

Sélection d’ancres pour localisation en intérieur par réseaux radios UWB

Antoine COURTAY, Mickaël LE GENTIL, