all triangulated points arranged to the selected solution. Scale factor (k_J) is the ratio of the real distance of static stereo on the triangulated distance of temporal stereo. In fact, this factor is simply computed by the ratio of ($\bar{Z}_J/Z_J$). In the case of noise absence, all points have the same scale factor. However, it is never the case in practice. To have an appropriate value, we consequently employ 1-point scheme on the scale factor set as in Algorithm [2].

Algorithm 2 RANSAC Scheme for Scale Estimation

1) Take the value (k_{av}) of one element in the factors set.
2) Find the number of inliers in the entire set. A factor k_J will be classified as an inlier if the difference is small enough ($|k_J - k_{av}|/min(k_J, k_{av}) < \varepsilon$). ε is set to 0.1 in our test. Mark the scale value if it is the best model (contain the most inliers).
3) Repeat the processing for all other elements in factors set. The value of the best model is considered as the estimated scale.

In order to have reliable scale estimation, after doing 1-point scheme, we additionally evaluate the best model if the number of iterations reaches to the bound. The model is invalid when the number of inliers do not attain an acceptable value ($N_{inliers} < \gamma$). In this case, we also reject the current process, the sub-thread returns invalid estimation. Otherwise, camera translation and 3-D point position are multiplied with the scale to get the non-scale value and only 3-D points computed from inliers are maintained. Through experiments, we found that the value of γ is set to 10 giving a good performance.

3) Optimal Pose Extraction: This block is the summary step in mapping thread and takes into account all predictions from sub-threads to calculate the optimal camera pose. In practice, for each sub-thread, we notice that relative pose is extracted from Essential matrix which is obtained beforehand from features matching set. Therefore, we propose to use inverse of mean error retained by inliers after essential matrix estimation as a weight factor of predicted pose. Equation (3) shows the computation of optimal current left camera pose C^l (also defined as $[R|t]$ in some references) from all predictions:

$$C^l = \Sigma_{n=1}^{N} \frac{\sigma_n}{\Omega} \hat{C}_n^l \tag{3}$$

where $\hat{C}_n^l$ is the predicted position of the sub-thread Th_n, σ_n is the inverse of mean error of inliers and $\Omega = \Sigma_{n=1}^{N} \sigma_n$. Besides, N represents the number of valid sub-threads which compute a prediction with positive weight. Contrariwise, when a sub-thread is marked to be invalid, its weight takes the value of 0 and it will be ignored in optimal pose extraction. When all prediction are invalid, current frame is not tracked. In this case, map is not updated and we proceed directly to the next frame.

D. Loop Detection Thread

Loop detection thread runs in parallel with mapping thread. It processes the current frame and tries to detect a loop closure.

1) Image Description: For loop detection and re-localization, our system implements a bag of words place recognition based on FAB-MAP 2.0 module exhibiting a robust performance as shown in [32]. We use the HOOFR key-points and their HOOFR descriptor to extract the bag of words in a large images set. A 256-bit binary descriptor contains in total 2^{256} different words. However, a huge vocabulary not only takes much time to build image description but also has a poor efficiency in loop detection. The issue is that we have a low tolerance when too many words are maintained in the vocabulary. In such case, a 3D point will be assigned easily to two different words when camera has little position change. Consequently, a low similarity between 2 images is presented through these images of the same scene. In experiments, we found that a vocabulary of 10000 words provides a favorable compromise between precision and execution time. The vocabulary is created offline one time from a large set of random images and is used for all test sequences.

2) Map and Key-Frame Set: In our system, map is represented as a set of $M_i = C_i^l, T_i^l, L_i$. Each M_i contains position of left camera C_i^l in global coordinates, relative transition T_i^l to previous left camera position and all 3-D landmarks positions L_i in camera coordinates. Our developed system is similar to recent SLAM systems that do not consider all processed frames as key-frames due to 2 main reasons:

- For each input frame, each element in key-frames set will be queried to compute the likelihood percentage. The frame sampling is aimed to reduce the size of key-frames set. Hence, the computation will be light. This strategy is suitable for implementing the application on an embedded system where memory resources are limited.
- There exists always an overlap between consecutive frames. It means that when images are taken from close times, they contain many common words. In many cases, two images take exactly the same words from environment. The overlap will cause problem in computing likelihood percentage when these two images attain the same value. In this case, all percentages have small values causing the ambiguity in loop detection.

Therefore, we consider a frame as a new key-frame when there is no likelihood percentage value bigger than η (0.99 in our experiments). Each key-frame is then updated into key-frame set $KE_i = id_i, V_i, D_i$ where id_i is the index of the key-frame position in the map, V_i and D_i are respectively features and their HOOFR description extracted from key-frame.

3) Frame Checking : First of all, in "Frame Checking" block, K binary words retrieved from the most K relevant features in HOOFR extraction are employed as well as vocabulary to build image description. Specifically, each of K binary words will be queried in vocabulary to find the best matching word (lowest Hamming distance). Image descriptor is formed by taking into account which words that image takes from the vocabulary. Likelihood percentage is then computed for all elements in key-frame set based on their image descriptor. If the maximum likelihood is less than η, "Frame Checking" decides that current frame is a new key-frame and we will update it to the key-frame set in "Map Processing" block.

When maximum likelihood is bigger than η, the frame is not a new key-frame. However, we fall into two possibilities: the overlap with previous images or potential loop detection. In practice, to manage key-frame set, a variable called "historical time" (ht) is additionally attached to each element in the set. Once key-frame is added to the set, this value is initialized as the size of the set at that moment. Besides, when loop closing is successfully processed at this key-frame, ht is updated by the size of the set at the update moment. A "newcomer" (newly added or processed) is identified when ht is closely to actual size of key-frame set. Moreover, after a loop closing, it is probably that we have many loop points nearby. In order to avoid too many loops processing, we count the number of new key-frames added from a loop point. "Frame Checking" recognizes a potential loop when two conditions below are satisfied:

- Historical time of maximum likelihood key-frame ht_m is smaller than the size of actual key-frame set by t ($ht_m < keyframeset.size() - t$).
- Ne new key-frames have been already added from the last loop point.

where t, Ne are respectively set to 5 and 10 in our experiments. Otherwise, it is recognized as an overlapping frame.

The features matching and relative pose estimation between current frame and maximum likelihood frame are performed only in the case of potential loop. We use the word "potential" because the current frame must finally be validated by pose estimation. In our experiments, most of the frames are recognized as a new key-frame or overlap with the previous frame. Features matching and Pose estimation tasks are processed only when a loop point is closely attained. Nevertheless, we found that some particular frames are potential loops but they are not the real loops. This is inevitable and it occurs when two images of different places take too many common points in the vocabulary. However, these frames are rapidly rejected after features matching due to the lack of valid correspondences or rejected in pose estimation based RANSAC since there is not enough inliers.

4) Features Matching: In loop detection thread, features matching block is between current frame and its max-likelihood frame when a potential loop is detected. As a precise loop requires severe checking conditions, we propose to use "cross Brute-Force" matching instead of the high distinction checking. The idea is that we keep the second and the third checking conditions as in features matching of mapping thread. However, we change the first condition as following:

- Two feature sets are matched using local Brute-Force matching in two direction. For each point I in the max-likelihood frame, we find the best local match J (smallest Hamming distance) in the current frame and vice-versa.
- The matches verify the first condition if they have the same matching results in two direction ($I \rightarrow J$ and $J \rightarrow I$).

This stricter condition allows us to detect the "false positive" of potential loop (a high similarity but not a same scene) where few matching pairs retained after checking.

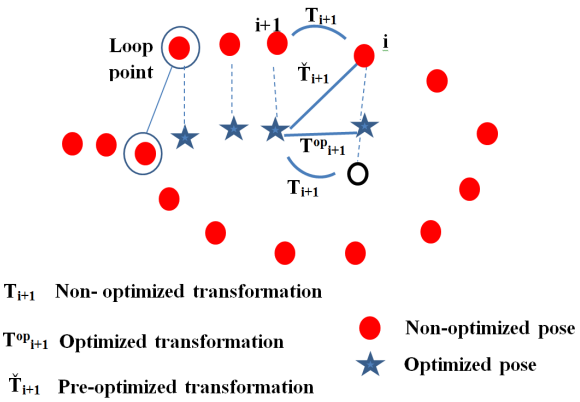


Fig. 2. Loop correction.

5) Relative Pose Estimation: RPC block in loop detection is similar to that in mapping thread except the change of the threshold λ. We also increase the value of λ to insure that only "true positive" of potential loop is handled. The reason is that after a tightening matching, we require more number of matching pairs retained to compute essential matrix. In experiments, we found that this combination exhibited a tremendous performance with no "false positive" loop passing.

E. Map Processing

Map Processing block considers results returned from mapping and loop detection threads to make the decision. Table I resumes all possibilities that the system can meet. If mapping consecutively fails after a fixed number of frames due to some reasons such as abrupt movement or occlusion, our system turns into tracking-lost state (tracking lost = true). In this state, each frame is processed only by loop detection thread. Mapping thread is disabled. Once the camera is relocated in the map, we return to tracking-active state. However, map optimization will be neglected as the lack of previous poses. Moreover, map will be discrete at the relocated point and the incoming optimization is limited to this point. In a normal situation when tracking-lost is false, if mapping is invalid for current frame while loop detection provides a legal result, we buffer the loop information. In the limited following frames, in the case that mapping revives, loop closing will be performed.

"Map correction" is called only if both loop and mapping threads return valid estimations. Once it is activated, the trajectory is optimized by distributing loop closing error along pose graph. The propagation starts from loop point, follows the trajectory back to the point to which loop point is attached. Figure 2 shows the correction applied for each position in the map. Assuming that position $(i+1)$ is optimized, we compute the transformation $\check{T}_{i+1}$ between the optimized pose C^{op}_{i+1} and the non-optimized pose C_i, while T_{i+1} is the transformation between two non-optimized poses already maintained in the node $(i+1)$. T^{op}_{i+1} is then estimated by equation (4) where μ represents the "propagation coefficient". In our experiments, we propose to compute μ depending on the

TABLE I
POSSIBILITIES AND DECISION OF "MAP PROCESSING" BLOCK

Blocks									
Tracking lost	TRUE			FALSE					
Mapping	-			Valid			Invalid		
Loop detection	Key-frame	Neutral	Loop	Key-frame	Neutral	Loop	Key-frame	Neutral	Loop
Decision									
Update pose graph	-	-	+	+	+	+	-	-	-
Add key-frame	-	-	-	+	-	-	-	-	-
Activate Map Correction	-	-	-	-	-	+	-	-	-
Re-localization	-	-	+	-	-	-	-	-	-

number of optimized positions (N_p) in total by equation (5). The optimized pose of node i (C_i^{op}) is finally computed by equation (6).

$$T_{i+1}^{op} = \mu \check{T}_{i+1} + (1-\mu) T_{i+1} \tag{4}$$

$$\mu = \pi / N_p \tag{5}$$

$$C_i^{op} = C_{i+1}^{op} * T_{i+1}^{op^{-1}} \tag{6}$$

The execution time of map correction depends on the map size. Following time, this step becomes costly with a large loop closure. To warrant frame-rate processing, we launch "map correction" as a thread in parallel and continue to process next frame. However, the key-frames set is blocked in order to avoid memory accessing dump during map correction. As a consequence, any new key-frame is added and we just update pose graph until the current correction thread is finished.

IV. HARDWARE-SOFTWARE MAPPING ON A CPU-GPU ARCHITECTURE

The heterogeneous CPU-GPU architecture is considered in our system due to its popularity in embedded computing platforms. The algorithm is analyzed based on the data flow for each functional block. The evaluation methodology consists on the identification of blocks consuming a significant processing time or having a low data dependence. During the experiments, we found that Features Matching block has a high computational cost but could be parallelized thanks to the independence in data flow. In this functional block, each point correspondence in an image will be found by comparing the HOOFR 256-bits descriptor of this point to that of each point in the other image. For a high localization precision, a large number of points detected is required, leading consequently to a high matching cost. However, processing of each point is not related to others so a parallelization can be performed. Otherwise, HOOFR features extraction is also accelerated using OpenMP to exploit all the computing cores of the CPU.

A. OpenMP Implementation of HOOFR Extraction

The HOOFR detection is more suitable to implement on CPU than GPU architecture due to 2 main reasons. Firstly, HOOFR is based on FAST detection which employs a segmentation test to accelerate feature extraction processing. In the segmentation test, a pixel can be rejected after one or two pixel tests. Such a strategy makes the difference in processing cost for each pixel (some pixels require much more time to be processed than others). Hence, it is not suitable to be implemented on a GPU architecture where each work-item requires the same complexity to make use of computation resources. Secondly, the next step after FAST detection is Hessian filtering. Hessian score is computed for all the features returned by FAST detector and then only some relevant features with the highest Hessian score are kept. This filtering is much more rapid on CPU thanks to the binary classification (used in std::nth_element function of C++ stdio.h library). However, binary classification needs a dynamic memory allocation which is not supported on GPU. Hence, in our system, we employed OpenMP to implement HOOFR feature extraction.

There are two parts in the images: passive zone and active zone. As we select a pattern of surrounding points to make its description, passive zone is a part of image where the pixels are close to the border so that the description pattern is out of image. Passive zone is determined by *edge_threshold* in HOOFR descriptor and it is useless to detect key-point inside this part. Otherwise, active zone is the part where key-points could be described without doubt by HOOFR descriptor. Active zone is divided into grid. The number of cells in X and Y axes are set based on the image resolution in such a way that each dimension of one cell is about 80 – 150 pixels. The detection performing on one cell is independent to that on others cells.

HOOFR detection is demonstrated on the first part of algorithm 3. Each OpenMP thread processes an image cell and individual key-point sets are created for each cell to assure data independence. *NUM_THREADS* represents the number of cells handled in parallel. We assign value to *NUM_THREADS* by the total number of cores inside the processors to make use of computing resources. A bigger value of *NUM_THREADS* is meaningless in practice since a maximum parallelism was employed. In each cell, FAST detection is performed with adapting threshold. Then we extract the relevant key-points corresponding to the highest HESSIAN scores. At the end of detection phase, all key-points are regrouped in one global set to which a structure defined as *Points_Distribution* is attached. This structure represents the distribution of key-points in the image and is later required in Matching block to specify the searching regions. We chose the static mode for OpenMP

Algorithm 3 OpenMP Implementation of HOOFR Extraction

```
//////******Detection******//////
#pragma omp parallel for num_threads(NUM_THREADS)
For each image cell do
   keypoints_cell ← FAST_Detection(fa);
  ///***adapting detection***///
  if ( keypoints_cell.size() < pts)
     keypoints_cell ← FAST_Detection(fa/2);
  end if
  if ( keypoints_cell.size() > pts)
     Compute_Hessian_score;
     keypoints_cell ← Retain_relevant_points;
  end if
end for
//////******Key-points Regrouping******//////
  [Points_Distribution, keypoints_set] ← Regroup_keypoints;
//////******Description******//////
#pragma omp parallel for num_threads(NUM_THREADS)
For each keypoint in keypoints_set do
   Compute_keypoint_descriptor;
end for
```

scheduling instead of the dynamic mode. The reason is that computational complexity in each thread is comparable to that in another thread. Static mode is hence more suitable in which the chunks are scheduled to threads during compilation while dynamic mode is not efficient due to the more locking. Similar to the detection phase, features description is also parallelized using OpenMP but the strategy is modified. We note that the number of key-points detected in each image cell is not constant. Specially, when non-texture parts appear in the scene, some image cells contain very few key-points in comparison to other cells. If we keep the parallelism on image cell level, the threads handling many key-points will be extremely more costly than the threads with few key-points. In such case, some computing units finish the work too fast and have wasting time to wait others. To avoid this issue, we propose to use OpenMP at key-point level as shown on the second part of algorithm 3. Orientation and description for each key-point are extracted without dependence on any another key-point. The same complexities are presented for all threads leading to an efficiency in work distribution among computing units.

B. GPU Implementation of Features Matching

In this paper, our system is developed on a CPU-GPU architecture. CUDA and OpenCL are two well-known languages for GPU programming. OpenCL is supported by several high-end GPUs (NVIDIA, AMD, Intel, etc.). It is also a framework for programming across various heterogenous platforms such as: CPU-GPU, CPU-DSP or CPU-FPGA. Otherwise, CUDA is less flexible when only supported by NVIDIA hardware. However, in some powerful embedded platforms (Tegra K1, X1, X2), NVIDIA supports only CUDA programming. In order to make HOOFR SLAM work on several architectures, we developed Features Matching block in three versions: OpenCL and CUDA versions running on a GPU and a standard C++ version running on a CPU.

OpenCL divides gpu memory access into 3 principal types: *global memory*, *local memory* and *private memory*. *Global memory* is accessible by all kernels in the GPU and is the only part which has data interaction (transfer and receive) with CPU memory. Otherwise, *local memory* is accessed by the kernels in the same work-group. It is integrated on-chip and has a fast accessing time in comparison to global memory. However, local memory is limited so that we should avoid abusing it. Finally, *private memory* is the fastest memory but the smallest part seen from kernel. It belongs to a work-item and is accessed only by this work-item. In features matching of HOOFR SLAM, we benefit all kinds of GPU memory to have an optimized implementation. CUDA uses the same manner to observe GPU memory but with a little change in naming: *global memory*, *shared memory* (corresponding to *local memory* in opencl) and *local memory* (corresponding to *private memory* in opencl). The CUDA programming nature is also similar to that of OpenCL. Hence, in the following, we only detail the implementation in OpenCL while the other could be deduced easily.

To implement features matching on GPU, key-point information must be transferred to the GPU global memory. As shown in figure 3, two parameters (*cel*, *des*) are required for each key-point in PNFs. *cel* is in the form of integer number corresponding to the cell where the key-point is located. It takes the values from 0 to (n-1). Besides, *des* is 256-bit HOOFR description of the key-point. In practice, *des* is performed using a matrix having 1 row and 32 columns with 32 elements of type "unsigned char". To regroup all parameters for PNFs, we create two matrices as in equations (7, 8) where *Pnf_Cels* and *Pnf_Dess* are respectively in dimension of (*pnf_np* x 1) and (*pnf_np* x 32), *pnf_np* is the total number of key-points in PNFs.

$$Pnf\_Cels = [cel_{11}\ cel_{12}\ ...\ cel_{1m}\ .....\ cel_{i1}\ cel_{i2}\ ...\ cel_{il}]^T \quad (7)$$

$$Pnf\_Dess = [des_{11}\ des_{12}\ ...\ des_{1m}\ .....\ des_{i1}\ des_{i2}\ ...\ des_{il}]^T \quad (8)$$

On the side of current frame, two parameters are also taken into account. Firstly, we create *Cur_Dess* matrix having the dimension of (*cur_np* x 32) for key-point description. *cur_np* is the number of key-points in current frame. Similar to *Pnf_Dess*, each row of *Cur_Dess* serves as one 256-bit description based on 32 unsigned char numbers. Secondly, key-points set of current frame is organized by the order of image cell so that a structure denoted as *Points_Distribution* is employed. This structure is transformed into an integer matrix with the dimension of (*N_CELLS* x 2) while *N_CELLS* is the number of image cells. In *Points_Distribution* matrix, each row corresponds to the distribution of one cell in the whole key-points set: the first element *ref* is the position where the first key-point of the cell is located in the whole set, the second element *nb* is the number of key-points of the cell.

In practice, *Pnf_Dess*, *Pnf_Cels*, *Cur_Dess* and *Points_Distribution* matrices are transferred to *GPU_Pnf_Dess*,

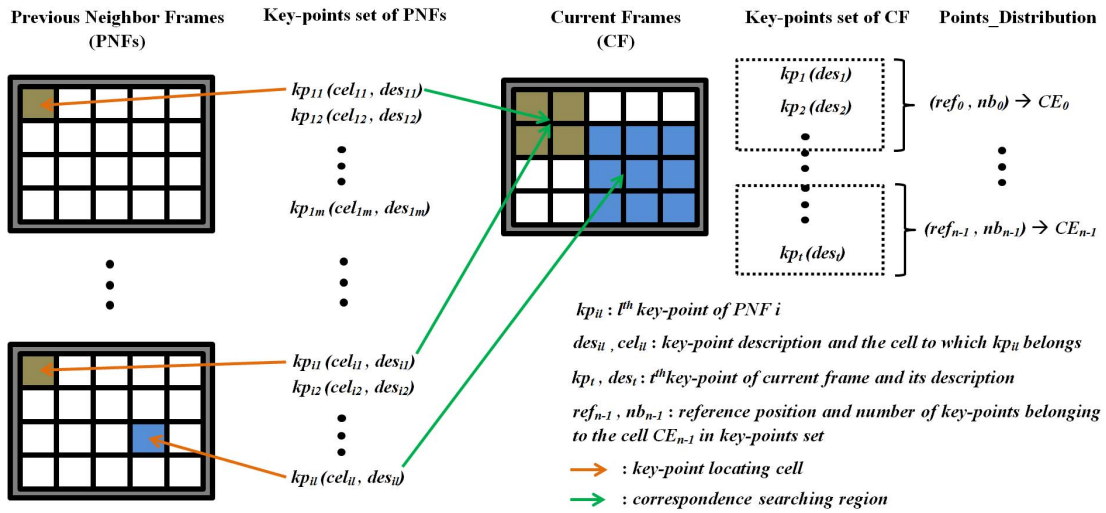


Fig. 3. Matching strategy.

