

**L'état de l'art du TPE3**  
**Cartographie et localisation simultanées avec la vision robotique**  
**Encadrement : Alain Boucher (IFI), Serge Stinckwich (MSI)**  
**Étudiant: Vu Hong Thuan.**

Dans les années récentes, le problème SLAM (Simultaneous Localization and Mapping) est considéré comme un problème important de la robotique.

Dans le problème de SLAM, on doit résoudre le problème de l'incertitude : l'incertitude de la position de robot, l'incertitude de la position de l'objet dans l'environnement, l'incertitude d'identifier et reidentifier des points d'intérêt dans l'environnement.

Le problème de SLAM est pour le robot, donc ses solutions doivent être réalisées en temps réel. Cela requiert que le temps de calculatoire, l'espace de stockage et l'incertitude sont petits pour construire une bonne carte acceptable et la trajectoire de robot.

### **Des approches pour le problème SLAM**

Pour résoudre le problème SLAM, plusieurs méthodes sont présentées. La méthode la plus connue est les méthodes probabilistes.

#### **Les méthodes probabilistes**

##### **EKF - Extended Kalma Filters**

La méthode la plus importante est EKF ([1],[2]). EKF est basé sur l'estimation gaussienne à chaque pose de robot et chaque point de repère. L'algorithme d'EKF contient trois étapes. Premièrement, dans l'étape de prédiction, on estime l'état de système (la position de robot, sa covariance, l'observation) au temps  $t$  en utilisant les données au temps  $t-1$ . Deuxièmement, dans l'étape d'observation, d'après l'observation de robot, on calcule la covariance de la nouvelle observation avec laquelle vieille. En fin, dans l'étape de mise à jour, on met à jour l'état de système courante avec la nouvelle observation.

L'incertitude du robot augmente de temps en temps par l'accumulation de bruit de mesure. Cette incertitude augmente jusqu'à le robot sent un point de repère vieille. À ce moment-là, les erreurs de ses poses se baissent. C'est à dire, si on augmente l'information sur les poses senties de robot, cette information se propage à tous les vieilles poses. C'est une caractéristique la plus importante de "SLAM posterior".

Plusieurs méthodes à améliorer l'EKF sont présentées telles que CEKF (Compressed EKF) ([3],[4]), Square Root SAM (Smooth and mapping) [5], SEIF (Sparse Extended Information Filter) [6], TJTF (Thin Junction Tree Filter)[7].

##### **Inférence de Bayes**

L'autre approche est de considérer le problème SLAM comme un modèle Markov et utiliser l'inférence de Bayes pour résoudre le problème [8]. Mais cette méthode n'est pas convenable à une carte avec beaucoup de point de repère. Il existe toujours en pratique que le nombre de point de repère est environ mille, millions points de repères.

##### **EM (Estimation et maximisation)**

En suite, on utilise l'EM ([9],[10]) pour estimer le maximum de vraisemblance de carte. Cette méthode se compose deux étapes. Dans l'étape de l'estimation, une estimation de toutes les poses de robot est réalisée à base de la carte courante et les données en tout le temps. Dans l'étape de maximisation, une carte plus meilleure est calculé à partir de l'estimation de la première étape. La carte meilleure est le maximum de l'espoir de la logarithme de vraisemblance jointe des poses de robot et les données en tout le temps.

L'EM a quelques avantages en comparant avec EKF. Premièrement, l'EM fournit une méthode sans demander de l'identification unique de point d'intérêt comme EKF. L'association de données est réalisée itérativement sur toutes les données d'observation. Donc, l'estimation dans le passé est revue. Mais, l'EM ne requiert pas une assume de bruit Gaussien comme l'EKF.

Mais l'EM présente quelques désavantages. Premièrement, l'EM procède toute la carte dans chaque étape. Donc, l'EM ne fournit pas une solution augmentant que l'estimation est meilleure si on ajoute une nouvelle observation comme l'EKF.

##### **Relaxation et « Multilevel Relaxation »**

Une l'autre méthode de SLAM relative à l'estimation de maximum de vraisemblance est la relaxation ([11],[12]). On utilise « Gausse-Seidel Relaxation » pour résoudre un système d'équation de problème SLAM. L'idée de cette méthode est « déplacer un noeud à sa position où ses voisinages pense qu'il est ». Cette méthode est appliquée beaucoup dans des algorithmes récents. Le « Multilevel Relaxation » [13] est en base d'une méthode multigrille (multigrid method) de résoudre un système d'équation différentielle partielle. Cet algorithme augmente la converge de Relaxation par optimiser la carte de chaque couche.

1. Relaxation « un couche » pour SLAM.

Relaxation « un couche » est une étape dans « multilevel relaxation ».

Entrée: un ensemble de relations entre des cadres. Chaque cadre est équivalente un pose de robot.

Sortie: un vecteur de maximum de vraisemblance de chaque cadre.

Étapes:

- Linéariser la fonction de mesure

- Calculer la fonction d'erreur quadrique

- Trouver le minimum de vraisemblance d'erreur quadrique par résoudre itérativement un système d'équation linéaire en utilisant « Gauss-Seidel Relaxation ».

## 2. « Multigrid »

C'est un processus de discrétiser et interpoler une équation différentielle partielle. à chaque couche, on restreint un problème et gérer un problème grossier par un opérateur de restriction. Un problème le plus grossier est résout. En fin, le problème est continué par un opérateur de interpolation.

## 3. Relaxation Multi-couche (Multilevel Relaxation)

1. On choisit chaque deux cadres de trajectoire de robot pour générer chaque niveau de grossièreté.
2. À chaque niveau, on implémente un opérateur Galerkin basé V-cycle de « multigrid » pour mettre à jour l'estimation de maximum de la vraisemblance à chaque nouvelle cadre.

## Filtreur de particule

Une autre appoche est le filtreur de particule. On attache chaque pose de robot avec des filtreurs de particule ([14],[15]). Chaque particule ont un poids de l'importance. Ce poids implique la relation de cette particule avec l'autre particule. Les filtreurs de particules sont échantillonné et re-échantillonné dans chaque fois ou le robot perçoit une nouvelle observation.

Les étapes de FastSLAM:

1. Initialiser SLAM.
  - Mettre la particule de la location de robot à la valeur de coordonnée de démarrer.
  - Mettre la carte à zéro.
2. Mettre à jour des particules

Si on a déjà reçu une odométrie. On générer par hasard une particule pour la nouvelle location de robot. La distribution de générer des particules base sur une échantillons de motion. L'échantillon de motion est un probabilité conditionnelle de nouvelle location du robot au temps  $t$ , sachant la location du robot au temps  $t-1$  et l'odométrie au temps  $t$ . Cette étape s'appelle l'étape d'échantillonnage probabiliste (probabilistic sampling step).

Si l'observation est reçue. à partir de chaque particule vieille, on génère une nouvelle particule selon l'état actuel. Les nouvelles particules s'appellent « proposal distribution » En suite, on re-échantillonne les particules par remplace de tous les vieilles particules avec les nouvelles particules à base de leur poids d'importance. Cela s'appelle l'étape de re-échantillonnage.

En fin, on recalcule les valeurs des moyennes et covariance avec chaque nouvelle particule à base de la mesure d'observation.

Un problème de méthode de particule est que le filtreur de particule dessine exponentiellement à l'échelle de dimension d'espace d'état. La dimension d'espace est acceptable environ 3 ou 4.

FastSLAM est une des algorithmes les plus faciles à implémenter.

### Des propriétés de FastSLAM.

- Le FastSLAM résout un problème plein et en-ligne de SLAM. Chaque particule a un échantillonnage de trajectoire entière de robot. Les équations de mise à jour sont actuellement pour les poses les plus récentes. Donc, le FastSLAM ressemble un filtreur, similaire EKF.

