S3LAM: SLAM à Scène Structurée

Mathieu Gonzalez, Eric Marchand, Amine Kacete, Jérôme Royan

S^{3}LAM: SLAM à Scène Structurée

Mathieu GONZALEZ[1], Eric MARCHAND[2], Amine KACETE[1], Jérôme ROYAN[1]

[1]IRT b<>com, 1219 Av. des Champs Blancs, 35510 Cesson-Sévigné

[2]Univ Rennes, Inria, IRISA, CNRS, Rennes, France

mathieu.gonzalez@b-com.com, eric.marchand@irisa.fr
amine.kacete@b-com.com, jerome.royan@b-com.com

Résumé – Nous proposons un nouveau système de SLAM (Simultaneous Localization And Mapping) qui utilise la segmentation sémantique des images. La segmentation sémantique présente un intérêt car elle contient une information haut niveau pouvant rendre plus précis le SLAM. Notre contribution est double : i) un système de SLAM basé sur ORB-SLAM2 capable de créer une cartographie sémantique. ii) Une modification de l'algorithme d'ajustement de faisceaux pour contraindre la position des points 3D de la cartographie en utilisant des a priori géométriques. Nous évaluons notre approche sur plusieurs jeux de données publics et montrons que notre approche améliore l'estimation de pose de caméra en comparaison de l'état de l'art.

Abstract – We propose a new SLAM (Simultaneous Localization And Mapping) system that uses the semantic segmentation of objects in the scene. Semantic information is relevant as it contains high level information which may make SLAM more accurate and robust. Our contribution is twofold: i) A new SLAM system based on ORB-SLAM2 that creates a semantic map made of *clusters* of points corresponding to objects instances. ii) A modification of the classical Bundle Adjustment formulation to constrain each *cluster* using geometrical priors. We evaluate our approach on sequences from several public datasets and show that it improves camera pose estimation with respect to state of the art.

1 Introduction

Le but du SLAM (Simultaneous Localization And Mapping) est de construire une cartographie de l'environnement observé par une caméra se déplaçant dans l'espace tout en estimant sa pose. Il s'agit d'un algorithme fondamental pour la robotique et la réalité augmentée qui est capable d'estimer correctement la pose d'une caméra dans des scènes à petite et grande échelle [1]. La cartographie peut prendre différente formes mais est toujours géométrique et manque d'information sémantique, qui pourrait rendre le SLAM plus robuste [2]. De plus l'information sémantique peut permettre de développer certaines applications. Par exemple les robots mobiles peuvent avoir besoin de reconnaître les objets dans une scène pour choisir leur trajectoire. Grâce aux réseaux de neurones convolutionnels (CNN) des méthodes de détections d'objets et de segmentation [3] sont désormais disponibles. Elles peuvent être intégrées au SLAM afin de créer des cartographies sémantiques [4]. La sémantique peut également permettre d'améliorer le SLAM par exemple pour la relocalisation [5], la gestion des objets dynamiques [6]. Certains travaux utilisent les objets comme des points de repères de haut niveau [7]. Dans ce papier nous proposons un système de SLAM monoculaire appelé S^{3}LAM pour *Structured Scene SLAM* basé sur ORB-SLAM2 et qui crée une cartographie sémantique en utilisant un réseau de segmentation panoptique [8]. Nous proposons de créer une nouvelle représentation pour la scène, dans laquelle les objets sont vus comme des groupes de points avec une information sémantique. Ceci nous permet de créer une cartographie contenant des instances d'objets, visibles à droite dans la figure 1. Nous appliquons à ces objets des contraintes géométriques dépendant de leur classe, ce qui permet d'améliorer l'estimation de pose de caméra et la cartographie. De plus nous obtenons une cartographie plus haut niveau qu'une simple cartographie géométrique.

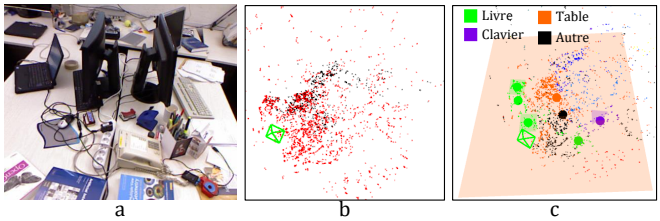

FIGURE 1 – (a) Image de la sequence fr1_desk. (b) cartographie construite par [1], (c) cartographie construite par notre approche avec objets et plans.

2 Etat de l'art

Le SLAM classique est résolu en estimant la pose de la caméra et la cartographie en maximisant la probabilité de ces variables compte tenu des images. ORB-SLAM2 [1] est considéré comme l'état de l'art actuel du SLAM visuel. Dans leur système, les mesures sont des points ORB. L'algorithme, inspiré par [9, 10] fonctionne en parallèle et permet d'affiner les estimations avec un ajustement par faisceaux (Bundle Adjustment, BA) local [11] qui minimise l'erreur de reprojection des points de la carte.

Les systèmes SLAM basés sur des objets consistent à détecter des objets dans la scène et à les insérer dans la carte pour ajouter des contraintes entre les images, ajoutant ainsi une cohérence temporelle [12, 7, 13]. Cela peut apporter de la robustesse au SLAM et de la précision en ayant accès à l'échelle des objets. Ces systèmes peuvent être divisés en deux grandes catégories. Tout d'abord les systèmes SLAM utilisant des objets connus à l'avance.

SLAM++ [12] propose d'estimer la pose des objets de la scène à partir d'images RGB-D. Chaque objet estimé est rendu à l'aide de son modèle 3D et la pose de la caméra est estimée en minimisant l'erreur d'alignement avec l'objet. D'autre part certains travaux proposent de modéliser des objets à l'aide de quadriques. QuadricSLAM [13] utilise des quadriques pour la localisation et la cartographie. L'idée principale est de générer des quadriques à partir des objets détectés en 2D. Les paramètres des quadriques et la pose de la caméra sont ensuite raffinés afin que la projection 2D des quadriques corresponde aux détections 2D des objets.

Les systèmes de SLAM planaires consistent à détecter des structures planes dans la scène et à les utiliser comme repères de haut niveau [14, 7]. L'objectif est double : premièrement, les plans sont généralement de grandes structures et peuvent donc contraindre différentes parties d'une scène. Deuxièmement la détection de plans dans la scène permet d'avoir une meilleure compréhension de sa structure physique, ce qui peut être nécessaire pour certaines applications. [7] utilise des plans pour contraindre la carte du SLAM. Les plans sont estimés à partir de 3 réseaux de neurones différents qui estiment la profondeur, les normales et la segmentation des plans. A partir de ces estimations, des plans sont insérés dans la carte. La distance point-plan est ensuite minimisée. [14] propose un SLAM monoculaire utilisant des plans pour contraindre la structure de la scène. Les plans sont estimés à l'aide d'un RANSAC sur toute la carte, ils doivent donc être suffisamment grands et la fréquence d'images doit être suffisamment faible pour que le RANSAC converge, ainsi leur approche est limitée à 5 images par secondes (ips). Les points planaires sont ensuite projetés dans les plans et optimisés pour minimiser l'erreur de reprojection.

3 S^{3}LAM : un SLAM de *clusters*

Dans S^{3}LAM, la carte est représentée comme un ensemble de nuages de points, appelés *clusters* et regroupés selon l'instance d'objet à laquelle ils appartiennent. Notre objectif est d'utiliser les connaissances préalables sur ces objets pour enrichir le SLAM, améliorer l'estimation de la pose de la caméra ainsi que pour obtenir une meilleure représentation de la structure de la scène. Nous représentons la pose de la caméra à l'instant i par la transformation entre le repère monde $\mathcal{F}_w$ et le repère de la caméra $\mathcal{F}_{c_i}$ avec la matrice homogène ${}^{c_i}\mathbf{T}_w \in SE(3)$. Nous segmentons chaque image clé à l'aide d'un réseau de segmentation panoptique. Nous mettons à jour la distribution de classe des points de la carte à l'aide de la sortie du réseau de segmentation. Ceci nous permet de créer des *clusters* sémantiques qui correspondent de manière unique aux objets. Pour certains *clusters* correspondant à des objets plans, un plan est ajusté à l'aide des points 3D. Un ajustement de faisceau (BA) avec une contrainte planaire est ensuite appliqué pour affiner l'estimation de la pose de la caméra et la position des points de la carte. Par rapport à l'état de l'art, notre algorithme se rapproche des méthodes qui représentent les objets sous forme de quadriques [13, 7] qui sont très génériques et des approches qui estiment des plans pour représenter la carte [14, 7]. Cependant, contrairement aux approches qui utilisent les quadriques, la notre donne une représentation plus proche de la réalité, ce qui peut encore améliorer la précision de l'estimation de la pose de la caméra. Et contrairement aux approches qui utilisent des plans, la notre n'a pas besoin d'informations de profondeur, ni de CNN spécifique pour estimer les plans et peut fonctionner en temps réel avec des plans de taille variée.