GPU_Pnf_Cels, *GPU_Cur_Dess* and *GPU_Points_Distribution* respectively on GPU global memory. These memory parts are set to "read-only" to not be changed by any work-item. Moreover, we also allocate on GPU global memory a "write-only" integer matrix referred as *GPU_Correspondence* (*pnf_np* x 3) on which matching result is returned. We note that all input matrices are aligned to 1-D array on the GPU memory since GPU programming do not support pointer-to-pointer variable.

A natural implementation at our first try is that we process the whole matching of one key-point on one work-item. However, by this naive approach, we encountered the "overhead computation" problem. In fact, when kernel has too high computational cost, kernel execution takes too much time to complete one work-item. At this time, the "watch-dog" in GPU driver considers that GPU is idle since there is no feedback from kernel during an amount of time. This confusion leads to the GPU frequency reduction which severely decreases GPU timing performance. Therefore, in order to avoid such issue, we keep the kernel light by splitting the matching of one key-point into several work-items. In practice, we search the correspondence in the current frame at the same cell and neighbor cells as mentioned in figure 3. The searching on one cell is rapid due to a small number of key-points so that it is suitable to be operated on one work-item.

Algorithms 4 and 5 show the calling function on CPU and the kernel running on GPU for feature matching in mapping thread. The main idea is to use 9 work-items in a work-group to find correspondence in 9 neighbor cells of current frame. In kernel, *cell_id* variable is the index of image cell where the work-item performs the searching. *cell_id* is one neighbor cell so that it is determined by local_id *kc* of the work-item and image cell *GPU_Pnf_Cels*[i] of the PNF key-point. Keypoints locating in the image cell *cell_id* of current frame are classified from position *ref_l* to position *ref_h* in the key-points set. Besides, *dist_min* and *trainIdx* correspond respectively to the distance and the index in key-points set of

Algorithm 4 Calling Function on Host (CPU)

```
function Matching
.....
workitems = (pnf_np+BLOCK_SIZE-1)/BLOCK_SIZE*BLOCK_SIZE;
global_work_size[] = {workitems,9};
local_work_size[] = {BLOCK_SIZE,9};
clEnqueueNDRangeKernel(cmd_queue, matching_kernel,
2, NULL, global_work_size, local_work_size,
0, NULL, NULL);
clFinish(cmd_queue);
.....
end function
```

the first matching, while *dist_min2* is the distance of the second matching. Opencl local memories are allocated to save 9 local results and are synchronized by **barrier** function. After the synchronization, only one of these 9 work-items (*kc*=0) continues handling the local results to extract the final matching. It specifies final *dist_min* and *dist_min2* from local results and validates the matching if the ratio *dist_min/dist_min2* is lower than *0.85*. BLOCK_SIZE represents the number of PNF key-points processed also in the same work-group. Thus, *local_work_size* is assigned to {BLOCK_SIZE, 9}. The value of BLOCK_SIZE depends on many factors defined in GPU architecture such as the maximum *local_work_size* in each dimension or the local memory capacity. In our implementation, BLOCK_SIZE is set to 16 which provides a good performance. OpenCL programming claims that *global_work_size* must be a multiple of *local_work_size* in all dimension. Hence, the first dimension of *global_work_size* must be the nearest multiple of BLOCK_SIZE that is greater or equal to *pnf_np*. The work-items having the global identification bigger than *pnf_np* will be stopped rapidly after the test at the first line in kernel. The second dimension of *global_work_size*

Algorithm 5 OpenCL Matching Kernel on Device

declare global arrays: GPU_Pnf_Dess, GPU_Pnf_Cels, GPU_Cur_Dess, GPU_Points_Distribution, GPU_Correspondence;
function KERNEL: MATCHING
declare 3 local arrays: DIS_min[9*BLOCK_SIZE elements], DIS_min2[9*BLOCK_SIZE elements], MatchingId_min[9*BLOCK_SIZE elements];
i ← get_global_id(0);
ki ← get_local_id(0); *////from 0 to BLOCK_SIZE-1*
kc ← get_local_id(1); *////from 0 to 8*
////identify neighbor cell
cell_id ← **Get_Neighbor_Cell_ID**(GPU_Pnf_Cels[i], kc);
//// Get keypoint descriptor
point_pnf_des ←
Get_Keypoint_Descriptor(GPU_Pnf_Dess[32*i]);
////Get local matches to from the neighbor cell
{DIS_min[9*ki+kc], TRAINIdx_min[9*ki+kc], DIS_min2[9*ki+kc]} ←
Find_Local_Best_And_Second_Matches (point_pnf_des, GPU_Cur_Dess, GPU_Points_Distribution);
barrier(CLK_LOCAL_MEM_FENCE);
*///**At this point, local matching result of 9 neighbor cells are saved at the positions from 9*ki to 9*ki+8 **///*
if (kc==0)
GPU_Correspondence ←
Find_Global_Matches(DIS_min, TRAINIdx_min, DIS_min2);
end if
end function

takes the value of 9 similarly to the second dimension of *local_work_size*. GPU programming is also employed for features matching in loop detection thread and we use the same approach as in mapping thread to find correspondence. However, matching conditions have a little changes leading to some modifications in matching kernel. Firstly, due to the fact that "cross BruteForce" is used, only the last matching will be searched in each matching direction. *dist_min2* will not be considered so that we do not need to allocate GPU memory to save it. After barrier function, the process is also simpler when only the last matching is extracted from 9 local ones. Secondly, in CPU calling function, two kernel calls (**clEnqueueNDRangeKernel**) are required: one for "*current frame* to *max_lilkelihood frame*" key-points matching and the second is for the opposite direction "*max_lilkelihood frame* to *current frame*". On the other hand, BLOCK_SIZE still keeps the value of 16. *local_work_size* and *global_work_size* in each kernel call are computed by the same manner as used in mapping thread CPU call.

Figure 4 presents the CPU-GPU mapping of the algorithm. In order to avoid memory access conflict, mapping and loop detection thread work on separate zones of CPU memory. Each zone is pinned respectively to that of GPU global memory where the corresponding matching kernel is performed.

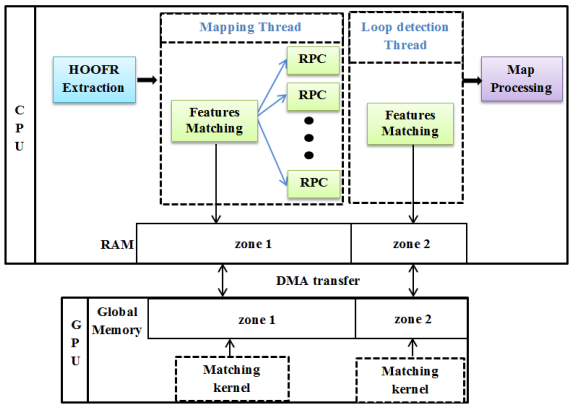


Fig. 4. CPU-GPU mapping.

TABLE II
ALGORITHM PARAMETERS

Parameter name	Value
Number of features	1500-2500
FAST threshold (fa)	12
Difference threshold in feature matching (φ)	0.85
Low threshold of pixel position change	2
High threshold of pixel position change	90
RANSAC threshold in E estimation	1.0-0.4
Inliers scale threshold (ε)	0.1
Number of binary words for loop detection (K)	150
Maximum neighbor frames (*nframes*)	4

Memory pin also allows to active DMA high-bandwidth data transfer between CPU and GPU.

V. System Implementation and Evaluation

A. Localization and Mapping results

We evaluate our proposed algorithm on five well-known datasets KITTI [33], Oxford [34], Malaga [35], MRT [36] and St_Lucia [37] with full image resolution. Table II regroups the main parameters of our algorithm used in the experiments. In order to warrant the precision, we detect 2000 features per image in KITTI, MRT, St-Lucia sequences; 2500 features in Oxford and 1500 features in Malaga sequences.

1) KITTI Dataset: The KITTI data set consists of 22 sequences from a car driven around a residential area. The car speed is up to 90 km/h, images are recorded at 10fps with a resolution of 1234x376 pixels and many moving objects exist in the scenes. Ground truth is provided in the 11 sequences (00-10) by an accurate GPS and a Velodyne laser scanner. Some sequences contain a significant loop-closure, i.e 00, 02, 05, and 07. We compare the performances with stereo ORB SLAM - one of the most robust algorithm which uses high cost bundle adjustment and contains loop closure in the state-of-the-art. We apply the algorithm on 11 first sequences, blue curves represent the ground truth provided by a precise RTK-GPS.

The entire localization of the 11 sequences are shown in figure 5 observed in 2D of X-Z axis. By reference to

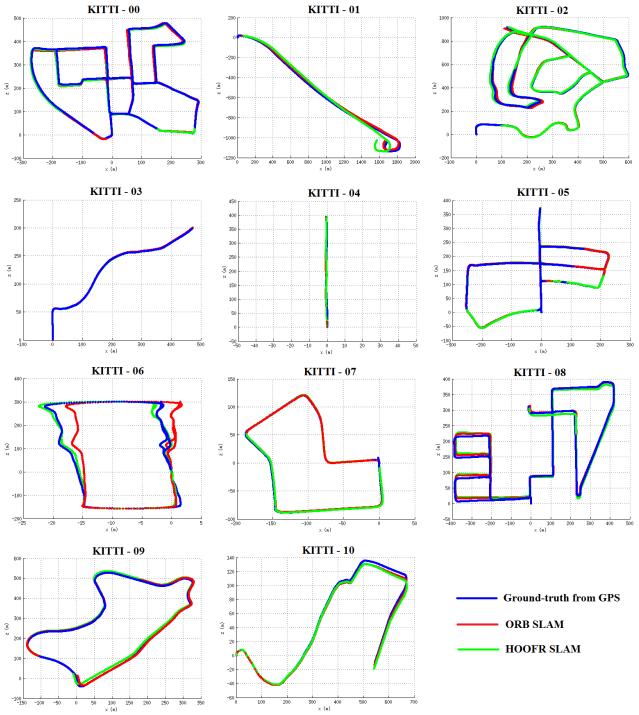


Fig. 5. Localization results of ORB SLAM and HOOFR SLAM in KITTI dataset.

TABLE III
ROOT MEAN SQUARE ERROR (RMSE) IN KITTI DATASET OF STEREO HOOFR SLAM CALCULATED FOR X AND Z AXIS

Seq	Dimension(mxm)	Frames	ORB RMSE(m)	HOOFR RMSE (m)
00	564 x 496	4541	4.7612	3.2306
01	1840 x 1140	1101	17.7170	50.2589
02	599 x 946	4661	6.6243	4.7042
03	471 x 199	801	1.2390	1.2609
04	0.5 x 394	271	0.3677	0.3225
05	479 x 426	2761	1.1884	1.3507
06	23 x 457	1101	1.6343	0.8061
07	191 x 209	1101	0.9304	0.9199
08	808 x 391	4071	4.8629	6.4138
09	465 x 568	1591	4.2835	6.7374
10	671 x 177	1201	2.7623	3.7944

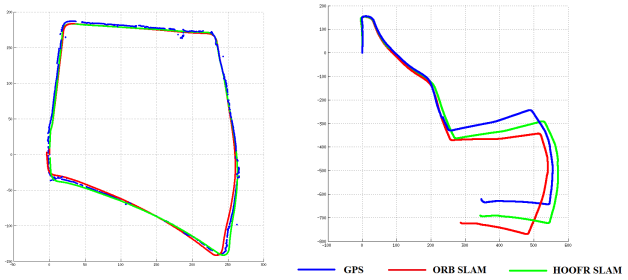


Fig. 6. Localization results of HOOFR SLAM on static (left) and dynamic (right) sequences of Oxford dataset.

camera, X is the horizontal line pointing to the right side, Y is the vertical line pointing to ground and Z is the line pointing forward. Regarding the figure, our proposal have a competitive performance with respect to ORB SLAM except the sequence 01. The reason is that this sequence is captured by a car traveling on a high way with very high velocity. ORB SLAM obtains a precise localization by saving the map-points history and using the high cost bundle adjustment optimization. In contrast, our algorithm is aimed to get high speed processing and reduce memory resources usage, so that the precision is sacrificed in this case.

Table III shows the Root Mean Square Error (RMSE) of trajectory for each sequence computed only for X and Z axis due to the fact that although GPS is corrected by RTK signal, ground-truth in Y axis is still not reliable. The results indicate that our system has a considerable accuracy with a trajectory error around 1% of its dimension (except 2% for sequence 01). The percentage is computed by the ratio of RMSE over the maximum value of 2 dimensions. Despite of the less complexity, our proposal even surpasses ORB-SLAM in some sequences such as 00, 02, 04 or 06.

2) Oxford Dataset: Oxford dataset is recorded by 6 cameras mounted onboard a vehicle traversing a route through central Oxford. The ground truth is provided by the fused GPS+Inertial solution. In order to evaluate HOOFR SLAM, we choose two sequences: the "static sequence" (recorded on 2014/05/06 at 12:54:54 GMT) contains very few moving objects and the "dynamic sequence" (recorded on 2014/06/24 at 14:47:45 GMT) is a challenging by a longer trajectory and in presence of many moving objects in the scenes. Figure 6 shows the performances of HOOFR SLAM on these two sequences.

On static environment, HOOFR SLAM and ORB SLAM present a very robust performance where RMSEs are respectively 2.24m and 2.22m. However, the localization error is increased on "dynamic sequence" where the RMSE is 40m for HOOFR SLAM and 70m for ORB SLAM. One of the most challenge of dynamic sequence is that there are some blurry images caused by sunlight. Hence, the degraded result is explained due to two factors: the moving objects and the poor image quality.

3) MALAGA Dataset: Malaga stereo dataset was gathered entirely in urban scenarios with a car equipped with a Bumblebee2 stereo camera running at a high rate (20fps). We chose the 10th sequence of dataset because it contains a very long trajectory, several loop closing and a huge variation of image brightness during the experiment. We also test localization performances of stereo HOOFR-SLAM in comparison to stereo ORB-SLAM and the result is shown in figure 7. To the best of our attempts, ORB-SLAM did not exhibit a converging trajectory. At some points when the image brightness is low, ORB-SLAM provides a poor localization or even lost tracking. Otherwise, our algorithm shows a remarkable localization result with *nframes* $= 2$ and the number of keypoints set to 1500. The argument explaining this situation is that ORB-SLAM uses ORB detector while our proposal uses HOOFR detector. Following our previous

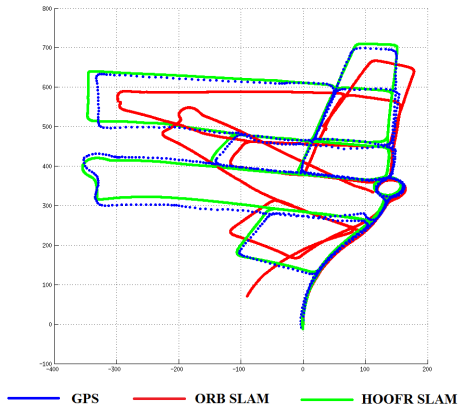


Fig. 7. Localization result using Malaga sequence: GPS (blue), ORB-SLAM (red) and HOOFR-SLAM (green).

publication [23], HOOFR detector has better repeatability than ORB detector in case of brightness change. By reference to GPS result, the reconstructed trajectory of our proposal is more reliable than that of stereo ORB-SLAM.

4) MRT and St-Lucia Datasets: In our experiments, we also validate HOOFR SLAM performances on two old datasets: MRT [36] and St-Lucia [37]. MRT is realized in 2010 using 20Hz calibrated stereo cameras. The stereo images are recorded from the AnnieWAY vehicle driven in a loop at a bridge in the city of Karlsruhe. Besides, St-Lucia dataset is gathered from 30 Hz calibrated stereo cameras embedded on a car driven on 9.5 km around the University of Queensland's St Lucia campus. The reconstruction results of HOOFR for these two datasets are shown in figure 8 including trajectory generated from GPS (no RTK correction). Comparing to GPS data, HOOFR exhibits a considerable localization result. Noting that although GPS devices used in the two datasets are not very precise, it allows us to recognize the general shape of the trajectories.

B. Timing

1) Architecture Description: In experiments, we have implemented HOOFR SLAM on two CPU-GPU platforms: a powerful Intel PC and an NVIDIA JETSON Tegra TX1 development system. Table IV shows their specifications as a recap.

Intel PC provides a mighty CPU containing 8 cores i7 running at 3.4 GHz. The CPU architecture optimizes memory access by offering 8MB smart cache that allows all cores to dynamically share access to the last level cache. The main memory (RAM) is 16 GB allowing storing a very long trajectory. This machine also integrates an NVIDIA GT-740 GPU with 384 shader cores, 2GB global memory and 28.8 GB/s memory interface bandwidth. The GPU programming supports CUDA and OpenCL.

On the other hand, since Tegra X1 (TX1) is a platform for embedded applications, its design is aimed to consume less energy. On this board, NVIDIA has elected to use ARM's Cortex containing a cluster of 4 high performance A57 big cores and a cluster of 4 high efficiency A53 little cores.

TABLE IV
ARCHITECTURE SPECIFICATIONS (JETSON TEGRA X1 EMBEDDED SYSTEM VS POWERFUL INTEL PC)

	TX1	Intel PC
CPU	4-cores ARM A57 4-cores ARM A53	8 intel cores i7
CPU clock rate	1.3-1.9 GHz	3.40 GHz
Cache	2 MB	8 MB
RAM	4 GB LPDDR4	16 GB
GPU	256-core Maxwell	384-core Geforce GT 740
GPU clock rate	1 GHz	1.07 GHz
Operating System	Ubuntu 14.04	Ubuntu 14.04
CUDA version	7.0	7.5
OpenCL version	-	1.2

TABLE V
MEAN OF EXECUTION TIME (MILLISECONDS) USING KITTI DATASET FOR EACH FUNCTIONAL BLOCK IN HOOFR SLAM ON THE INTEL POWERFUL PC AND THE TX1

	nframes = 2			
	Intel		Tegra TX1	
	CPU (8 cores)	CPU-GPU	CPU (4 cores)	CPU-GPU
HOOFR Extraction	8.536	8.555	16.783	16.731
Mapping	52.126	27.332	119.937	99.185
Loop detection average	15.001	7.881	21.916	16.466
Loop detection cost-time	36.553	15.253	95.248	80.223
Map Processing	0.349	0.137	0.584	0.403

TABLE VI
MEAN PER-FRAME EXECUTION TIME COMPARISON ON KITTI

Algorithm	Execution time (ms)	
	Intel PC	Tegra TX1
Stereo ORB SLAM	69.924	190.710
CPU - HOOFR SLAM	62.235	137.235
GPU - HOOFR SLAM	36.154	116.552

However, only one cluster could be activated at a time. The A57 CPU cluster operates at 1.9 GHz, has 2MB of L2 cache shared by the four cores with 48KB L1 instruction cache and 32KB L1 data cache per core. The A53 CPU cluster operates at 1.3 GHz, with 512KB of L2 cache shared by four cores, 32KB instruction and also 32 KB data L1 cache per core. The GPU of TX1 is designed using Maxwell architecture, includes 256 shader cores and clocks at up to 1GHz. GPU memory interface offers a maximum bandwidth of 25.6 GB/s with the capacity of 2GB global memory. NVIDIA supports only CUDA for gpu programming on TX1 (no OpenCL), so that we use CUDA version of HOOFR SLAM matching block during the experiments on this board.

2) Timing Evaluation : We evaluate the mean processing times of the proposed algorithm on 11-first sequences of KITTI dataset. All timings are given in milliseconds. The values are the mean of 11 sequences where timing on each

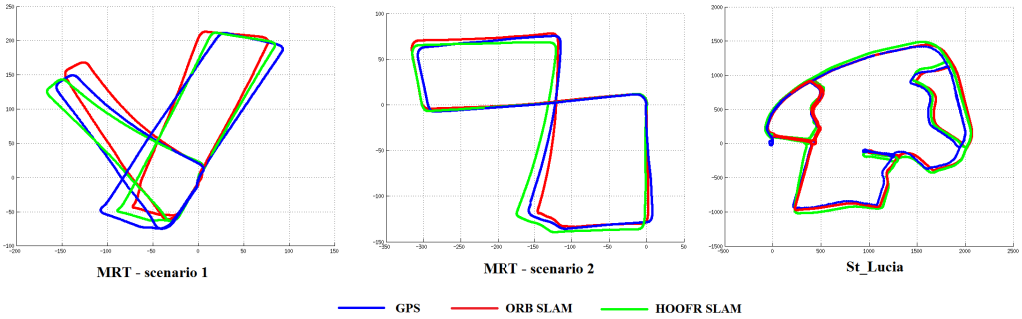


Fig. 8. HOOFR SLAM reconstruction using MRT and St-Lucia datasets.