- Le FastSLAM est facile à travailler avec plusieurs hypothèses d'association de données. On prend une décision d'association de donnée à base de chaque particule. On n'utilise pas une seule association de données pour un filtreur entière. FastSLAM peut échantillonner les corrects postérieurs que deux types d'algorithmes avant ne peuvent pas demander.

- Le FastSLAM est efficacement à implémenter. On utilise une méthode d'arbre avancé pour représenter l'estimation de carte. Le temps de mettre à jours est un logarithme de la taille de carte et un linéaire de nombre de particule.

Rao-Blackwellized (Rao-Blackwellized Particle Filter – RBPF) ([17],[18]) est un variant de filtreur de particule. Le RBPF peut conduire un résultat plus correct que « Kalma Filter » et « Hidden Markov Modele Filter ».

## La table de comparaison de ces méthodes

	(R1)			(R2)	(R3)		
	UDA	nonlinear	quality	memory	update	global update	loop
Maximum Likelihood		✓	✓	$m$	$(n + p)^3$		
EKF			✓	$n^2$	$n^2$		
CEKF			✓	$n^{\frac{3}{2}}$	$k^2$	$kn^{\frac{3}{2}}$	
Relaxation		✓	✓	$kn$	$kn$		$kn^2$
FastSLAM	✓	✓	?	$Mn$	$M \log n$		
SEIF			?	$kn$	$k^2$		
TJTF			✓	$k^2n$	$k^3$	$k^3n$	
Treemap		✓	✓	$kn$	$k^2$	$k^3 \log n$ resp. $kn$	
Multilevel Relaxation		✓	✓	$kn$	$kn$		

R1, R2, R3 sont trois exigences du problème SLAM. R1: l'incertitude est bordé par un seuil. R2: l'espace de stockage est linéaire avec la taille de carte. R3: la mise à jour est linéaire avec la taille de carte. C'est la table de performance de l'algorithme de SLAM avec  $n$  points de repères,  $m$  mesures,  $p$  poses de robot,  $k$  points de repère sur une image (point d'intérêt). UDA signifie « uncertain data association ». Cette table est extraite de [13].

### SLAM visuel

Le robot utilise beaucoup de moyens de percevoir l'environnement tels que le laser, l'infrarouge, la caméra. La caméra n'est pas considérée comme une centre des recherches de SLAM. Malgré que la caméra est compacte, bien-compris, bon marché, et ubiquitaire. Mais, il est difficile à construire une carte dans une longue terme en temps réel à partir de point d'intérêts extraits de l'image car la rate de données d'une caméra est beaucoup plus haute que celle de l'autre senseur. Dans les années récentes, des recherches sur SLAM visuel donnent beaucoup d'approches. Les approches donnés sont souvent basées sur le modèle de calcul de la probabilité mentionnée avant en combinaison avec des approches de l'observation par caméra.

### Odométrie visuelle

C'est une méthode de localisation avec l'entrée étant une seule série des images ([20],[32],[33]). La base de cette méthode est de traquer des points d'intérêts. Des points d'intérêts sont faits la correspondance entre deux images et liés dans tout le processus pour construire une trajectoire de robot. Ensuite, l'estimation de motion de robot est réalisée par l'utilisation d'une architecture géométrique hypothèse-et-test.

Dans chaque image, on détecte des Harry coins [21] comme point d'intérêt. Les coins sont considéré stable avec le problème de déformation base. En contraste à traqueur de KLT [22], on détecte des points d'intérêts dans tous les frames, et fait seulement la correspondance entre des points d'intérêts. Dans le processus d'estimation de motion, on peut réaliser une dans deux implémentations sur le monocular ou le stéréo.