3.1 Création des *clusters*

Segmentation panoptique : La segmentation panoptique est une combinaison de la segmentation sémantique où on attribue à chaque pixel une classe donnée, et de la segmentation d'instance où plusieurs objets de la même classe sont segmentés séparément. Pour représenter le réseau de segmentation panoptique nous définissons une fonction $g(\mathbf{I}_i) \rightarrow \mathbf{P}_i, \mathbf{L}_i$ qui, étant donné une image RGB à l'instant i, $\mathbf{I}_i$ estime une carte de probabilité $\mathbf{P}_i \in [0,1]^{W \times H \times C}$ [1]. Ainsi pour chaque pixel $\mathbf{x} = (u, v)$ de l'image on peut obtenir une distribution de probabilité $(\mathbf{P}_i(u,v,1), ..., \mathbf{P}_i(u,v,C))$ où $\mathbf{P}_i(u,v,c)$ correspond à la probabilité que ce pixel appartienne à la classe c. La deuxième sortie du réseau est la carte d'instance $\mathbf{L}_i \in \mathbb{N}^{W \times H}$ dans laquelle chaque objet est segmenté et reçoit un identifiant unique. Cet identifiant n'est pas cohérent dans le temps. Pour résoudre ce problème nous proposons une stratégie en 2 étapes : au niveau du réseau de segmentation et dans le SLAM. Tout d'abord, pour chaque instance détectée dans $\mathcal{L}^i$ nous calculons l'intersection avec les instances de $\mathcal{L}^{i-1}$ et $\mathcal{L}^{i-2}$ pour suivre l'identifiant. Avec cette stratégie, les *clusters* sont bien définis. Cependant cette approche ne fonctionne pas lorsqu'un *cluster* qui a quitté le champ de vision de la caméra réapparaît dans l'image. Pour cela, nous définissons une fonction pour fusionner les *clusters* lorsque la distance entre leur centre de gravité est inférieure à un seuil τ et que plus de 80% des descripteurs des points caractéristiques des *clusters* correspondent.

Création de *clusters* : Suite à la segmentation 2D, nous définissons une fonction $f(\{\mathbf{P}_i\}, \{\mathbf{x}_i\}) \rightarrow \mathbf{p}^{3D}$ où $\{\mathbf{P}_i\}$ est un ensemble de cartes de probabilité à différents instants, $\{\mathbf{x}_i\}$ est un ensemble de points caractéristiques correspondant à un point 3D ${}^w\mathbf{X}$ et $\mathbf{p}^{3D} = (p_1, ..., p_C)$ est sa distribution de probabilité. Cette fonction est la fusion de plusieurs observations et peut être écrite à l'aide de la règle de Bayes :

$$p_c = \mathbb{P}(c|\{\mathbf{P}_i\}) = \frac{1}{Z}\mathbb{P}(c|\{\mathbf{P}_{i-1}\})\mathbf{P}_i(u,v,c) \qquad (1)$$

où c est la classe de ${}^w\mathbf{X}$ et Z est un facteur de normalisation. Ainsi f nous permet d'obtenir une carte sémantique où chaque point ${}^w\mathbf{X}$ a une distribution de probabilité $\mathbf{p}^{3D}$, un identifiant l extrait de la carte d'instance $\mathbf{L}^i$ ainsi qu'une classe $c^* = \text{argmax}\, \mathbf{p}^{3D}$. Cette fusion permet à la carte d'être temporellement cohérente, même si la segmentation panoptique est bruitée. En utilisant cette carte sémantique, nous pouvons créer K groupes de points $\{O_{k,k\in[1,K]}\}$ selon leur classe sémantique et leur instance. Chaque groupe peut être défini comme $O_k = \{\{{}^w\mathbf{X}\}, c_k, l_k\}$ où $\{{}^w\mathbf{X}\}$ est la position d'un ensemble de points appartenant au groupe, c_k est la classe du groupe et l_k son identifiant.