TABLE VII
KITTI-07 PROCESSING TIME (ms) ON INTEL PC AND NVIDIA TX1 WITH DIFFERENT VALUES OF *nframes*

$nframes$	1			2			3			4		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
Intel (CPU)	33.646	58.254	78.156	36.512	65.689	82.241	38.989	76.263	96.358	43.989	81.124	98.416
Intel (CPU-GPU)	18.154	36.487	50.164	19.498	39.456	57.129	20.846	41.354	61.487	22.498	50.462	64.624
TX1	40.268	101.265	130.748	46.894	130.128	170.854	48.657	152.238	201.418	50.658	162.624	240.859

sequence is also the mean after 5 lauches. Table V represents the timing of each functional block in our proposal pipeline. The number of neighbor frames is 2. In the table V, Loop detection average is the sum of execution time divided by the total number of frames. However, this value can not be a good representation because execution time of Loop detection thread is not constant. In fact, with an overlapped frame or in case of not enough inliers, loop detection thread is terminated rapidly. Otherwise, when loop closure is reached, this thread becomes higher cost because the relative movement is estimated. To have a better representation, we presents "Loop detection cost-time" which is the mean time of loop detection thread when the movement estimation is performed.

We notice that Mapping thread and Loop detection thread are launched in parallel. Hence, per-frame time is only the sum of HOOFR extraction and Mapping (the most consuming thread). Moreover, when loop closure is valid, map correction inside Map Processing is launched in other thread so that it does not slow down the new frame acquisition. On PC Intel, without GPU implementation, the algorithm runs at ∼62 ms per frame. By offloading processing to GPU, we have a better performance when the mean of execution time of the whole algorithm is decreased to 36 ms per frame.

For Tegra TX1 embedded system, it is obvious that the processing task is much slower than that of the Intel PC because of many reasons: lower frequency of CPU and GPU, smaller cache memory resources and low number of CPU and GPU cores. On this platform, our partitioning exhibits a considerable performance where the algorithm takes in average ∼116 ms per frame. We also evaluated the timing performance of ORB-SLAM and table VI shows the mean per-frame timing comparison between our proposal (HOOFR SLAM) and ORB SLAM on two platforms using KITTI dataset. With Intel PC, the ORB execution time is approximately 69ms per frame (7 ms costly than CPU-only version or 32ms costly than

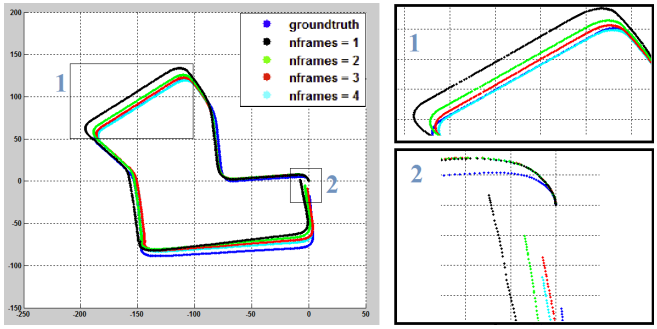


Fig. 9. KITTI-07 localization results using different values of *nframes*.

CPU-GPU version of our algorithm). On TX1 embedded platform, ORB-SLAM takes 190 ms per frame (53 ms costly than CPU version or 74 ms costly than GPU version of our proposal). Compared to ORB performance, our HOOFR algorithm exhibits a lighter processing task while maintaining a competitive accuracy. It will be an advantage when camera frequency becomes higher because HOOFR can handle more frames in this case. For fast camera moving, handling more frames allows avoiding severe change in camera position of each input frame.

To evaluate the timing in more details, we studied the timing and localization precision in terms of the number of neighbor frames (*nframes*). Table VII presents the minimum, the maximum and the mean of per-frame processing time on KITTI-07 when the *nframes* parameter changes from 1 to 4. For the powerful Intel PC, we still have a frame-rate running at less than 100ms when the *nframes* increases to 4 for both GPU and without GPU version. For TX1 embedded system, due to limited resources, processing time did not meet the frame-rate (10 Hz) performances. The variation of time in each frame is primarily as a consequence of the motion estimation step. In fact, in order to compute essential matrix from matching

set, this step uses RANSAC scheme which selects the subset by random choices and the proportion of inliers is not identical for different matching sets. Some of high proportion of inliers normally take less time to compute than that of low proportion. Besides, Figure 9 shows the effect of *nframes* on the localization result. Ground-truth is always presented by the blue curve. We notice that more we take into account the number of neighbor frames, more we get a higher localization precision. The explanation for this exhibition could be found at the features detection level. In fact, at some points in the trajectory, especially in turning scenarios, current frame contains less common points with nearest neighbor frame than with a further neighbor frame. Therefore, the motion estimation with further neighbor frame provides more confidence and has a higher weight. By integrating a more precise prediction in optimal pose extraction, we have a lower localization error.

VI. Conclusions

In this work, a novel estimation algorithm for feature-based stereo VSLAM has been presented. Our parallelized lightweight VSLAM framework was obtained as a result of a hardware-software mapping study addressing feature extraction, data processing, hardware building implementation and benchmarking. We referred to this approach as HOOFR SLAM since it integrates our previous work on HOOFR features [23]. This binary descriptor is employed for motion estimation and loop closure detection. Motion estimates are integrated over time following a hybrid filtering/key frame strategy. That is, position is estimated using a widowed weighted mean using previous neighbor frames. Weights are computed from inter-frame robust feature matching. A thorough experimental research was carried out on five well-known datasets (KITTI, Oxford, Malaga, MRT and St-Lucia) providing considerable localization results in terms of RMSE (around 1% of sequence dimension). Our real-time algorithm implementation on high performance Intel-based PC architecture processes frames at more than 20 Hz using KITTI dataset. On the Tegra TX1 embedded system, the processing time is close to real time performances with 6 fps running rate. The emergence of new heterogeneous CPU-GPU architectures such as Xavier Nvidia (8 Core ARM64 CPU, 512 Core Volta GPU) should help to embed the HOOFR SLAM algorithm with real-time constraints.

However, real-time execution of HOOFR SLAM on an embedded system that consumes few watts remains a perspective of this work. This can be achieved taking advantage of hardware accelerators such FPGAs. An important effort will be done for offloading HOOFR extraction and features matching steps on FPGA following a hardware-software co-design approach so as to achieve a real-time HOOFR SLAM System.

References

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[2] L. Riazuelo, J. Civera, and J. M. M. M. Montiel, "C^2 TAM: A Cloud framework for cooperative tracking and mapping," *Robot. Auton. Syst.*, vol. 62, no. 4, pp. 401–413, 2014.

[3] G. Bresson, T. Féraud, R. Aufrère, P. Checchin, and R. Chapuis, "Real-time monocular SLAM with low memory requirements," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1827–1839, Aug. 2015.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[5] G. Pascoe, W. Maddern, M. Tanner, P. Piniés, and P. Newman, "NID-SLAM: Robust monocular slam using normalised information distance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1435–1444.

[6] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 15–22.

[7] A. J. Davison and D. W. Murray, "Mobile robot localisation using active vision," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 1998, pp. 809–825.

[8] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 865–880, Jul. 2002.

[9] A. J. Davison and N. Kita, "3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Dec. 2001, p. I.

[10] L. M. Paz, P. Jensfelt, J. D. Tardos, and J. Neira, "EKF SLAM updates in O(n) with divide and conquer SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 1657–1663.

[11] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *J. Mach. Learn. Res.*, vol. 4, no. 3, pp. 380–407, 2004.

[12] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 428–439, Sep. 2009.

[13] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Feb. 2010.

[14] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Editors choice article: Visual SLAM: Why filter?" *Image Vis. Comput.*, vol. 30, no. 2, pp. 65–77, 2012.

[15] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1066–1077, Oct. 2008.

[16] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A system for large-scale mapping in constant-time using stereo," *Int. J. Comput. Vis.*, vol. 94, no. 2, pp. 198–214, 2011.

[17] T. Pire, T. Fischer, J. Civera, P. De Cristóforis, and J. J. Berlles, "Stereo parallel tracking and mapping for robot localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2015, pp. 1373–1378.

[18] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2015, pp. 1935–1942.

[19] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3903–3911.

[20] D. Rodriguez-Losada, P. S. Segundo, M. Hernando, P. de la Puente, and A. Valero-Gomez, "GPU-mapping: Robotic map building with graphical multiprocessors," *IEEE Robot. Autom. Mag.*, vol. 20, no. 2, pp. 40–51, Jun. 2013.

[21] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *Proc. Robot., Sci. Syst.*, 2015, pp. 1–9

[22] B. Vincke, A. Elouardi, and A. Lambert, "Real time simultaneous localization and mapping: Towards low-cost multiprocessor embedded systems," *EURASIP J. Embedded Syst.*, vol. 2012, p. 5, Dec. 2012.

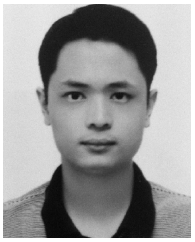
[23] D.-D. Nguyen, A. E. Ouardi, E. Aldea, and S. Bouaziz, "HOOFR: An enhanced bio-inspired feature extractor," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 2977–2982.

[24] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[25] E. Michaelsen, W. V. Hansen, M. Kirchhof, J. Meidow, and U. Stilla, "Estimating the essential matrix: GOODSAC versus RANSAC," in *Proc. Symp. ISPRS Commission III Photogramm. Comput. Vis. (PCV)*, Bonn, Germany, Sep. 2006.

[26] D.-D. Nguyen, A. E. Ouardi, and S. Bouaziz, "Enhanced bio-inspired feature extraction for embedded application," in *Proc. 14th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2016, pp. 1–6.

[27] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. 13th Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, Sep. 2014, pp. 834–849.
[28] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.
[29] H. Lim, J. Lim, and H. J. Kim, "Real-time 6-DOF monocular visual SLAM in a large-scale environment," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 1532–1539.
[30] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, Jun. 2004.
[31] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 9, pp. 920–932, Sep. 1994.
[32] M. Cummins and P. Newman, "Highly scalable appearance-only SLAM—FAB-MAP 2.0," in *Proc. Robot., Sci. Syst.*, vol. 5, Seattle, CA, USA, 2009, pp. 1–8.
[33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
[34] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The oxford robotcar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.
[35] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez, "The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 207–214, 2014.
[36] F. Moosmann and C. Stiller, "Velodyne SLAM," in *Proc. IEEE Intell. Vehicles Symp.*, Baden-Baden, Germany, Jun. 2011, pp. 393–398.
[37] M. Warren, D. McKinnon, H. He, and B. Upcroft, "Unaided stereo vision based pose estimation," in *Proc. Australas. Conf. Robot. Automat.*, G. Wyeth and B. Upcroft, Ed. Brisbane, QLD, Australia: Australian Robotics and Automation Association, 2010, pp. 1–9.



Dai-Duong Nguyen received the degree in electrical engineering in 2014 (mention in industrial information) from the Hanoi University of Science and Technology, Hanoi, Vietnam, and the M.S. degree in information, system and technology from Paris-Sud University, Orsay, in 2015. He is currently pursuing the Ph.D. degree with MOSS Group, SATIE laboratory, Paris-Sud University. His research activities are focused on vision systems for SLAM applications.