#### Implémentation monoculaire

1. Traquer des points d'intérêts sur certaines cadres. Estimer des poses de robot entre 3 cadres en utilisant l'algorithme « 5-points » [23] et « preemptive RANSAC » [24].
2. Trianguler les points d'intérêts à points 3D en utilisant le premier et dernier observation de chaque traque.
3. Traquer certaines images additionnelles. Calculer la pose de camera basée sur des points 3D connus en utilisant l'algorithme « 3-points » [25] et « preemptive RANSAC ».
4. Re-trianguler des points 3D en utilisant les observations première et dernière dans le traque du point. Répéter l'étape 3 sous certaines fois.
5. Répéter de l'étape 1 à l'étape 4 sous certaines fois.
6. Insérer la valeur de « pare-feu » et répéter de l'étape 1 sous certaines fois.

#### Stéréoscopique implémentation

1. Faire la correspondance entre les images gauche et droites de la paire de stéréo. Trianguler des correspondances à des points 3D.
2. Traquer les points sur certaines cadres. Calculer la pose en utilisant « preemptive RANSAC ». L'algorithme « 3-point » (considéré dans les images gauches) est utilisé comme un générateur des hypothèses.
3. Répéter l'étape 2 en certaines fois.
4. Trianguler tous les correspondances des nouveaux points d'intérêts en utilisant des observations des images gauche et droite. Répéter l'étape 2.

## 5. Re-trianguler tous les points 3D pour construire un pare-feu. Répéter de l'étape 2.

On utilise le pare-feu pour restreindre l'accumulation et la propagation de l'erreur en impliquant simplement que la triangulation n'est pas réalisée si l'observation utilisée dans cette triangulation est plus loin que la valeur de pare-feu récent. On a indiqué que l'odométrie visuelle peut être comparable avec « Differential Global Positioning System (DGPS) » et « Inertial Navigation System (INS) ».

### Gamma-SLAM

On utilise l'odométrie visuelle et un filtreur de particule RaoBlackwellized. On utilise une distribution gamma pour représenter les hauteurs des objets de l'environnement. Donc, cet algorithme s'appelle Gamma-SLAM [26]. On maintient une distribution postérieure sur la variance de hauteur dans chaque cellule. Certains problèmes SLAM visuel utilisent la technique de stéréo [27]. La plupart des algorithmes de SLAM visuel en base de stéréo aujourd'hui sont basés des repères.

On peut représenter la carte par une grille. Chaque cellule représente une pose de robot. Dans cet algorithme, on peut considérer une carte comme une grille de la probabilité conditionnelle qui confirme la correction de la carte.

Cette algorithme utilise gamma distribution pour calculer la vraisemblance valeur de l'hauteur des objets dans l'environnement. En comparant avec l'algorithme qui utilise seulement Odométrie Visuelle, le résultat obtenu est moins bon que par utiliser gammaSLAM. Une construction d'une carte avec l'hauteur des objets est bon pour la planification et la navigation de robot dans l'environnement non-structuré.

### Mini SLAM

On a introduit une approche pour un problème de SLAM dans un environnement large avec des exigences de perception et computation minimale [28]. Cette approche basée sur une représentation d'un problème de SLAM par un graphe des poses de robot et des relations entre eux. Pour estimer la covariance pour des liens dans le graphe, miniSLAM est introduit.

- Une carte de problème SLAM est présentée par un graphe. Chaque nœud présente une cadre correspondance une pose de robot et l'estimation de cette pose. L'arc représente une relation entre deux cadres. Cette relation se compose de 2 relations: la similarité visuelle et la similarité odométrique. La sortie est un vecteur d'estimation du maximum de vraisemblance pour tous les poses de cadres.

Différence avec l'autre SLAM:

N'utilise pas d'estimation de position des points de repères relatifs. Donc, il est facile à l'appliquer à un environnement grande. Il utilise une estimation des différences entre des points de repères entre des images séquentielles et l'incertitude de cette estimation. Des différences considères sont des relations visuelles et des relations odométriques. Il utilise l'algorithme « Multilevel Relaxation » pour l'estimation du maximum de vraisemblance.

Les auteurs ont réussi avec miniSLAM dans un pratique avec des images omni-dimensionnelles.

