De SLAM Robuste à SLAM Dynamique par Auto-apprentissage des outliers

A. Bojko[1] R. Dupont[1] M. Tamaazousti[1] H. Le Borgne[1]

[1] CEA, LIST, Vision and Content Engineering Laboratory,
Point Courrier 173, F-91191 Gif-sur-Yvette, France

adrian.bojko@cea.fr romain.dupont@cea.fr
mohamed.tamaazousti@cea.fr herve.le-borgne@cea.fr

Résumé

Cet article concerne les SLAM Dynamiques, qui consistent à localiser une caméra en présence d'objets dynamiques. Nous proposons une approche plus intrinsèque que les SLAM Dynamiques actuels pour segmenter les objets dynamiques, en s'appuyant sur un processus robuste en général présent dans le SLAM, et non externe au SLAM. Nous montrons que nous pouvons auto-apprendre l'apparence des objets dynamiques via les outliers afin d'éviter les échecs du SLAM, dont ceux causés par les inversions de consensus de mouvement. Nous améliorons ORB-SLAM2 avec notre méthode, puis la validons en utilisant des données monoculaire, stéréo et RGB-D réelles de deux bases de données : TUM RGB-D et une nouvelle base que nous introduisons, focalisée sur les inversions de consensus de mouvement.

Mots Clef

slam, apprentissage, outliers, inversion de consensus

Abstract

This article deals with Dynamic SLAMs, which consist in localizing a camera in the presence of dynamic objects. We propose a more intrinsic way to segment dynamic objects, relying on a robust process usually present in SLAMs rather to segment dynamic objects, rather than external process. We show that we can self-learn the appearance of dynamic object via outliers to prevent SLAM failures, including those caused by motion consensus inversions. We improve ORB-SLAM2 with our method, then validate it using real monocular, stereo and RGB-D data from two datasets : TUM RGB-D and a new dataset that we introduce, which highlights motion consensus inversions.

Keywords

slam, learning, outliers, consensus inversion

1 Introduction

Les algorithmes de SLAM visuels (*Simultaneous Localization and Mapping*) sont fréquemment utilisés pour les véhicules autonomes [28], en réalité augmentée [21] et en robotique [7]. Ces algorithmes se basent en général sur des points d'intérêt de l'image et supposent que la caméra se déplace dans un environnement statique [17] (hypothèse de monde statique).

Il existe deux écoles de pensée qui étendent le SLAM pour faire face à des situations où l'hypothèse du monde statique n'est pas vérifiée : les SLAM Robustes et les SLAM Dynamiques. Chacune considère respectivement différentes entités de la scène : en 3D, le bruit et les objets dynamiques (i.e. en mouvement) ; en 2D, lors de la détection de points, des *inliers* / *outliers* (entrées / sorties aberrantes) et des masques. La figure 1 montre ces relations et comment nous les combinons pour former notre approche.

Les SLAM Robustes de l'état de l'art tels que ORB-SLAM2 [26] utilisent RANSAC (*Random Sample Consensus*) ou des fonctions de coût robustes lors des optimisations des poses de caméra. Ces méthodes détectent le mouvement dominant de la scène (consensus de mouvement) et excluent les éléments qui ne respectent pas le consensus. Ces dernières sont appelées *outliers*, par opposition à *inliers*. Les SLAM Robustes échouent lorsque la majeure partie de la scène ne correspond pas au mouvement réel de la caméra par rapport à l'arrière-plan, une situation que nous appelons "inversion de consensus de mouvement". Les situations où cela peut se produire incluent par exemple les foules (mouvements chaotiques dans la majeure partie de l'image) et les environnements urbains (véhicules passant devant la caméra).

Les SLAM Dynamiques récents tels que DynaSLAM [3] segmentent les objets dynamiques puis masquent les points d'intérêt correspondants, les empêchant d'être inclus dans les calculs du SLAM. [3] montre que la qualité du masquage a un effet direct sur les performances du SLAM.

Nous proposons une nouvelle approche qui mélange les deux précédents : un SLAM à la fois Dynamique et Robuste, comme illustré dans la figure 2. Nous avons séparé les *outliers* en deux catégories : les *outliers* éparses et les *outliers* denses.

Nous supposons que pour les SLAMs basés sur des points d'intérêt, les *outliers* éparses caractérisent le bruit tandis que les *outliers* denses caractérisent les objets dynamiques. Notre approche utilise uniquement les *inliers* et les *out-*

Méthode	Composition de la scène 3D	Image 2D (Points d'intérêt)
SLAM Robuste	Environnement statique + Bruit	**Points d'intérêt** = Inliers + Outliers
SLAM Dynamique	Environnement statique + Objets dynamiques	**Points d'intérêt** = Non masqués (statiques) + Masqués (dynamiques)
Notre SLAM	Environnement statique + Bruit + Objets dynamiques	**Points d'intérêt** = Inliers + Outliers éparses + Outliers denses

FIGURE 1 – Classification des SLAMs : relation entre différentes interprétations de la scène 3D et les points d'intérêt. Nous supposons que le bruit est lié aux *outliers* éparses et les objets dynamiques aux *outliers* denses ; cette supposition est la clé pour lier les deux approches et apprendre l'apparence d'objets dynamiques à partir des *inliers* et *outliers*.

liers d'un SLAM robuste pour auto-apprendre l'apparence des objets dynamiques, sans utiliser de processus externes qui choisissent quels objets sont dynamiques. Nous présentons également une nouvelle base de données qui inclut des inversions de consensus de mouvement, que nous avons utilisé pour tester la robustesse et la détection dynamique d'objets de ORB-SLAM2 [26], un SLAM de référence amélioré avec notre méthode. Notre base de données contient des séquences faciles, sans inversions de consensus de mouvement, et des séquences difficiles avec des inversions de consensus de mouvement qui font échouer le SLAM ; nous montrons que les inversions de consensus de mouvement provoquent des échecs du SLAM tels que les faux départs et que le problème ne peut être ignoré.

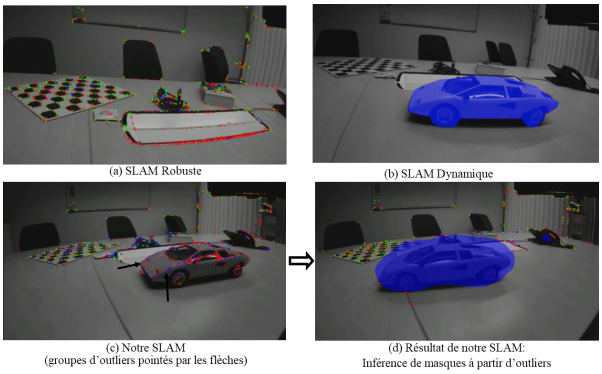

FIGURE 2 – (a) : SLAM Robuste : le bruit de l'image est rejeté. (b) SLAM Dynamique : les objets dynamiques sont masqués, indépendamment du SLAM. (c) : Notre SLAM : nous identifions les groupes d'*outliers* denses qui apparaissent soudainement, ce qui indique les objets dynamiques. (d) A travers notre pipeline, nous apprenons l'apparence des objets dynamiques qui génère des groupes d'*outliers*, et inférons des masques pour toute la séquence.