Abdelhafid Elouardi received the M.S. degree from Pierre & Marie Curie University in 2001, and the Ph.D. degree in electronics from Paris-Sud University, France, in 2005. He was with Henri Poincaré University, Nancy, as a Researcher, from 2005 to 2006. He is currently an Associate Professor with Paris-Sud University. With the Embedded Systems Team of SATIE lab, his research interests include hardware-software co-design, evaluation and instrumentation of embedded systems, design of smart architectures for image and signal processing, and SLAM applications.

Sergio A. Rodriguez Florez received the M.S. and Ph.D. degrees from the University of Technology of Compiègne, France, in 2007 and 2009, respectively. Since 2011, he is an Associate Professor with Paris-Sud University. His research activities are focused on dynamic scene analysis through multi-modal perception intended to enhance intelligent transportation systems applications.

Samir Bouaziz received the Ph.D. degree in electronics from Paris Sud University, Orsay, France, 1992. He is currently a Full Professor with Paris-Sud University. His research focuses on hardware-software designs of embedded systems for autonomous vehicles and robots. His research is led by time constraints and complexity consideration for a good fit between hardware and algorithms, instrumentation and benchmark systems to understand human-vehicle interactions.

2014 13th International Conference on Control, Automation, Robotics & Vision
Marina Bay Sands, Singapore, 10-12th December 2014 (ICARCV 2014)

Multiple Obstacle Detection and Tracking using Stereo Vision: Application and Analysis

Bihao Wang
CNRS Heudiasyc UMR 7253
Université de Technologie de Compiègne
Compiègne, France
Email: bihao.wang@hds.utc.fr

Sergio Alberto Rodríguez Florez
CNRS IEF UMR 8622
Université Paris-Sud
Paris, France
Email: sergio.rodriguez@u-psud.fr

Vincent Frémont
CNRS Heudiasyc UMR 7253
Université de Technologie de Compiègne
Compiègne, France
Email: vincent.fremont@hds.utc.fr

***Abstract*—Vision systems provide a large functional spectrum for perception applications and, in recent years, they have demonstrated to be essential in the development of Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicles. In this context, this paper presents an on-road objects detection approach improved by our previous work in defining the traffic area and new strategy in obstacle extraction from U-disparity.Then, a modified particle filtering is proposed for multiple object tracking . The perception strategy of the proposed vision-only detection system is structured as follows : First, a method based on illuminant invariant image is employed at an early stage for free road space detection. A convex hull is then constructed to generate a region of interest (ROI) which includes the main traffic road area. Based on this ROI, an U-disparity map is built to characterize on-road obstacles. In this approach, connected regions extraction is applied for obstacles detection instead of standard Hough Transform. Finally, a modified particle filter framework is employed for multiple targets tracking based on the former detection results. Besides, multiple cues, such as obstacle's size verification and combination of redundant detections, are embedded in the system to improve its accuracy. Our experimental findings demonstrates that the system is effective and reliable when applied on different traffic video sequences from a public database.**

***Index Terms*—Stereo Vision, On-road Obstacles Detection, Particle Filter, Visual Tracking.**

I. Introduction

Obstacle detection and tracking are key issues of the Advanced driver assistance systems (ADAS). In the context of driver assistance, the purpose of obstacle detection and tracking system is to detect and monitor the dynamic behavior of one or more obstacles in the vicinity of the host vehicle. Hence, the ADAS can help to avoid potential collisions and to provide essential information for decision making. A reliable detection and tracking approach in real environments has been a challenge in the last two decades, especially considering various type of objects, their time varying number, and their states estimation from noisy observations at discrete intervals of time. Sensors, such as radar and LIDAR, have been used for this purpose using sensor data fusion approaches [1], [2]. In recent years, many vision-only based approaches have been developed [3], [4] for the versatility of information they can provide and their low cost. In these approaches, learning-based methods focus on the detection and tracking of specific obstacles: like pedestrians and vehicles [5], [6]; while motion-based methods can extract the moving objects [4], [7]. In this paper, we present an on-road object detection approach from our previous work [8], [9] which can effectively detect the road area in traffic scene and a multi-object tracking strategy based on particle filter. Since U-V-disparity map [10] is an effective method to detect objects regardless their appearance and motion model, we propose to use it in the obstacle detection part. In order to predict the obstacle's position and moving direction, tracking is added in the system as a complementary to object detection [5], [11]. Especially, particle filters [11], [12] are widely used to solve multiple time varying obstacles tracking problems. Their strength lies in their ability to represent non-Gaussian distributions which can capture and maintain target properties.

First, stereo-vision-based obstacle detection is applied on a region of interest (ROI), which is composed of the main traffic area. To obtain the ROI, a fast road surface detection method [8] is firstly applied. Then, a convex hull algorithm is introduced to achieve the complete traffic area. Within the ROI, a method of connected region extraction from U-disparity map is developed to locate the obstacles in the image, and furthermore, to refine their position information from the sub-region of disparity map extracted from the primary location of the obstacles. To improve the detection accuracy, multiple cues are integrated in the system, such as an adaptive height gating for obstacle detection in different distances and a combination of close detected area in the U-disparity map.

After obtaining the position, shape and depth information of the on-road obstacles, multiple target tracking hypotheses are managed by the means of a bank of particle filters. Target-to-track association is carried out following a global nearest neighbor (GNN) criterion. The tracking is performed in the image plane of the left camera in stereo vision system. In the 2D image plan, the obstacle's position and size is effected by it's distance to camera. To cope with this factor, the observed dynamics of the tracked obstacles is employed to define an adaptive association gate. Besides, a dynamic noise generation function is implemented in the filter. Considering the number of obstacles is time varying, for each iteration of the filtering, multiple hypotheses are made to create, delete and update the existing tracks. Fig. 1 shows the outline of the proposed perception system.

The approach is structured into two parts: on-road ob-

978-1-4799-5199-4/14/$31.00 ©2014 IEEE

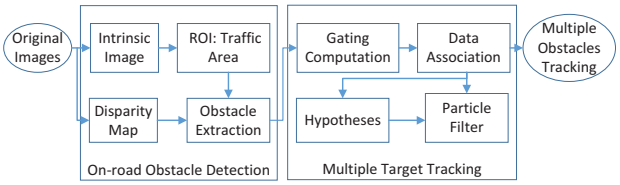


Figure 1: System Outline

stacles detection based on connected region in U-disparity map; and modified particle filter tracking of multiple targets. The strengths of this approach are: (1) It proposes a reliable detection and tracking system that can be directly applied in different driving scenarios. (2) It is capable of detecting all the on-road obstacles with efficiency and accuracy, regardless of their shapes and poses. (3) It presents a modified particle filter for visual tracking, which has a great tolerance for the dynamics of obstacles in image plan caused by depth factor.

The paper is organized as follows: First, a stereo-vison based on-road obstacle detection is introduced in Section II which includes two parts, traffic area extraction (Section II-A) and obstacle detection (Section II-B). Then, a modified particle filter tracking of multiple targets is presented in Section III. Experimental results and analysis on publicly available dataset [13] are shown in Section IV. Finally our paper ends with conclusions and future work.

II. On-road Obstacles Detection

The disparity map, $I\Delta$, can be extracted from stereo images[14]. It refers to the displacement of the relative features or pixels between two views. A bigger disparity value corresponds to a closer distance to the camera. In $I\Delta$, an obstacle is represented as a homogeneous part with the same disparity value. U-V disparity maps are built by accumulating the pixels with same disparity value along the u, v axis of $I\Delta$ separately. The V-disparity map, $I_v\Delta$, is usually employed to estimate the longitudinal profile of the road and to detect the presence of obstacles by the means of a line extraction algorithm. However, in complex scenario, the V-disparity map $I_v\Delta$ is ambiguity prone. When obstacles are closed to each other, their representative lines in $I_v\Delta$ are mixed together. On the contrary, the U-disparity map preserves more information of the scene: the objects width, their relative positions and their depth information are kept. Therefore, in this paper, the U-disparity map leads to an accurate obstacles detection, while the V-disparity only assist in this procedure.

A. Traffic Area Extraction

In order to reduce the computational cost and to improve the detection efficiency on the road area, a suitable road ROI needs to be defined. It removes off-road information which may interfere the precision of on-road obstacle detection, for instance, continuous high walls/buildings along the road. From this consideration, a free road surface detection combined with a convex hull construction is proposed hereafter.

1) Application of Convex hull : According to [8], the free road surface I_R can be computed by the conjunction of intrinsic road surface and the ground plane. Convex hull algorithm here provides the smallest convex area which contains all the free road surface I_R. It fix up the holes and the depressions caused by on-road obstacles. A complete traffic area, i.e. the ROI, is then generated from free road surface. Therefore, obstacle detection can be focused on this approximated road traffic area. Even if the convex hull may not exactly follows the shape of the road, in most of the cases, it is sufficient to provide a satisfying ROI for further detection. For other cases, the tracking process detailed in Section III will efficiently deal with this issue.

B. U-disparity map based obstacles detection

In traffic scenes, there exist pedestrians, vehicles, traffic lights and signs, etc. For that purpose, the U-disparity map can be used to handle all the on-road obstacles, without prior knowledge on their types and motion models.

1) Connected-region extraction: In the U-disparity map $I_u\Delta$, obstacles are usually represented as straight lines. However, when an obstacle is passing near the camera side, both the frontage and side face of the obstacle are observed. The obstacle is then represented by a polyline: an horizontal part for frontage and a connected oblique part for its side face (As shown in Fig. 2 for the second obstacle in the second column). This situation happens frequently in driving scenes. To cope with this problem and to simplify the detection processing, a connected-component extraction algorithm is introduced in this paper to replace the classical Hough line extraction.

After a preprocessing using Eq. (1), high intensity regions are preserved in the U-disparity map $I_u\Delta$ and the other pixels are set to 0 (i.e. background).

$$I_u\Delta = \mathrm{sgn}(I_u\Delta(p) - \varepsilon) \tag{1}$$

The definition of the intensity threshold, ε, is related to the camera calibration parameters and object's depth information. In our experiment, it has been set to 8 to 10 accumulated pixels.

After applying morphological operations (here, erosion and clean), noisy pixels are removed from $I_u\Delta$. Each connected-region L being preserved in $I_u\Delta$, indicates a potential obstacle O_L. Thus, the passing-by obstacle's information can be obtained by connected-component extraction algorithm. These information include: left bound u_l and right bound u_r of O_L on the u-axis of the image; and its disparity value d_O. The complementary information about O_L like the height h_O and the bottom position on the v-axis v_b can be extracted from $I_v\Delta$ and furthermore refined by sub-region of the disparity map which contains the obstacle.