C'est un point attentif. Parce qu'il y a une différence entre le changement de le nombre de similaires d'image omni-dimensionnelle et celui d'image non omni-dimensionnelle. Donc, je pense que l'on ne peut pas appliquer ce calcul pour le robot ayant un caméra non omni-dimensionnelle avec un champ de vue petit.

### $\sigma$ SLAM

$\sigma$ SLAM,  $\sigma$  signifie stéréo, utilise une estimation stéréoscopique pour estimer la position de repère [29]. Et une estimation de motion est conduite par un flux optique clairsemé. On distingue des repères par « Scale Invariant Feature Transform » (SIFT). Cela est différent avec des approches courantes qui utilise une odométrie mécanique pour estimer la position de robot et une scan par laser pour distinguer des repères.

Cet algorithme a proposé une « proposal distribution » dans l'approche de « RaoBlackwellized Particle Filter » (RBPF). C'est une mixture entre le modèle de motion et une distribution

On a conduit une carte dense des 3D landmarks identifié par SIFT. L'invariance et la nouvelle « proposal distribution » améliorent le problème de fermer un cycle large (« closing large loop »).

On a indique que cet algorithme est pour un environnement avec la lumière haute (beaucoup de fenêtre en glass). On a indique que l'on peut produire une grille 2D d'occupation. Cette grille est bien pour la planification et la navigation de robot. Une observation est un ensemble des correspondances entre les points de repères de la vue courante et les points de repères stockés dans la carte de robot.

### MonoSLAM

C'est une méthode pour une seule camera incontrôlée. Créer une carte en ligne clairsemée et persistente. C'est à dire d'analyser une séquence des images pour produire une reconstruction de trajectoire de robot et structure de la scène observée. Mais cette méthode est bien pour une séquence courte de l'image. L'auteur concentre à faire la localisation comme un travail principal. La carte est construite mais c'est clairsemé et optimisée dans la processus de localisation.

Cette méthode représente le monde sur une carte 3D probabiliste [34]. Carte représente l'estimation courante de l'état de

caméra et les points d'intérêts et encore l'incertitude de cette estimation. La carte est initialisé au début et mise à jour par la méthode EKF à chaque fois de la motion de robot et son observation.

On a choisi des larges rapiécers de l'image comme les points de repères naturel ([35],[36]). On détecte les patches par utilisation une technique mentionné dans [22]. Dans l'initialisation de SLAM, le robot ne connaît rien dans l'environnement. Mais, dans ce cas, on aide le robot par ajouter une petite information sur l'environnement. C'est souvent 4 points de repères avec leurs positions.

On considère que la caméra se déplace avec une vitesse de motion et motion angulaire constant. On cherche des rapiécers template pour l'estimation de position de l'image [37] en utilisant les faits de correspondance. Pour augmenter la vitesse de recherche, on calcule une prédiction la position de point d'intérêt et l'incertitude de cette prédiction.

D'après l'identification et la mesure première de point d'intérêt, on crée une ligne 3D sur le carte sur laquelle est le point d'intérêt. C'est une ligne à partir de la position estimée vers la direction d'observation de point d'intérêts. Un ensemble d'hypothèses de point d'intérêts est distribué sur cette ligne. En suite, on calcule une approximation de la profond de ce point d'intérêt. En fin, on construire ce point d'intérêt en forme plein avec une représentation Gaussienne.

Gestion de carte est de gérer le nombre de points d'intérêts et déterminer quand on peut ajouter un nouveau point d'intérêt ou supprimer un point d'intérêt de la carte. On essaie de maintenir le nombre de point d'intérêts « visible » est proche une valeur prédéterminée. La visibilité de point d'intérêt est estimé par la distance relative de caméra et ce point, et la position de caméra au temps de l'initialisation de ce point. Le point d'intérêt doit être dans l'image et la caméra doit se placer pas trop loin.

Un point d'intérêt est ajouté si le nombre de point d'intérêts « visible » est plus petit qu'un seuil. Un point d'intérêt est supprimé si le nombre d'essai failure de le détecter et le faire correspondance est plus petit qu'un nombre prédéterminé. L'orientation de chaque point d'intérêt est estimée comme la mention dans [38].