En partie 2, nous dressons l'état de l'art des SLAM Dynamiques. Nous présentons notre approche en partie 3, notre base de données en partie 4 et terminons par nos expériences en parties 5.

2 État de l'art

Les SLAM dynamiques actuels sont divisés en ceux qui visent à détecter uniquement des points d'intérêt statiques, et ceux qui masquent les points d'intérêt dynamiques après leur détection.

2.1 Détection seulement de points statiques

Les approches qui ne détectent que des points d'intérêt statiques sont des détecteurs de points qui ne détectent que des points d'intérêt stables i.e. non dynamiques. SuperPointVO [11] calcule des points d'intérêt et les classe en stable / instable / à ignorer ; ces étiquettes sont déterminées en exécutant un mini SLAM sur de courtes séquences vidéo, puis en observant la stabilité des points 3D reconstruits. Cela fait que SuperPointVO est a priori incapable d'apprendre d'algorithmes SLAM déjà robustes tel que ORB-SLAM2 car les points reconstruits ne sont plus mis en correspondance au moment où ils deviennent instables – les points reconstruits deviennent instables quand l'objet sur lequel ils sont se déplace, mais quand l'objet se déplace les points d'intérêt correspondants quittent les fenêtres de mise en correspondance. Une méthode similaire est MD-NET [33], qui utilise des informations de mouvement annotées, et fait ainsi une hypothèse forte sur la nature des objets dynamiques. Spécifiquement développé pour l'estimation de mouvement propre, DS-DSO [38] optimise simultanément les points d'intérêt et la profondeur, tout en intégrant une notion de temps lors de son entraînement.
Il y a plusieurs limites communes à tous ces détecteurs de points d'intérêt : la méthode pour intégrer les informations sur les objets dynamiques dépend du réseau de neurones profonds sous-jacent ; donc ils ne peuvent pas facilement être utilisés sur d'autres détecteurs, et notamment les détecteurs non appris (qui n'utilise pas d'apprentissage automatique). Leur comportement en situation d'inversion de consensus de mouvement n'est pas connu car leurs données d'entraînement n'incluent pas ces situations. L'utilisation d'un détecteur de points d'intérêt profond nécessite le remplacement du détecteur original dans le SLAM, ce qui peut être indésirable.

2.2 Masquage des *outliers*

Les méthodes de masquage d'*outliers* consistent en l'ajout à des SLAM existants de filtres qui masquent les objets dynamiques. La plupart des méthodes, tel que [3, 39, 44], masquent les objets dynamiques après la détection de points d'intérêt ; les points localisés sur un objet dynamique sont soit supprimés, soit traités à part, par exemple

pour suivre l'objet dynamique correspondant. De rares méthodes commencent par le masquage des objets dynamiques et ne détectent des points d'intérêt que sur les zones de l'image non masquées [19].

Les approches géométriques n'utilisent aucune forme d'apprentissage automatique [30]. Les approches qui s'appuient sur le flot optique [1, 7, 8] calculent le déplacement de pixels entre images mais peuvent ne pas fonctionner quand les objets dynamiques couvrent la majorité de la scène ou ont un mouvement erratique [37]. Les approches qui s'appuient sur les cartes de profondeur [20, 36] utilisent l'information 3D afin d'identifier des objets saillants et déterminer s'ils sont dynamiques, mais ne fonctionnent pas si la distance aux objets est trop courte ou trop importante : par exemple, pour une Kinect, (caméra de profondeur utilisée par [35]), la portée est de 0.5m à 4m. Les méthodes qui s'appuient sur des regroupement / avant/arrière-plan [23, 32, 37] identifient les objets dynamiques en les groupant et en assignant des probabilités à des points avec des déplacements similaires, mais sont coûteuses en temps de calcul et ne fonctionnent pas correctement avec les mouvements dégénérés tel que les mouvements très lents [30].

Les approches bout-en-bout intègrent des informations sur des objets dynamiques pendant l'entraînement, en s'appuyant soit sur des approches non apprises pour générer la base d'apprentissage [43], soit sur l'ajout de RNN (Réseaux Neuronaux Récurrents) [13, 41]. Comme les SLAMs utilisant des CNN (Réseaux de Neurones Convolutifs) sont déjà relativement robustes aux objets dynamiques [9], les SLAM CNN bout-en-bout sont prometteurs. Néanmoins, les approches bout-en-bout sont limitées par l'étendue de la base d'entraînement, avec de mauvaises performances s'il n'y a pas suffisamment de données (par ex. la séquence EuRoC MH04 dans [41]) ; cette limite n'est pas applicable au SLAM non-appris puisqu'ils ne sont pas besoin de bases de données. Les approches entièrement apprises sont imprévisibles dans des situations d'inversion de consensus de mouvement car elles n'ont pas été entraînées pour les gérer.

Les approches améliorées par de l'apprentissage sont des SLAM géométriques améliorés avec des méthodes d'apprentissage automatique. Les méthodes sémantiques [19, 34, 39] identifient des classes connues puis les masquent, mais elles sont limitées par l'étendue et la disponibilité des bases d'apprentissage, et peuvent ne pas fonctionner avec des objets inconnus [44]. Les méthodes de stabilité / éphéméralité [2, 12] apprennent l'apparence des objets dynamiques en comparant des reconstructions 3D dans le temps, mais requièrent des passages de plusieurs heures dans le même environnement, et peuvent demander des capteurs supplémentaires tels que des LiDARs. Les approches hybrides [3, 4, 16, 22, 27, 29, 31, 42, 44] combinent au moins deux des approches précédentes (tel que carte de profondeur + sémantique) et leurs hypothèses sur les objets dynamiques (objet saillant + classe "humain").