2) Obstacle localization with sub-Disparity map : For the obstacles standing at the same distance to camera, their accurate height information is mixed in $I_v\Delta$. In order to refine the location of each potential obstacle O_L, a sub-region of disparity map $I_O\Delta$ for each obstacle is extracted from the complete disparity map $I\Delta$ according to their primarily

Algorithm 1 On-road Obstacle Detection Algorithm

Input: - Stereo color images $I_l,\ I_r$
Output: Number of detected obstacles N_{obs}, and their location information $O_{1...N_{obs}}$

for k =first frame **do** last frame ▷ Evolution of frames
  ▶ Disparity map $I_\Delta \leftarrow (I_l,\ I_r)$ and free road surface I_R;
  ▶ Convex hull construction: $I_{ROI} \leftarrow I_R$;
  ▶ U-V-disparity map on ROI: $[I_u\Delta, I_v\Delta] \leftarrow (I_{ROI},\ I_\Delta)$;
  ▶ Label the connected-regions $L_{1,...,N}$ in $I_u\Delta$:
  for $i = 1$ **do** N ▷ Location extraction
    ▶ Extract primary position and disparity value $[u_l, u_r, v_b, h_L, d_L] \leftarrow (L_i, I_u\Delta, I_v\Delta)$
    ▶ Generate sub-Disparity map for each object O_i: $I_O\Delta \leftarrow (u_l, u_r, v_b, h_L)$
    ▶ Extract obstacle by (2): $I_O \leftarrow I_O\Delta$
    ▶ Refine obstacle position $[x_O, y_O, w_O, h_O, d_O] \leftarrow I_O$,
    if $h_O \geq \delta(d_O)$ **then** ▷ Eliminate false alarm
      ▶ $N_{obs} \leftarrow N_{obs} + 1$;
      ▶ $O_{N_{obs}} = [x_O, y_O, w_O, h_O, d_O]$
    end if
  end for
end for

estimated location in the image. Pixels in these sub-regions of disparity map are classified into two classes, object or background, according to their disparity value.

$$\begin{cases} I_O = 1 \text{ object}, & \text{if } I_O\Delta(p) \in [d_1, d_2] \\ I_O = 0 \text{ background}, & \text{otherwise} \end{cases} \tag{2}$$

where, I_O is the binary image with labeled obstacles. $d_{1,2} = d_O \pm \sigma$ where, σ is the bias of possible disparity value of the same obstacle. Obstacle's position and size information is then refined in I_O. In this approach, the obstacle's information is represented by its centroid (x_O, y_O), its width w_O, its height h_O and its disparity d_O. Compare to region growing algorithms, the sub-disparity map extraction is much faster and effective. A detection example is showed in Fig. 2.

3) Multiple cues integration: In some cases, the connected component extraction may lead to false alarms or multiple detections on a single obstacle. Thus, multiple cues can be combined to refine the detection result.

- Height limitation of potential obstacles: If the object's height is smaller than a threshold δ, it will not be considered as an obstacle. This threshold is proportional to the disparity value of this potential obstacle $\delta \propto d_O$. The closer O_L stands to the camera, the higher δ will be. Thus, the system can eliminate part of the false alarms. As in Fig. 2, the yellow lines in left middle image are the false alarms detected by $I_u\Delta$.
- Combination of closely stand connected-regions: Since U-disparity image is accumulated on discrete values from the disparity map, there could exist fragments of the same obstacle using the representation of connected-region. This will lead to redundant detections. To handle this problem, a combination operations, like bridge and dilation morphological operations are introduced.

The complete on-road obstacle detection pipeline is summarized in Algorithm 1.

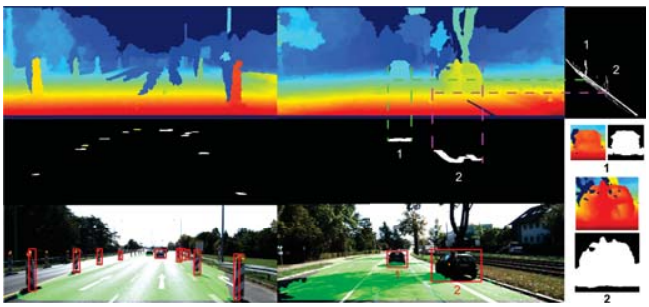


Figure 2: Procedure of on-road obstacle detection. For first two columns, from top to bottom: disparity map; U-disparity map within ROI (convex hull of the green area in detection results); detection result. Top-right is V-disparity map, examples of sub-regions of Disparity map for obstacle extraction are shown under it.

III. Multiple Obstacles Tracking

When multiple obstacles have been detected and they are tracked over frames, several unknowns must be properly handled, such as the number of targets at each time step and the data association between targets and tracks. On-road obstacles are regarded as the targets, see Fig. 1. Hereafter, a modified particle filter based on the condensation algorithm, is proposed to track every target. A dynamic noise update function is introduced and a self-adaptive gating is also designed to provide a reliable data association result.

A. Particle filter model

The condensation algorithm, as a special case of particle filtering[15], provides a well-established methodology for generating samples from the required distribution without requiring assumptions about the state-space model or the state distributions. The samples from the distribution are represented by a set of particles; each particle has a weight representing the probability of that particle being sampled from the probability density function. At each time step, particles' weight and spatial distribution are used for tracker state estimation and re-sampling. In this paper, the tracking of the surrounding obstacles is carried in 2D image plane. In order to define an uniform state space, all the states are described in pixel level. The depth information is then represented by its corresponding disparity value. The filter model is build as follows:

- State vector:

$$S = [x, y, v_x, v_y, w, h, d]^T \tag{3}$$

It is composed by the centroid object position (x, y) on the image; the velocity of the centroid v_x, v_y ; the width w; the height h and the disparity value d respectively.

- Observation:

$$Z = [x_O, y_O, 0, 0, w_O, h_O, d_O]^T \tag{4}$$

The values x_O, y_O, w_O, h_O, d_O represent the information of detection result of Algorithm 1. Since the velocity of the object cannot be measured directly, it has been set to a 0 value.

- Estimation:

$$S(k) = \sum_{i=1}^{N_s} s^i(k) \cdot \pi^i(k) \tag{5}$$

where, S is the condensation state of the tracker, N_s is the number of samples. $s^i(k)$ is current sample states, where, $i = 1, \ldots, N_s$, and k represents the time step. The variable π^i is the normalized weight distributed for each sample.

- Prediction:

with
$$s^i(k+1) = f(s^i(k)) \tag{6}$$

$$s^i(k) = [x^i(k), y^i(k), v_x^i(k), v_y^i(k), w^i(k), h^i(k), d^i(k)] \tag{7}$$

$s^i(k)$ and $s^i(k+1)$ are consecutive particle states of current time and next time respectively. f is the dynamic model for evolution. In this approach, f is a constant velocity model. Thus, (6) can be written as :

$$\begin{cases} x(k+1) = x(k) + T \cdot v_x(k) + W_x(k) \\ y(k+1) = y(k) + T \cdot v_y(k) + W_y(k) \\ v_x(k+1) = v_x(k) + W_{v_x}(k) \\ v_y(k+1) = v_y(k) + W_{v_y}(k) \\ w(k+1) = w(k) + W_w(k) \\ h(k+1) = h(k) + W_h(k) \\ d(k+1) = d(k) + W_d(k) \end{cases} \tag{8}$$

$$W(k) = [W_x(k), W_y(k), W_{v_x}(k), W_{v_y}(k), W_w(k), W_h(k), W_d(k)]^T \tag{9}$$

where, $W(k)$ is the noise vector at time step k that added for filter evolution.

- Update and Re-sampling:

$$\pi^i(k+1) = \frac{P(s^i(k+1 \mid k) \mid Z(k+1))}{\sum_{i=1}^{N} P(s^i(k+1 \mid k) \mid Z(k+1))} \tag{10}$$

For each evolution, particles $s^i(k+1 \mid k)$ are predicted from their previous state $s^i(k)$ by Eq. (7). A new confidence density $P(s^i(k) \mid Z(k))$ is then distributed to $s^i(k+1 \mid k)$ through the comparison between particle states and associated observation $Z(k+1)$. The weights of samples are then updated from the confidence density by Eq. (10). Subsequently, a new set of N_s particles $s^i(k+1)$ are constituted from the current sample set $\{s^i(k+1|k)\}$ with probability proportional to the confidence distribution [15].

B. Dynamic noise update

In the image coordinate frame, the scale and the displacement of an obstacle change rapidly with its distance to the camera. To handle this issue, the noise vector added in Eq. (8) should follow this change as well, i.e. $W(k) \propto d_O(k)$. Therefore, the tracker is able to keep up with observation state in 2D image coordinates. At every sampling time, a dynamic noise update function is build as follows:

$$W(k) = C \cdot d_O(k) \cdot Z(k) \tag{11}$$

where, C is the coefficient vector that needs to be adjusted given an application. For a given equipment, these parameters can be set for once, because the variations lead by depth are related to the camera's essential matrix [16]. The noise vector $W(0)$ is initialized with high values to provide a broader range of sample distribution. During the filtering, the noise vector $W(k)$ is set with lower order of values to provide a convergent range for particle predictions.

C. Data association

Global nearest neighbor (GNN) is the most natural data association process with a low complexity. When obstacle observations are perceived from the camera, Mahalanobis distances between each observation and prediction are calculated. In our approach, the distance between object and track is defined as follows:

$$dist = c_1.\Delta_x^T \Delta_y + c_2.\Delta_w^T \Delta_h + c_3 \mid \Delta_d \mid \tag{12}$$

where, $c_{1,2,3}$ are the coefficients for different measurements which indicate their contributions to the distance calculation. Δ is the difference between estimation and observation measured on the state vector $[x, y, w, h, d]$ separately. Thus, the obstacle's centroid (x, y) in 2D image is not the only criterion that contributes to the distance calculation, but also the width, the height (w, h) and the disparity value d are considered as well.

1) Self-adaptive gate : Target-to-Track association is limited by the use of a gate which is set to a constant value for eliminating unlikely association. In 2D image plane, the closer the obstacle stands to the camera, the greater Mahalanobis distance it might have with respect to the tracks in 2D image coordinates. For the on-road obstacles which are far from camera, they have smaller scales and stand closer to each other; a big gate will lead to mismatching. On the contrary, for nearby obstacles, they may not be able to be associated with the proper tracks because of a small gate. Thus, a constant gate is not sufficient for all the obstacles standing in different distances. In this paper, a self adaptive gate is modified with respect to every observed obstacle according to their scale and depth information:

$$G_O = a \cdot w_O(k)^T h_O(k) + b \mid d_O(k) \mid \tag{13}$$

where, G_O is the gate for each obstacle according to their observation $[x_O, y_O, 0, 0, w_O, h_O, d_O]^T$ at time k. Here, we set $a = 0.5,\ b = 0.2$, which is basically the radius of the circumscribed circle plus a small percentage of the disparity value. Thus, the gate is only related to the current observation's scale and depth information. This design greatly improved the reliability of data association algorithm.

2) Multiple hypotheses: Obstacles and tracks are associated by global minimal distance within the gate. If there is no association established. This leads to two possible situations: non-associated obstacle or non-associated track. In the first case, it is assumed that a new obstacle is just detected, and a new track needs to be created for this obstacle. In the second case, non-associated track will be preserved and updated for a short time period unless the tracking failed up to a threshold. In that case, the track would be pruned.

IV. Experimental Results

The proposed algorithm has been evaluated on different sequences of KITTI dataset [13]. There are different types of road. Tracklet labels of the dataset are used as "ground truth" for a comparison and evaluation of this work.

- Dataset 1: urban road.

Algorithm 2 Modified Particle Filter Algorithm

► Initialization: Set $k = 0$, generate a sample set $\{s^i(t,k)\}$ for each detected target/obstacle at current time k, where, $i = 1,\ldots,N_s$, $t = 1,\ldots,N_{obs}(k)$. $N_{obs}(k)$ is the number of detected obstacles at time k. Particle $s^i(t,k)$ is draw from Gaussian distribution around $Z(t,k)$
for $k = 1$ **do** last frame ▷ Evolution of frames
  for $t = 1$ **do** $N_{obs}(k)$ ▷ Tracking of each obstacle
    ► Compute adaptive gate $G_O(t,k)$ for data association by (13) for each obstacle
    ► Associate tracker with obstacle by GNN algorithm
    ► Update the weight of particles $\pi^i(t,k)$ by (10)
    ► Re-sampling of particles $s^i(t,k)$ from current sample set according to $\pi^i(t,k)$
    ► Estimate the tracker state by (5)
    ► Predict the state of particles $s^i(t,k+1)$ by (8)
    if Non-associated Obstacle **then** ▷ New obstacle
      ► Generate new tracker sample set $\{s^i(t,k)\}$ for the obstacle, where, $i = 1,\ldots,N_s$
    end if
  end for
  for Non-associated tracker **do** ▷ Obstacle left the scene
    if the tracker has not been associated for a period **then**
      ► Prune track hypothesis.
    end if
  end for
end for

- Dataset 2: high way.
- Dataset 3: rural road
- Dataset 4: busy urban road

The algorithm is implemented in a standard PC with Windows 7 Enterprise OS, Intel CPU of 2.66 GHz. The development environment is MATLAB R2013b. Disparity map is obtained from LIBELAS toolkit [17] and particle filter functions from OpenCV [18] are integrated in the code. The run-time is about 3.4s per frame for on-road detection processing and 0.05s per frame for multiple obstacle tracking algorithm. The detection distance in disparity map is limited to 35m in front of the camera.

A. Experiments design

KITTI 3D tracklet labels provide the position and motion history of the obstacles appeared in the scene. After projection onto the image plane, the tracklet 2D position could be seen as ground truth trajectories for object detection and tracking. However, there are some considerations that need to be made for our detection and tracking result to be evaluated based on tracklet labels. First, tracklets only labels vehicles and pedestrians, other type of obstacles are not included, such as traffic cones in Dataset 2 . However, they should be detected as on-road obstacles for ADAS. Second, the tracklet labels also provide off road information of obstacles which are beyond the traffic area considered in our approach. Third, our stereo vision based detection distance is set up to 35m, while the tracklet reaches to 70m. Thus, the evaluation is constrained to the intersection between our detection results and labeled tracklets. To establish a comparable evaluation platform, a sequence of 100 frames is chosen from the four datasets respectively. Then, the GNN association is applied to pair our experimental results with tracklets. Hence, on-road tracklets are picked out by data association. From the tracklets label list, all the obstacle presences within 35m distance to the camera are preserved as ground truth. To ensure the evaluation result integrity, some special cases, like false alarms, missed detection and redundant detections, are also noted manually during the experiment of detection and tracking.

B. On-road obstacle detection

Dataset 1 contains 150 detections on road, 136 of them are associated with trackelet, the rest are false alarms and redundant detections. Dataset 2 contains 363 detections on road, 48 of them are associated with tracklet. In Dataset 2, except for false alarms and redundant detections, 309 non-associated detections are traffic cones standing on the road which are not listed in tracklet. Dataset 3 contains 37 detections on road, 34 of them are associated with tracklet, 4 are false alarms; Dataset 4 contains 257 detections, 215 of them are associated with tracklet; the rest of them are composed of detection of traffic signs, false alarms and redundant detections.

The detection results are evaluated following 5 indicators stated in Table I: false alarm, missed detection, redundant detection and average error of position and size. According to the experiment observation, most of false alarms are caused by the obstacles standing beside the road edge, e.g. trees. In Dataset 2, there is no obstacle beside road edge, so false alarms rarely occur. Most of missed detections appear particularly in the two sides of the image on the bottom when obstacles closely pass by the camera. They are usually induced by the errors of the disparity map. Imprecise disparity map values can also lead to redundant obstacle detections. The average error (AEC) illustrates the average distance between detected obstacle centroid and labeled tracklet centroid (ground truth). In addition, the average error which measures the variation of size scale (AES) is also listed in Table I. The two indicators are measured on pixel-level of the 2D image plan.

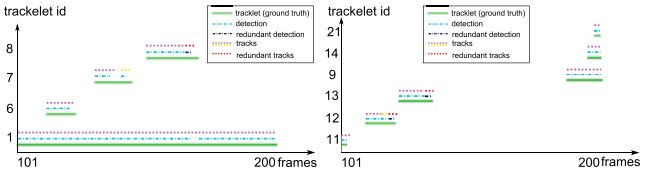
Measures	false alarm	missed detection	redundant detection	AEC	AES
Dataset 1	8.6%	4.0%	3.3%	5.3px	8.7px
Dataset 2	1.1%	3.3%	1.3%	6.8px	12.4px
Dateset 3	10.8%	8.1%	0%	9.6px	20.2px
Dataset 4	4.6%	3.1%	4.2%	10.9px	17.1px

Table I: Evaluation of the on-road detection results

C. Multiple obstacle tracking

Multiple target tracking can not only record and predict the motion of the obstacles, but also can deal with the occasionally missed detections and/or occlusions. As show in Fig. 4a, for the obstacle with tracklet id of 1, there's one frame of missed detection, while the track remains complete by filling the blank by prediction. During the tracking, four qualities are evaluated: rate of track fragmentation, rate of overlap, the average precision of tracker's position and size at each time step. As show in Table II, the tracking result

Figure 3: Examples of tracking results



(a) Dataset 1 (b) Dataset 2

Figure 4: Consistence of the tracks related to tracklets

is solid, with most track fragmentation rates under 6.81% during the sequence. The rate of tracks overlap in Dataset 1 is a bit less than in the other datasets, because in Dataset 1, the left side of road is lower than the right side. Thus, the disparity distribution of the road surface fluctuates, which makes the left-side vehicle hardly being detected. During the experiment, track fragmentation happens a lot, when obstacles move closely. They are illustrated in Fig. 4b, in which different colors of track stands for different tracks. Under the high speed circumstances, obstacles that move towards the camera have a high relative speed. As projected in the image, their centroid move faster and their sizes change rapidly over time. When the tracker can not catch up with the target for a certain period, it will be pruned and a new tracker will be created for the obstacle. One should notice here that, in Table II the missed detection and false alarms are effectively reduced by tracking process compare to Table I. Because tracking can fill up the gaps caused by instant missed detections, and prune the false alarms which happen occasionally.

Measures	false alarm	missed detection	track fragmentation	rate of overlap	AEC	AES
Dataset 1	5.3%	2.6%	2.36%	87.6%	5.5px	7.5px
Dataset 2	0%	2.4%	6.81%	96.3%	8.6px	11.8px
Dataset 3	7.8%	5.2%	0%	91.3%	19.5px	18.8px
Dataset 4	1.5%	2.7%	2.12%	93.6%	11.0px	18.0px

Table II: Evaluation of the multiple traget tracking results

V. Conclusion

In this paper, a previous work for on-road obstacle detection using stereo vision was improved by a reliable definition of traffic area and a multiple object tracking scheme based on particle filtering. Road traffic area extraction is first integrated in the detection process. On this ROI, connected-component extraction replaces Hough Transform for a flexible and fast detection of the obstacles. Moreover, multiple cues are considered to improve the detection accuracy. During the particle filtering, a self-adaptive gate for data association and dynamic filter noise function have been applied to enhance the tracking performance. Experimental results using a public dataset demonstrate that our detection and tracking system is efficient and reliable. Most obstacles appeared in 35m can well be detected and tracked. The proposed algorithm can work under dynamic circumstances without any prior-knowledge. Nevertheless, the use of 2D coordinates has a certain limit for further localization and tracking of the obstacles. Therefore, the next research step will be to focus on exploring the proposed approach from a 3D point of view.

References

[1] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *The International Journal of Robotics Research*, 29(14):1707–1725, 2010.

[2] Roberto Manduchi, Andres Castano, Ashit Talukder, and Larry Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous robots*, 18(1):81–102, 2005.

[3] Akihito Seki and Masatoshi Okutomi. Robust obstacle detection in general road environment based on road extraction and pose estimation. *Electronics and Communications in Japan (Part II: Electronics)*, 90(12):12–22, 2007.

[4] Amirali Jazayeri, Hongyuan Cai, Jiang Yu Zheng, and Mihran Tuceryan. Vehicle detection and tracking in car video based on motion model. *Intelligent Transportation Systems, IEEE Transactions on*, 12(2):583–595, 2011.

[5] Xue Fan, Shubham Mittal, Twisha Prasad, Suraj Saurabh, and Hyunchul Shin. Pedestrian detection and tracking using deformable part models and kalman filtering. *Journal of Communication and Computer*, 10:960–966, 2013.

[6] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and J-Y Audibert. Density-aware person detection and tracking in crowds. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2423–2430. IEEE, 2011.

[7] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *IJSCE, ISSN*, pages 2231–2307, 2012.

[8] Bihao Wang and Vincent Frémont. Fast road detection from color images. In *Intelligent Vehicles Symposium Proceedings, 2013 IEEE*, pages 1209–1214. IEEE, 2013.

[9] Bihao Wang, Vincent Frémont, and Sergio A Rodríguez. Color-based road detection and its evaluation on the KITTI road benchmark. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 31–36. IEEE, 2014.

[10] Zhencheng Hu and Keiichi Uchimura. UV-disparity: an efficient algorithm for stereovision based scene analysis. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 48–54. IEEE, 2005.

[11] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004.

[12] Hossein Tehrani Niknejad, Akihiro Takeuchi, Seiichi Mita, and David McAllester. On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):748–758, 2012.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[14] L.G. Shapiro and G.C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[15] Michael Isard and Andrew Blake. Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[17] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Computer Vision–ACCV 2010*, pages 25–38. Springer, 2011.

[18] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.