**Table de comparaison entre l'odométrie visuelle et monoSLAM**

	Odométrie Visuelle	MonoSLAM
Modèle de vision	Stéréoscopique, Monoculaire	Monoculaire
Faire la correspondance	Tous les point d'intérêt	Gestion de point d'intérêt sous une gestion de carte
Application	$\sigma$ SLAM, Gamma SLAM	Seulement dans monoSLAM, open source
Point d'intérêt	Coin Harry	Large rapiécer de l'image (11x11)

**Table de comparaison entre 5 approches de SLAM visuel.**

	MiniSLAM	$\sigma$ SLAM	GammaSLAM	MonoSLAM
Modèle de calcul	Multilevel Relaxation	RaoBlackwellized Particle Filter	RaoBlackwellized Particle Filter	EKF
Odométrie	Odométrie mécanique de robot	Odométrie visuelle	Odométrie visuelle	Approche de monoSLAM comme une odometrie visuel
Représentation de carte	Graphe de cadre	Une grille 2D d'occupation	Une grille de la probabilité	Carte 3D probabiliste
Modèle de vision	Stéréoscopique	Stéréoscopique	Stéréoscopique	Monoculaire
Point d'intérêt à traquer	SIFT	SIFT	Coin Harry	Large rapiécer de l'image (11x11)
Caméra	omni-dimentionnelle			Une seule camera incontrôlée.

## SLAM 3D

Beaucoup de recherche pour construire une carte volumétrique 6DoF (6 degree of freedom) [39]. Mais tous les

algorithmes utilisés des scanners laser pour fournir des données 3D. C'est un matériau principal pour construire une carte volumétrique. Une caméra ne suffit pas de la capacité de fournir ce type de données. Donc, on n'utilise pas une caméra comme un senseur pour le problème de SLAM 3D.

## Conclusion

Ce rapport est pour l'état de l'art de SLAM en concentrant à SLAM visuel. Plusieurs de l'amélioration de la solution de problème de SLAM dans les années récentes. En particulier, la tendance de l'algorithme de SLAM visuel est de concentrer à utiliser une seule caméra pour résoudre problème de SLAM.

## Référence

1. R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. Springer Verlag, New York, 1988, pp. 167 – 193.
2. S. Thrun, "Robotics mapping: A survey," School of Computer Science, Carnegie Mellon University, Tech. Rep., Feb. 2002.
3. J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Transactions Robotics and Automation*, vol. 17, no. 3, pp. 242 – 257, 2001.
4. J. Guivant and E. Nebot, "Solving computational and memory requirements of feature based simultaneous localization and map building algorithms," Australian Centre for Field Robotics, University of Sydney, Sydney, Tech. Rep., 2002.
5. F. Dellaert. Square Root SAM: Simultaneous location and mapping via square root information smoothing. In *Robotics: Science and Systems (RSS)*, 2005.
6. S. Thrun, D. Koller, Z. Ghahmarani, and H. Durrant-Whyte, "SLAM updates require constant time," School of Computer Science, Carnegie Mellon University, Pittsburgh, Tech. Rep., 2002.
7. Mark A. Paskin, "Thin junction tree lters for simultaneous localization and mapping," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, G. Gottlob and T. Walsh, Eds., San Francisco, CA, 2003, pp. 1157–1164.
8. S. Thrun, "Probabilistic algorithms in robotics," *AI Magazine*, vol. 21, no. 4, pp. 93–109, 2000.
9. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
10. G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.
11. T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, 2000, pp. 3841–3846.
12. T. Duckett, S. Marsland, and J. Shapiro, "Fast, on-line learning of globally consistent maps," *Autonomous Robots*, vol. 12, no. 3, pp. 287– 300, 2002.
13. U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localisation and mapping," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 196–207, April 2005.
14. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle ltering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the Eighteenth Int. Joint Conf. on Artificial Intelligence (IJCAI-03)*. San Francisco,
15. Michael Montemerlo and Sebastian Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Proc. ICRA*, 2003.
16. Dirk Haehnel, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *IROS*, 2003.
17. Pantelis Elinas, Robert Sim, and James J. Little. SLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution. In *Proc. 2006 IEEE ICRA*, 2006.
18. A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic bayesian networks. In *16th Conference on Uncertainty in AI*, pages 176–183, 2000.
19. Stephen Se, Timothy Barfoot, and Piotr Jasiobedzki. Visual motion estimation and terrain modeling for planetary rovers. In *Proc. ISAIRAS 2005*, 2005.
20. A. Levin and R. Szeliski, "Visual odometry and map correlation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, Washington, DC, USA, 2004.
21. C. Harris and M. Stephens, "A Combined Corner and Edge Detector, Proc". Fourth Alvey Vision Conference, pp.147-151,1988.
22. J. Shi and C. Tomasi, Good Features to Track, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.
23. D. Nist'er. An Efficient Solution to the Five-Point Relative Pose Problem, *IEEE Conference on Computer Vision and Pattern Recognition*, Volume 2, pp. 195-202, 2003.
24. D. Nist'er. Preemptive RANSAC for Live Structure and Motion Estimation, *IEEE International Conference on Computer Vision*, pp. 199-206, 2003.
25. R. Haralick, C. Lee, K. Ottenberg and M. N'olle, Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem, *International Journal of Computer Vision*,
26. Tim K. Marks, Andrew Howard, Max Bajracharya, Garrison W. Cottrell, and Larry Matthies. "Gamma-SLAM: Stereo Visual SLAM in Unstructured Environments Using Variance Grid Maps". 2007.