3 Approche proposée

Toutes ces approches font des hypothèses directement sur la nature des objets dynamiques ; nous supposons que les *outliers* du SLAM suffisent à caractériser les objets dynamiques de séquences faciles. Autrement dit, nous supposons que l'information nécessaire pour identifier un objet dynamique est déjà présente dans l'environnement et exploitable à travers un SLAM Robuste basé points d'intérêt dans des situations sans inversion de consensus de mouvement. Nous ne faisons pas de suppositions directement sur la nature de ces objets et montrons que l'on peut apprendre l'apparence de divers objets dynamiques uniquement à partir de quelques séquences vidéo de l'ordre de 1000 images.

3.1 Pipeline

Notre approche consiste globalement à segmenter les objets dynamiques à partir des *outliers* SLAM de séquences sans inversion de consensus de mouvement, puis à apprendre leur apparence. Nous utilisons ensuite ce masque pour rendre le SLAM robuste aux séquences difficiles qui présentent des inversions de consensus de mouvement.
Nous avons observé que les apparitions soudaines et concentrées d'*outliers* correspondent à des objets dynamiques s'il n'y a pas d'inversion de consensus de mouvement. Un exemple courant est un petit objet reconstruit par le SLAM qui se déplace soudainement. En effet, le SLAM rejette, par construction, les points d'intérêt non statiques : des groupes d'*inliers* qui deviennent soudainement des *outliers* sont en réalité des points d'intérêt dynamiques. Les *outliers* sont normalement plutôt dispersés : quand il y a une augmentation soudaine et localisée de la densité d'*outliers*, il est probable qu'il y ait au mêmes coordonnées un objet violant l'hypothèse de monde statique, plutôt que juste des points d'intérêt isolés. Nous avons également observé que les mêmes objets qui génère des *outliers* dans certaines séquences peuvent provoquer des inversions de consensus dans d'autres, par exemple une voiture vu de loin puis de très près. Ces deux observations nous ont mené à construire notre système en trois parties :

1. **Apprentissage** :
   (a) **Prétraitement des *outliers* et des *inliers*** : accumulation d'*outliers* et d'*inliers* de séquences faciles (suivi d'une étape de filtrage pour les SLAMs non-déterministes) ;
   (b) **Segmentation outlier-objet** : segmentation et apprentissage de l'apparence d'objets dynamiques, en comparant la densité d'*outliers* et d'*inliers* entres différentes images ;
2. **Inférence (Masquage d'objets dynamiques)** : ajout d'un module de masquage au SLAM afin de retirer les points d'intérêt se trouvant sur des objets dynamiques.

3.2 Prétraitement des *outliers* et des *inliers*

(A) Accumulation d'*outliers* et d'*inliers*. L'accumulation de points consiste en la collecte des coordonnées des

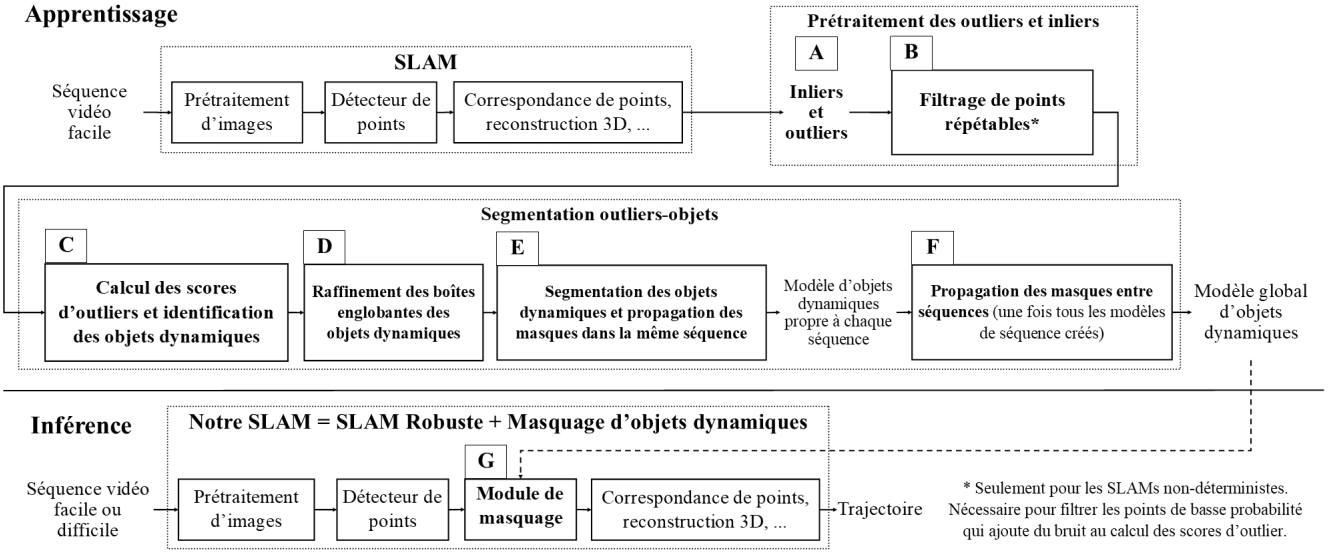

FIGURE 3 – Pipeline d'entraînement et d'inférence de notre approche SLAM Robuste.

outliers et des *inliers* pour chaque image de chaque séquence facile. Une fois initialisé, un SLAM qui s'appuie sur des points d'intérêts effectue en général trois étapes en boucle :

1. Détection de points d'intérêt 2D ;
2. Correspondance 2D-3D entre points d'intérêt et points 3D déjà reconstruits + triangulation de nouveaux points 3D ;
3. Ajustement de faisceaux (*Bundle adjustment*) : optimisation robuste des correspondances 2D-3D et des poses caméra.

Nous enregistrons les coordonnées des *outliers* et des *inliers* à l'issue de l'ajustement de faisceaux : les *outliers* sont les points d'intérêt dont la correspondance 2D-3D est rejetée, et les *inliers* sont ceux dont la correspondance n'a pas été rejetée. Nous sauvegardons à la fin du SLAM la trajectoire complète de la caméra. Afin d'augmenter la précision autant que possible, nous empêchons tout arrêt anticipé de l'ajustement de faisceaux (ce qui est en général nécessaire pour un traitement temps réel).

(B) Filtrage de points répétables (seulement pour les SLAMs non-déterministes). Les SLAMs peuvent être non-déterministes à cause de la parallélisation des calculs et de l'initialisation aléatoire (par ex. avec RANSAC), et nous avons observé que les *inliers* / *outliers* rarement observés sur plusieurs exécutions ont tendance à être du bruit et donc peu utiles à la détection d'objets dynamiques. Ainsi, nous exécutons le SLAM n fois et supprimons les *outliers* et *inliers* rarement observés.

3.3 Segmentation outlier-objet

Le but de cette étape est d'identifier les objets dynamiques et de propager leurs masques à toutes les images de toutes les séquences faciles où ils sont vus. L'idée clé est d'utiliser des fenêtres glissantes pour trouver des changements anormaux dans *outliers* et *inliers*, qui sont caractéristiques des objets dynamiques lorsqu'ils se déplacent. Une fois que nous avons identifié toutes les fenêtres contenant des objets dynamiques, nous les propageons d'abord dans leurs séquences respectives à l'aide de réseaux de segmentation vidéo, puis nous les propageons à différentes séquences à l'aide de réseaux de segmentation sémantique. On apprend enfin un modèle global, qui connaît l'apparence de tous les objets dynamiques de toutes les séquences.

(C) Calcul des scores d'*outliers* et localisation des objets dynamiques. À chaque image de chaque séquence, nous utilisons des fenêtres glissantes rectangulaires de tailles différentes et de pas fixe afin d'évaluer les variations du ratio *inlier* / *outlier*.
Soit w une fenêtre sur l'image p et w' la fenêtre correspondante sur l'image p'. Alors le score d'outlier S est :

$$S = \left(\frac{\text{densité d'}\textit{outliers}\text{ de } w}{\text{densité d'}\textit{inliers}\text{ de } w}\right) / \left(\frac{\text{densité d'}\textit{outliers}\text{ de } w'}{\text{densité d'}\textit{inliers}\text{ de } w'}\right) \quad (1)$$

Nous considérons que w contient un **objet dynamique** si S est inférieur à un seuil S_{max}, fixé par l'utilisateur. Pour compenser le mouvement de la caméra, nous calculons l'homographie $H = K.R_{p,p'}.K^{-1}$ où $R_{p,p'}$ est la rotation relative entre les images comparées (calculée à l'aide de la trajectoire SLAM) et K la matrice intrinsèque de la caméra. H est la rotation de la caméra transformée en déplacement de pixels. Nous appliquons H à la fenêtre w' lorsque nous la comparons avec la fenêtre correspondante w pour que les deux fenêtres correspondent au même emplacement physique [1].

1. Nous supposons que la rotation de la caméra est beaucoup plus importante que sa translation sur de courtes périodes donc rendant une

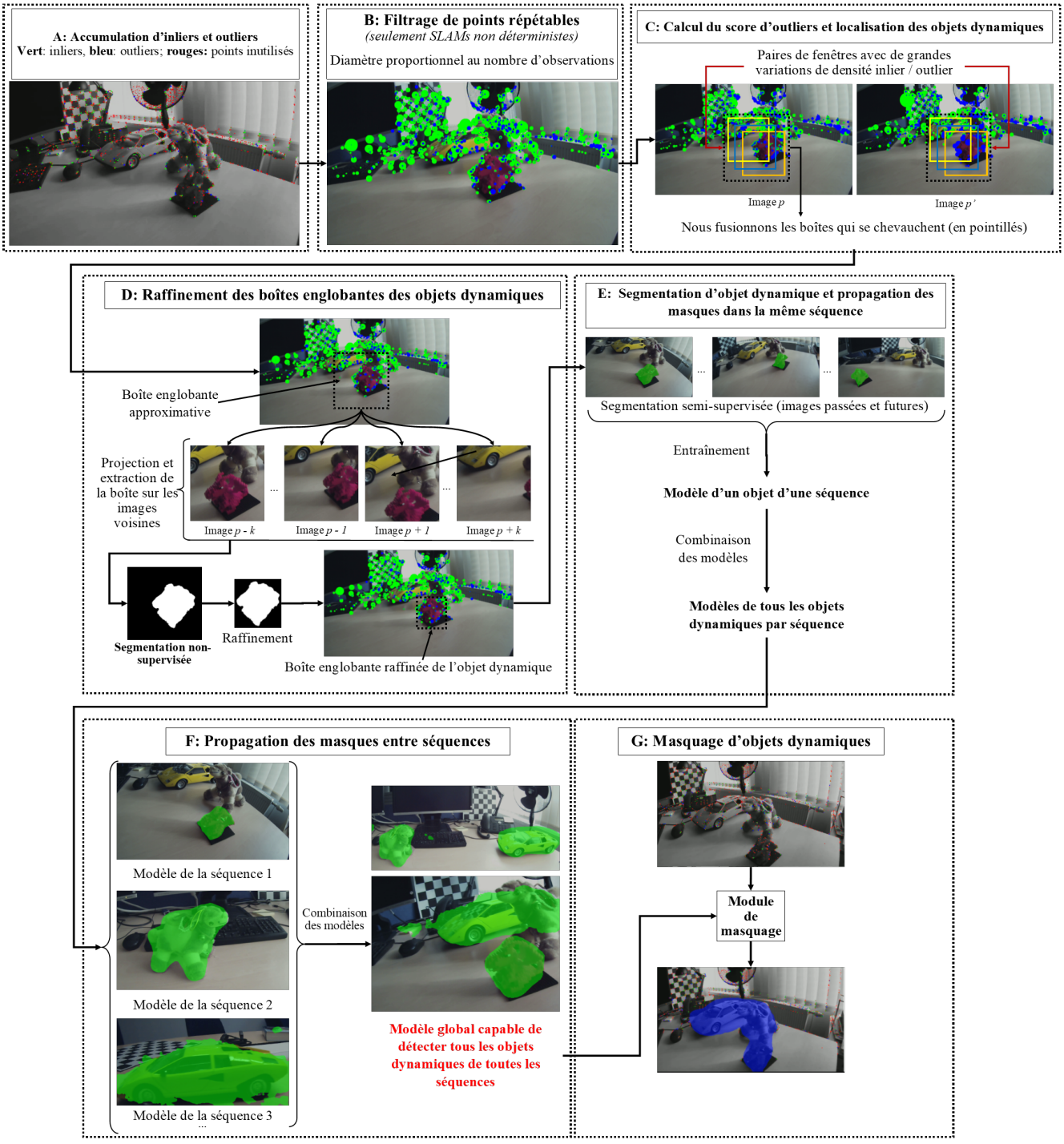

FIGURE 4 – Illustration des étapes de notre approche SLAM Robuste. Les étapes A-B sont des étapes de prétraitement, où nous collectons les *outliers* et *inliers*. Les étapes C à F sont des étapes d'apprentissage : nous calculons les boîtes englobantes initiales pour les objets dynamiques en recherchant des variations importantes de densités d'*outliers*, puis nous raffinons et segmentons les objets dynamiques, et enfin les propageons à toutes les séquences. L'étape G est l'inférence : nous calculons les masques pour les images d'entrée du SLAM et masquons les points d'intérêt dans les zones masquées.