3.2 Application des contraintes planaires

Estimation de la structure : De nombreux objets peuvent être approximés avec un modèle géométrique plus ou moins complexe, allant d'un plan au modèle 3D exact de l'objet. Ceci permet tout d'abord d'ajouter des contraintes au processus d'optimisation

1. W et H représentent les dimensions d'une image, C est le nombre de classes.

pour améliorer l'estimation de la pose. Deuxièmement, nous obtenons une représentation plus précise du monde en comprenant les structures qu'il contient, contrairement aux méthodes récentes qui utilisent des quadriques pour représenter des objets et qui ne représentent qu'approximativement l'étendue spatiale des objets mais pas leur forme. Dans notre travail à titre d'exemple nous proposons de modéliser certains objets à l'aide de plans. Non seulement nous modélisons de grandes surfaces telles que le sol, mais également de petits objets comme des livres. Les plans sont représentés en utilisant un vecteur 4D $\pi = (a,b,c,d)^\top$ avec $||\pi||_2 = 1$ et les points du plan $\mathbf{X}$ en coordonnées homogènes satisfont l'équation $\pi^\top \mathbf{X} = 0$.

Contrairement à la plupart des SLAM planaires, nous n'avons pas besoin d'utiliser plusieurs CNN spécifiques [7] ou une information de profondeur pour estimer les paramètres du plan, ce qui limite l'applicabilité de ces systèmes. Au lieu de cela, pour chaque *cluster* a priori planaire, nous ajustons un plan en utilisant les points 3D triangulés de ce *cluster*. Ceci est fait en calculant une SVD à l'intérieur d'une boucle de RANSAC pour rendre l'estimation plus robuste à une mauvaise classification et triangulation de la même manière que [14]. Cependant contrairement à [14] nous ne sommes pas limités à de simples scènes avec peu de très grands plans. En effet, lorsque nous créons des *clusters* sémantiques, nous sommes capables d'adapter des plans même pour de petits objets et nous pouvons ainsi appliquer notre approche dans une plus grande variété de scènes. De plus, la procédure est facilitée par le regroupement et se fait en temps réel comparé aux 5 ips de [14].

Optimisation de la carte : Nous avons choisi d'inclure la contrainte comme régularisateur comme on peut le voir dans l'équation (2).

$$
\begin{aligned}
{}^{c}\hat{\mathbf{T}}_w, {}^{w}\hat{\mathbf{X}} = \underset{{}^{c}\mathbf{T}_w, {}^{w}\mathbf{X}}{\operatorname{argmin}} & \sum_{i,j} \rho(||\mathbf{x}_{i,j} - proj({}^{c_i}\mathbf{T}_w, {}^{w}\mathbf{X}_j)||_{\Sigma_{i,j}}) \\
& + \sum_{k} \sum_{j \in O_k} \rho(||\pi_k^\top {}^{w}\mathbf{X}_j||_\sigma)
\end{aligned}
\tag{2}
$$

où $||\pi_k^w \mathbf{X}_j||$ est la distance 3D entre le point 3D ${}^{w}\mathbf{X}_j$ et le plan π_k qui correspond au *cluster* O_k, σ correspond à son incertitude, et ρ est une fonction de coût robuste (dans notre cas la fonction de Huber).

4 Expériences

4.1 Détails des expériences

S^{3}LAM fonctionne à 20 ips.Toutes les expériences sont réalisées sur un ordinateur de bureau avec une Nvidia RTX2070. La valeur de l'incertitude point-plan σ est fixée à 100.

Jeu de données. Notre approche est évaluée sur des séquences du jeu de données TUM RGB-D [15]. Nous montrons également que notre approche peut fonctionner à plus grande échelle et dans des scènes extérieures en l'évaluant sur des séquences du jeu de données KITTI raw [16].

Métriques. Pour tenir compte de la stochasticité inhérente à ORB-SLAM2, nous exécutons chaque séquence 10 fois et rapportons la médiane de la RMSE de l'erreur de trajectoire absolue (ATE, Absolute Trajectory Error définie dans [15]).

TABLE 1 – Comparaison de l'ATE (mm) sur les données TUM.