27. M. Dailey and M. Parnichkun. Simultaneous localization and mapping with stereo vision. In Proc. ICARCV, 2006.
28. Henrik Andreasson, Tom Duckett and Achim Lilienthal, "Mini-SLAM: Minimalistic Visual SLAM in Large-Scale Environments Based on a New Interpretation of Image Similarity" in Robotics and Automation, 2007 IEEE International Conference. Apr, 2007
29. Pantelis Elinas, Robert Sim, and James J. Little.  $\sigma$ SLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution. In Proc. 2006 IEEE ICRA, 2006.
30. N. Karlsson, E. D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," in IEEE Int. Conf. on Robotics and Automation (ICRA), Barcelona, Spain, April 2005, pp. 24–29.
31. A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in Proc. International Conference on Computer Vision, Nice, Oct. 2003.
32. D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), 2004, pp. 652–659.
33. J. Campbell, R. Sukthankar, and I. Nourbakhsh, "Techniques for evaluating optical flow for visual odometry in extreme terrain," in IEEE/RSJ Int. Workshop on Robots and Systems (IROS-04), Sendai, Japan, October 2004.
34. R. Smith, M. Self, and P. Cheeseman, "A Stochastic Map for Uncertain Spatial Relationships," Proc. Fourth Int'l Symp. Robotics Research, 1987.
35. A.J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," Proc. Ninth Int'l Conf. Computer Vision, 2003.
36. N.D. Molton, A.J. Davison, and I.D. Reid, "Locally Planar Patch Features for Real-Time Structure from Motion," Proc. 15th British Machine Vision Conf., 2004.
37. A.J. Davison, "Active Search for Real-Time Vision," Proc. 10th Int'l Conf. Computer Vision, 2005.
38. H. Jin, P. Favaro, and S. Soatto, "A Semi-Direct Approach to Structure from Motion," The Visual Computer, vol. 19, no. 6, pp. 377-394, 2003.
39. Andreas Nüchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann, "6D SLAM - 3D Mapping Outdoor Environments" Journal of Field Robotics (JFR), Special Issue on Quantitative Performance Evaluation of Robotic and Intelligent Systems, Wiley & Son, ISSN 1556-4959, Volume 24, Issue 8-9, pages 699 - 722, August - September, 2007.