(D) Raffinement des boîtes englobantes des objets dynamiques. Nous fusionnons d'abord tous les boîtes englobantes qui se superposent, même partiellement, sur une même image. Ensuite, nous projetons chaque boîte englobante sur les k prochaines images ainsi que les k images précédentes avec compensation en rotation (cf. étape précédente). Nous créons ensuite une sous-séquence vidéo à partir des boîtes englobantes projetées et utilisons l'implémentation originale de COSNet [25] (un réseau de segmentation non supervisé d'objets vidéo) sur l'image centrale. COSNet est ici approprié car ce réseau segmente les objets d'intérêt par corrélation, ce qui est le cas de l'objet dynamique dans la sous-séquence, en particulier quand il se déplace. Pour chaque sous-séquence, si aucun objet n'est segmenté par COSNet, nous supprimons la boîte englobante ; autrement, nous redimensionnons la boîte englobante afin qu'elle soit tangente à l'objet segmenté par COSNet.

(E) Segmentation des objets dynamiques et propagation des masques dans la même séquence. Pour chaque objet dynamique, maintenant que nous avons une boîte englobante le délimitant correctement, nous utilisons l'implémentation originale de SiamMask [40] (un réseau de segmentation semi-supervisé d'objets vidéo) vers les images futures et passées, nous arrêtant si la boîte englobante atteint les bords de l'image. Ce réseau à la particularité de fournir la segmentation d'un objet sur toute une vidéo uniquement à partir d'une boîte englobante initiale précise ; nous obtenons ainsi la segmentation de l'objet sur les images proches. Si le nombre d'images segmentées est suffisant ($\geq p_{min}$), nous entraînons un réseau DeepLabv3+ [2] [6] en utilisant les images segmentées par SiamMask. Nous avons ainsi un modèle par objet dynamique et par séquence. Nous réalisons, par séquence, une inférence pour chaque objet dynamique de cette dernière puis superposons les masques, que nous utilisons enfin pour entraîner un modèle connaissant l'apparence de tous les objets dynamiques de la séquence.

(F) Propagation des masques entre séquences. Finalement, ayant un modèle d'objets dynamiques par séquence, nous segmentons chaque séquence avec chaque modèle, et produisons une base d'apprentissage en superposant tous les masques produits pour une même séquence. Nous entraînons enfin un modèle global d'objets dynamiques en utilisant la base d'apprentissage.

3.4 (G) Masquage d'objets dynamiques

Cette étape est directe : ayant le modèle global des objets dynamiques, entraîné sur toutes les séquences faciles, nous inférons des masques pour les images d'entrée et supprimons tous les points d'intérêt des zones masquées.

(a) *easy_dragon_1* (b) *easy_drom_1* (c) *easy_lambo_1*

(d) *hard_dragon_1* (e) *hard_drom_1* (f) *hard_lambo_1*

(g) *hard_lambo_2* (h) *hard_lambo_3* (i) *hard_lambo_static_cam*

FIGURE 5 – Notre base de données. Les séquences faciles ne causent d'inversion de consensus, mais les séquences difficiles oui.

4 Base de données d'inversion de consensus

Critère de mesure de robustesse. Nous proposons un critère qui s'appuie sur l'inversion de consensus de mouvement afin de quantifier le niveau de robustesse d'une séquence vidéo. Nous considérons d'abord que le SLAM subit une inversion de consensus de mouvement si :

1. Le SLAM continue à se localiser une fois initialisé (sauf perte ponctuelle par ex. à cause d'une occlusion) ;
2. Le SLAM subit un échec critique i.e. a une erreur en trajectoire (ATE RMSE) un ordre de grandeur plus important que celui qu'il aurait pour une trajectoire similaire, dans le même environnement, mais sans inversion de consensus ;
3. Les échecs critiques ont lieu lorsque le SLAM fonctionne en conditions normales (caméra bien calibrée, nombre de points d'intérêt / image approprié, etc.) et en temps réel (pas de lecture accélérée ou ralentie).

Qualitativement, l'inversion de consensus de mouvement est assez souvent évidente car la caméra suit l'objet dynamique, comme nous le voyons dans la figure 6. Supposons que nous ayons une base de données de séquences vidéo classées en deux niveaux de difficulté selon la présence d'inversions de consensus de mouvement :

1. **Facile** : n'inclut pas d'inversion de consensus.
2. **Difficile** : inclut des inversions de consensus.

Nous considérons qu'un SLAM est robuste aux inversions de consensus de mouvement s'il peut traiter toutes les séquences faciles et difficiles sans subir d'échecs critiques.

correction homographique suffisante.

2. https ://github.com/srihari-humbarwadi/person_segmentation_tf2.0

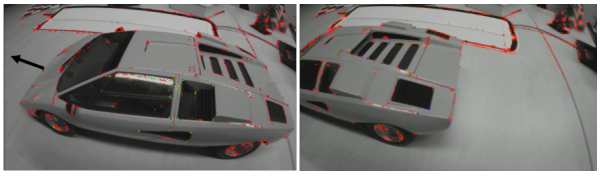

FIGURE 6 – Exemple d'inversion de consensus de mouvement. La caméra est statique, et pourtant le SLAM interprète le déplacement de la voiture vers la gauche comme un déplacement de la caméra vers la droite (*inliers* en vert, *outliers* en bleu, points non utilisés en rouge).

TABLE 1 – Base de données d'inversion de consensus.

Niveau de difficulté	Séquence	Effet attendu de l'inversion de consensus
Facile	easy_dragon_1	-
Facile	easy_drom_1	-
Facile	easy_lambo_1	-
Difficile	hard_lambo_2	Faux départ
Difficile	hard_lambo_static_camera_1	Faux départ
Difficile	hard_dragon_1	Faux départ
Difficile	hard_drom_1	Faux départ
Difficile	hard_lambo_1	Faux départ
Difficile	hard_lambo_3	Faux départ

4.1 Séquences pour tester la robustesse

Nous avons testé les séquences de cinq bases de données avec ORB-SLAM2 monoculaire : TUM RGB-D [35], EuRoC [5], KITTI [15], Aqualoc [14] et UZH-FPV Drone Racing [10]. Nous n'avons constaté aucune inversion de consensus de mouvement, donc nous avons décidé de créer notre propre base de données de test de robustesse. Notons que l'inversion de consensus de mouvement est une situation assez peu courante, mais qui peut provoquer des échecs SLAMs – elle ne doit donc pas être ignorée.