Sequence	[1]	[7] (plans)	[7] (quadriques)	[14]	S^{3}LAM
fr1_xyz	9.2	10.3	10.0	-	**8.8**
fr1_floor	18.1	16.9	-	-	**14.7**
fr1_desk	13.9	12.9	**12.6**	-	13.2
fr2_xyz	2.4	**2.2**	**2.2**	-	2.4
fr2_desk	8.0	7.3	**7.1**	-	7.8
fr3_nos_txt_near	20.3	-	-	-	**15.3**
fr3_nos_txt_near (un plan)	20.3	-	-	-	**13.5**
fr3_nos_txt_near (boucle)	14.5	-	-	11.4	**10.9**
fr3_str_txt_near	14.0	-	-	**10.6**	11.2
fr3_str_txt_far	10.6	-	-	**8.8**	9.2
fr1 (moy.)	13.9	13.4	-	-	**12.2**
fr2 (moy.)	5.2	4.8	**4.7**	-	5.1
fr3 (moy.)	13.0	-	-	**10.3**	10.4

4.2 Impact des contraintes planaires

Dans cette section, nous étudions l'impact de l'ajout de contraintes planaires à la formulation du BA classique. Nous rapportons les résultats de nos expériences dans le tableau 1. Nous comparons notre approche à ORB-SLAM2 [1] et à [7] qui rapportent le plus grand nombre d'expériences et utilisent à la fois des quadriques et des plans. Nous comparons également l'approche basée sur le plan de [14] qui rapporte des expériences sur quelques séquences planaires.

Comme nous pouvons le voir, les contraintes planaires améliorent la localisation de la caméra par rapport ORB-SLAM 2. L'amélioration la plus importante est obtenue sur des scènes presque parfaitement planes comme fr1_floor. La ligne fr3_nost_txt _near (un plan) dans le tableau 1 montre notre approche en utilisant un seul plan pour tous les groupes de livres ce qui montre que plus les plans sont grands plus l'amélioration de l'ATE est importante. Les séquences fr3_nos_txt_near et fr3_nos_txt_near (un plan) sont exécutées avec la fermeture de boucle désactivée pour mieux montrer l'impact de notre approche. L'activation de la fermeture de la boucle est indiquée ligne fr3_nos_txt_near (boucle). Les séquences fr2 sont les plus difficiles pour notre approche car les plans principaux ne sont pas bien segmentés et encombrés, ce qui donne une estimation bruitée.

Par rapport à l'approche basée sur 3 CNNs de [7] nous pouvons voir que nous obtenons de meilleurs résultats pour les scènes planaires, cependant dans les scènes où les plans ne sont pas bien segmentés, l'utilisation des quadriques apporte une amélioration plus importante, liée au fait que la détection d'objet est plus précise que la segmentation panoptique. Par rapport à l'approche de [14], nous obtenons des résultats similaires, en revanche, notre approche est plus générique car nous ne sommes pas limités à des scènes principalement planaires et notre approche tourne en temps réel.

Ces résultats montrent que notre approche est générique car elle améliore l'estimation de la pose de la caméra dans une grande variété de scènes, tandis que d'autres approches se concentrent soit sur des scènes contenant des objets, soit sur des scènes parfaitement planes. Nous rapportons également dans le tableau 2 l'évaluation de notre approche dans une scène en exterieur, sur le jeu de données KITTI raw. Comme nous pouvons le voir nous améliorons l'estimation de la pose de la caméra par rapport à ORB-SLAM2.

4.3 Analyse qualitative de S^{3}LAM

Nous montrons dans la figure 2 quelques exemples de cartes obtenues avec notre approche. Comme nous pouvons le voir, chaque objet (dont le centroïde est représenté par une sphère) présent dans

TABLE 2 – Comparaison de l'ATE (cm) sur les données KITTI.

sequence	ORB-SLAM 2 [1]	S^{3}LAM
0926-0011	17.7	**15.5**
0926-0013	18.0	**7.5**
0926-0014	76.2	**64.5**
0926-0056	49.8	**49.3**

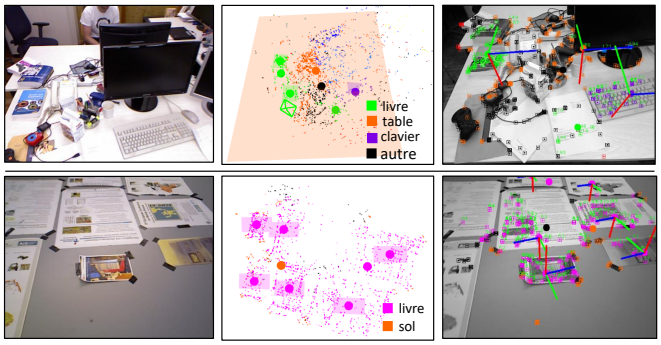

FIGURE 2 – Exemples de cartes obtenues par notre approche. De gauche à droite : image de la séquence, cartographie contenant les *clusters* et les plans, systèmes de coordonées des plans projetés dans l'image (normale affichée en rouge).

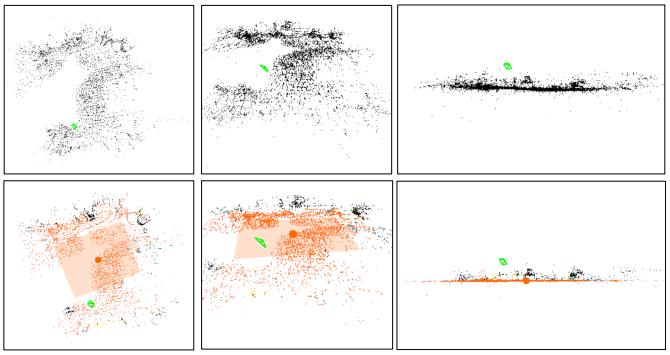

FIGURE 3 – Exemple de carte créées par [1] (en haut) et notre approche (en bas) avec une vue de dessus, de ¾ et de côté.

la scène a été instancié de manière unique, ce qui conduit à une carte plus compréhensible. Nous montrons également pour chaque objet planaire le plan estimé, correspondant aux objets table, clavier et livre. Dans la figure 3 nous montrons une comparaison de la carte obtenue avec ORB-SLAM2 et la carte obtenue avec notre BA planaire. Comme nous pouvons le voir, la partie inférieure de la carte correspondant au sol (visible en orange) est plus plane avec notre approche.

5 Conclusion

Dans cet article, nous avons proposé un nouveau système de SLAM sémantique monoculaire appelé S^{3}LAM. Notre méthode utilise la segmentation panoptique 2D d'une séquence d'images pour créer des *clusters* de points 3D selon leur classe et leur instance. Ce regroupement nous permet d'estimer de manière robuste la structure de certains *clusters* et de modifier la formulation du BA avec des contraintes structurelles. Nous montrons sur des séquences de plusieurs jeux de données publics que notre approche conduit à une amélioration de l'estimation de la pose de la caméra.

Références

[1] R. Mur-Artal et al. ORB-SLAM2 : An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. on Robotics*, 33(5) :1255–1262, 2017.

[2] C. Cadena et al. Past, present, and future of simultaneous localization and mapping : Toward the robust-perception age. *IEEE Trans. on robotics*, 32(6) :1309–1332, 2016.

[3] A. Kirillov et al. Panoptic feature pyramid networks. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[4] J. McCormac et al. Semanticfusion : Dense 3D semantic mapping with convolutional neural networks. In *2017 IEEE Int. Conf. on Robotics and automation (ICRA)*, pages 4628–4635, 2017.

[5] C. Toft et al. Long-term 3D localization and pose from semantic labellings. In *IEEE Int. Conf. on Computer Vision Workshops*, pages 650–659, 2017.

[6] B. Bescos et al. DynaSLAM : Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4) :4076–4083, 2018.

[7] M. Hosseinzadeh et al. Real-time monocular object-model aware sparse SLAM. In *2019 Int. Conf. on Robotics and Automation (ICRA)*, pages 7123–7129, 2019.

[8] Yuxin W. et al. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[9] E. Mouragnon et al. Real time localization and 3D reconstruction. In *2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 363–370, 2006.

[10] G. Klein et al. Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234, 2007.

[11] B. Triggs et al. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[12] R. F. Salas-Moreno et al. SLAM++ : Simultaneous localisation and mapping at the level of objects. In *IEEE Conf. on computer vision and pattern recognition*, pages 1352–1359, 2013.

[13] L. Nicholson et al. QuadricSLAM : Dual quadrics from object detections as landmarks in object-oriented SLAM. *IEEE Robotics and Automation Letters*, 4(1) :1–8, 2018.

[14] C. Arndt et al. From points to planes - adding planar constraints to monocular SLAM factor graphs. In *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4917–4922, 2020.

[15] J. Sturm et al. A benchmark for the evaluation of RGB-D SLAM systems. In *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, Oct. 2012.

[16] A. Geiger et al. Are we ready for autonomous driving ? the kitti vision benchmark suite. In *2012 IEEE Conf. on computer vision and pattern recognition*, pages 3354–3361, 2012.