Pour créer notre base de données, nous avons utilisé une caméra MYNT EYE D1000-120 (stéréo 1280x720 à 30 FPS). Nous avons enregistré les séquences vidéo du tableau 1, séparées en facile / difficile. *faux départ* indique que le SLAM estime un mouvement alors qu'il ne devrait pas (car la caméra est statique ou car il n'y a pas suffisamment de points d'intérêt statiques visibles), et *dérive* indique que le SLAM dérive significativement de la trajectoire attendue. Nous illustrons les séquences dans la figure 5. Nous calculons la vérité terrain avec ORB-SLAM2 stéréo robuste aux séquences difficiles (en utilisant notre méthode), avec l'arrêt anticipé de l'ajustement de faisceaux désactivé.

5 Expériences

5.1 Protocole expérimental

Nous évaluons notre méthode sur la base de données TUM RGB-D [35] (séquences dynamiques) et sur notre propre base de données. Le but de la base de données TUM RGB-D est l'évaluation des systèmes RGB-D SLAM ; elle a été enregistrée à l'aide d'une Microsoft Kinect en 640x480 à 30Hz, et les poses de la caméra au sol ont été obtenues à partir d'un système de capture de mouvement. Les séquences *dynamiques* sont un ensemble de huit séquences qui enregistrent les personnes en mouvement, nommées au format *fr3_sitting_** ou *fr3_walking_**.

Nous utilisons considérons ORB-SLAM2 comme algorithme SLAM de référence. Compte tenu de son comportement non déterministe, nous calculons la médiane de l'ATE RMSE (*Absolute Trajectory Error*) sur 10 exécutions. L'ATE RMSE est lui-même calculé à l'aide d'un alignement Sim (3) sur les images clés.

Nous utilisons les mêmes paramètres dans notre pipeline pour toutes les expériences, sur TUM RGB-D ou sur notre base de données. Nous avons trouvé empiriquement les paramètres appropriés : des fenêtres glissantes de taille 100x100 / 200x200 / 300x300 / 400x400 avec un pas de 50px, une différence de 3 images pour calculer les scores d'*outliers*, un intervalle de ± 15 images (environ 1s à 30 FPS) pour la segmentation non supervisée, un minimum de $p_{min} = 50$ images pour accepter une segmentation semi-supervisée, et le score d'outlier max $S_{max} = 0.15$ pour déterminer si une fenêtre contient un objet dynamique.

Pour rendre ORB-SLAM2 robuste, nous utilisons les séquences *fr3_sitting_static* et *fr3_walking_static* lors des tests sur TUM RGB-D, et nos trois séquences faciles (*easy_dragon_1*, *easy_drom_1*, *easy_lambo_1*) pour tester notre base de données. Ce choix est dans le but de prouver que nous n'avons besoin que de très peu de données pour détecter un objet vidéo et apprendre son apparence.

5.2 Comparaisons avec l'état de l'art

TABLE 2 – ATE RMSE (m) évalué sur TUM RGB-D en mode monoculaire (sans cartes de profondeur)

Séquence	ORB-SLAM2	DynaSLAM [3])	**Nous**
fr3_walking_rpy	**0.075**	N/D	0.116
fr3_walking_static	0.005	N/D	**0.004**
fr3_walking_xyz	0.183	**0.012**	**0.012**
fr3_walking_halfsphere	0.018	**0.017**	0.018
fr3_sitting_rpy	0.036	N/D	**0.030**
fr3_sitting_static	0.008	N/D	**0.004**
fr3_sitting_xyz	0.008	**0.007**	0.009
fr3_sitting_halfsphere	**0.015**	N/D	0.074

TABLE 3 – ATE RMSE (m) évalué sur TUM RGB-D en mode RGB-D (avec cartes de profondeur)

Séquence	ORB-SLAM2	DS-SLAM [44])	DynaSLAM [3]	Unified [39]	**Nous**
fr3_walking_rpy	0.152	0.444	**0.035**	0.032	0.137
fr3_walking_static	0.014	0.008	0.006	**0.005**	0.007
fr3_walking_xyz	0.282	0.025	0.015	**0.014**	**0.014**
fr3_walking_halfsphere	0.321	0.030	0.025	0.019	**0.018**
fr3_sitting_rpy	**0.021**	N/D	N/D	N/D	0.024
fr3_sitting_static	0.011	**0.007**	N/D	N/D	0.010
fr3_sitting_xyz	0.011	N/D	0.015	**0.009**	0.013
fr3_sitting_halfsphere	0.025	N/D	0.017	0.021	**0.014**

Nous comparons nos résultats avec d'autres SLAMs Dynamiques de l'état de l'art, également basés sur le masquage d'objets dynamiques : DS-SLAM [44], DynaSLAM [3] and Unified [39]. Nous reportons leurs résultats dans nos tableaux : le tableau 2 présente les résultats en monoculaire et le tableau 3 les résultats en mode RGB-D.

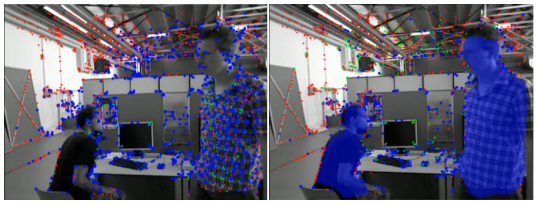

FIGURE 7 – Masquage de *fr3_walking_rpy* (mode RGB). Le masquage empêche l'apparition d'un grand nombre d'*outliers* (en bleu) et augmente la quantité d'*inliers* (en vert). Les points d'intérêt non utilisés sont en rouge.

Nous avons obtenu des résultats comparables à l'état de l'art dans les deux modes. Les autres approches [44, 3, 39] utilisent Mask R-CNN [18] ou FCIS [24], qui sont des réseaux de segmentation d'instance entraînés sur des bases de données externes. Nous avons entraîné notre réseau en utilisant les *outliers* de seulement deux séquences monoculaires de la base de données TUM RGB-D, *fr3_sitting_static* et *fr3_walking_static*, mais nous obtenons des résultats comparables ou même meilleurs.
Cette expérience montre que notre hypothèse selon laquelle les *outliers* sont caractéristiques des objets dynamiques est vérifiée et que nous pouvons segmenter les objets dynamiques à l'aide de ces *outliers*. ORB-SLAM2 souffre d'une inversion de consensus de mouvement, comme le montre la figure 7, mais pas en monoculaire. En observant les *inliers* sur les personnes lorsqu'elles se déplacent, nous avons observé que les points d'intérêts ne sont plus mis en correspondance dès qu'ils sortent d'une petite fenêtre autour du point reconstruit en 3D, ni triangulés lorsque la personne se déplace ; cela ne se produit pas en RGB-D, d'où les meilleures performances de ORB-SLAM2 monoculaire.
Enfin, nous avons cherché pourquoi il y a une dégradation pour la séquence *fr3_walking_rpy* : nous avons observé que notre masquage n'est pas toujours complet lorsque la caméra tourne de 90°, ce qui est fréquent dans cette séquence et explique la dégradation.

5.3 Évaluation de la robustesse sur notre base de données

Les tableaux 4 et 5 donnent les résultats en monoculaire et les tableaux 6 et 7 en stéréo.

Résultats en monoculaire. L'impact des masques sur les séquences faciles est négligeable, mais il y a une nette amélioration pour les séquences difficiles. Le SLAM devient effectivement résistant face à toutes les séquences difficiles et ne souffre plus d'inversion de consensus de mouvement.

Résultats en stéréo. Le masquage en mode stéréo a des effets positifs et réduit l'ATE RMSE d'un ordre de grandeur par rapport à l'absence de masquage. Le SLAM devient résistant à toutes les séquences difficiles car il ne souffre plus d'inversion de consensus de mouvement.
Dans l'ensemble, les SLAM monoculaire et stéréo bénéficient du masquage des objets dynamiques et deviennent résistants aux inversions de consensus de mouvement. Les gains sont importants en terme d'ATE RMSE sur les séquences difficiles, tout en n'ayant quasiment aucun effet sur les séquences faciles. ORB-SLAM2 stéréo est devenu résistant à l'inversion de consensus de mouvement.

TABLE 4 – Impact du masquage sur le SLAM monoculaire, séquences faciles. En mètres.

Séquence	ATE RMSE (pas de masques)	ATE RMSE (avec des masques)
easy_dragon_1	0.038	**0.034**
easy_drom_1	**0.007**	**0.007**
easy_lambo_1	0.081	**0.011**

TABLE 5 – Impact du masquage sur SLAM monoculaire pour les séquences dures provoquant de faux départs ou une dérive importante. En mètres.

Séquence	ATE RMSE (pas de masques)	ATE RMSE (avec des masques)
hard_lambo_2	Faux départ	**Pas de faux départ**
hard_lambo_static_camera_1	Faux départ	**Pas de faux départ**
hard_dragon_1	0.049	**0.009**
hard_drom_1	0.056	**0.006**
hard_lambo_1	0.110	**0.007**
hard_lambo_3	0.098	**0.013**

TABLE 6 – Impact du masquage sur SLAM stéréo, séquences faciles. En mètres.

Séquence	ATE RMSE (pas de masques)	ATE RMSE (avec des masques)
easy_dragon_1	0.044	**0.040**
easy_drom_1	0.004	**0.001**
easy_lambo_1	**0.008**	0.009

TABLE 7 – Impact du masquage sur SLAM stéréo pour les séquences dures provoquant des faux départs ou une dérive importante. En mètres.

Séquence	ATE RMSE (pas de masques)	ATE RMSE (avec des masques)
hard_lambo_2	Faux départ	**Pas de faux départ**
hard_lambo_static_camera_1	**Pas de faux départ**	**Pas de faux départ**
hard_dragon_1	0.066	**0.001**
hard_drom_1	0.022	**0.003**
hard_lambo_1	0.048	**0.001**
hard_lambo_3	0.074	**0.006**

Limitations. La propagation des masques peut échouer si un objet dynamique est trop similaire à l'arrière-plan ou occlut car il confond l'algorithme de semi-supervision. L'objet doit être bien reconstruit avant de se déplacer : de préférence bien texturé et le SLAM doit avoir le temps de reconstruire l'objet. Enfin, les séquences doivent être filmées dans des conditions favorables au SLAM : bon éclairage, pas de rotation pure ni de mouvements trop rapides.

6 Conclusion

Nous avons introduit et testé une méthode pour segmenter un objet dynamique à partir des *outliers* et des *inliers* SLAM extraits de séquences vidéo faciles et avons montré comment le modèle appris se généralise aux séquences difficiles, prouvant notre hypothèse initiale. La méthode résultante combine les avantages du SLAM Robuste et du SLAM Dynamique, et donne ainsi des résultats meilleurs que l'état de l'art sur TUM RGB-D. Nous évitons ainsi les inversions de consensus sur notre base de données. Parmi nos perspectives, nous envisageons de développer et distribuer notre base de données et d'étendre notre méthode aux SLAMs directs.

Références

[1] Alcantarilla, P.F., Yebes, J.J., Almazán, J., Bergasa, L.M. : On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In : IEEE International Conference on Robotics and Automation (ICRA) (2012)

[2] Barnes, D., Maddern, W., Pascoe, G., Posner, I. : Driven to Distraction : Self-Supervised Distractor Learning for Robust Monocular Visual Odometry in Urban Environments. In : IEEE International Conference on Robotics and Automation (ICRA) (2018)

[3] Bescos, B., Fácil, J.M., Civera, J., Neira, J. : DynaSLAM : Tracking, Mapping, and Inpainting in Dynamic Scenes. IEEE Robotics and Automation Letters **3**(4) (2018)

[4] Brasch, N., Bozic, A., Lallemand, J., Tombari, F. : Semantic Monocular SLAM for Highly Dynamic Environments. In : IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2018)

[5] Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R. : The EuRoC micro aerial vehicle datasets. The International Journal of Robotics Research **35**(10) (2016)

[6] Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. : Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In : Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV. Lecture Notes in Computer Science, Springer International Publishing, Cham (2018)

[7] Cheng, J., Sun, Y., Chi, W., Wang, C., Cheng, H., Meng, M.Q. : An Accurate Localization Scheme for Mobile Robots Using Optical Flow in Dynamic Environments. In : IEEE International Conference on Robotics and Biomimetics (ROBIO) (2018)

[8] Cheng, J., Sun, Y., Meng, M.Q.H. : Improving monocular visual SLAM in dynamic environments : an optical-flow-based approach. Advanced Robotics **33**(12) (2019)

[9] Cimarelli, C., Cazzato, D., Olivares-Mendez, M.A., Voos, H. : A case study on the impact of masking moving objects on the camera pose regression with CNNs. In : IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (2019)

[10] Delmerico, J., Cieslewski, T., Rebecq, H., Faessler, M., Scaramuzza, D. : Are We Ready for Autonomous Drone Racing ? The UZH-FPV Drone Racing Dataset. In : International Conference on Robotics and Automation (ICRA) (2019)

[11] DeTone, D., Malisiewicz, T., Rabinovich, A. : Self-Improving Visual Odometry. arXiv :1812.03245 [cs] (2018)

[12] Dymczyk, M., Stumm, E., Nieto, J., Siegwart, R., Gilitschenski, I. : Will It Last ? Learning Stable Features for Long-Term Visual Localization. In : International Conference on 3D Vision (3DV) (2016)

[13] Feng, T., Gu, D. : SGANVO : Unsupervised Deep Visual Odometry and Depth Estimation With Stacked Generative Adversarial Networks. IEEE Robotics and Automation Letters **4**(4) (2019)

[14] Ferrera, M., Creuze, V., Moras, J., Trouvé-Peloux, P. : AQUALOC : An underwater dataset for visual–inertial–pressure localization :. The International Journal of Robotics Research (2019)

[15] Geiger, A., Lenz, P., Urtasun, R. : Are we ready for autonomous driving ? The KITTI vision benchmark suite. In : 2012 IEEE Conference on Computer Vision and Pattern Recognition (2012)

[16] Hachiuma, R., Pirchheim, C., Schmalstieg, D., Saito, H. : DetectFusion : Detecting and Segmenting Both Known and Unknown Dynamic Objects in Real-time SLAM. In : Proceedings British Machine Vision Conference (BMVC) (2019)

[17] Hartley, R., Zisserman, A. : Multiple View Geometry in Computer Vision. Cambridge University Press, New York, NY, USA, 2 edn. (2003)

[18] He, K., Gkioxari, G., Dollár, P., Girshick, R. : Mask R-CNN. In : IEEE International Conference on Computer Vision (ICCV) (2017). https ://doi.org/10.1109/ICCV.2017.322

[19] Kaneko, M., Iwami, K., Ogawa, T., Yamasaki, T., Aizawa, K. : Mask-SLAM : Robust Feature-Based Monocular SLAM by Masking Using Semantic Segmentation. In : IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2018)

[20] Kim, D.H., Kim, J.H. : Effective Background Model-Based RGB-D Dense Visual Odometry in a Dynamic Environment. IEEE Transactions on Robotics **32**(6) (2016)

[21] Klein, G., Murray, D. : Parallel Tracking and Mapping for Small AR Workspaces. In : IEEE and ACM International Symposium on Mixed and Augmented Reality (2007)

[22] Li, P., Qin, T., Shen, S. : Stereo Vision-Based Semantic 3D Object and Ego-Motion Tracking for Autonomous Driving. In : Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Springer International Publishing (2018)

[23] Li, S., Lee, D. : RGB-D SLAM in Dynamic Environments Using Static Point Weighting. IEEE Robotics and Automation Letters **2**(4) (2017)

[24] Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y. : Fully Convolutional Instance-Aware Semantic Segmentation. In :

IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

[25] Lu, X., Wang, W., Ma, C., Shen, J., Shao, L., Porikli, F. : See More, Know More : Unsupervised Video Object Segmentation With Co-Attention Siamese Networks. In : IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

[26] Mur-Artal, R., Tardós, J.D. : ORB-SLAM2 : An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics **33**(5) (2017)

[27] Reddy, N.D., Singhal, P., Chari, V., Krishna, K.M. : Dynamic body VSLAM with semantic constraints. In : IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2015)

[28] Rosinol, A., Rebecq, H., Horstschaefer, T., Scaramuzza, D. : Ultimate SLAM ? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High Speed Scenarios. IEEE Robotics and Automation Letters **PP** (2018)

[29] Runz, M., Buffier, M., Agapito, L. : MaskFusion : Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In : 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (2018)

[30] Saputra, M.R.U., Markham, A., Trigoni, N. : Visual SLAM and Structure from Motion in Dynamic Environments : A Survey. ACM Comput. Surv. **51**(2) (2018)

[31] Schorghuber, M., Steininger, D., Cabon, Y., Humenberger, M., Gelautz, M. : Slamantic - leveraging semantics to improve vslam in dynamic environments. In : The IEEE International Conference on Computer Vision (ICCV) Workshops (2019)

[32] Scona, R., Jaimez, M., Petillot, Y.R., Fallon, M., Cremers, D. : StaticFusion : Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments. In : IEEE International Conference on Robotics and Automation (ICRA) (2018)

[33] Song, Y., Zhu, D., Li, J., Tian, Y., Li, M. : Learning Local Feature Descriptor with Motion Attribute for Vision-based Localization. arXiv :1908.01180 [cs] (2019)

[34] Strecke, M., Stuckler, J. : EM-Fusion : Dynamic Object-Level SLAM With Probabilistic Data Association. In : The IEEE International Conference on Computer Vision (ICCV) (2019)

[35] Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D. : A benchmark for the evaluation of RGB-D SLAM systems. In : IEEE/RSJ International Conference on Intelligent Robots and Systems (2012)

[36] Sun, Y., Liu, M., Meng, M.Q.H. : Improving RGB-D SLAM in dynamic environments : A motion removal approach. Robotics and Autonomous Systems **89** (2017)

[37] Sun, Y., Liu, M., Meng, M.Q.H. : Motion removal for reliable RGB-D SLAM in dynamic environments. Robotics and Autonomous Systems **108** (2018)

[38] Tang, J., Ambrus, R., Guizilini, V., Pillai, S., Kim, H., Gaidon, A. : Self-Supervised 3D Keypoint Learning for Ego-motion Estimation. arXiv :1912.03426 [cs] (2019)

[39] Wang, K., Lin, Y., Wang, L., Han, L., Hua, M., Wang, X., Lian, S., Huang, B. : A Unified Framework for Mutual Improvement of SLAM and Semantic Segmentation. In : International Conference on Robotics and Automation (ICRA) (2019)

[40] Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.S. : Fast Online Object Tracking and Segmentation : A Unifying Approach. In : Proceedings of the IEEE conference on computer vision and pattern recognition (2019)

[41] Wang, S., Clark, R., Wen, H., Trigoni, N. : End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. The International Journal of Robotics Research **37**(4-5) (2018)

[42] Xu, B., Li, W., Tzoumanikas, D., Bloesch, M., Davison, A., Leutenegger, S. : MID-Fusion : Octree-based Object-Level Multi-Instance Dynamic SLAM. In : International Conference on Robotics and Automation (ICRA) (2019)

[43] Ye, C., Mitrokhin, A., Fermüller, C., Yorke, J.A., Aloimonos, Y. : Unsupervised Learning of Dense Optical Flow, Depth and Egomotion from Sparse Event Data. arXiv :1809.08625 [cs] (2018)

[44] Yu, C., Liu, Z., Liu, X., Xie, F., Yang, Y., Wei, Q., Fei, Q. : DS-SLAM : A Semantic Visual SLAM towards Dynamic Environments. In : IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2018)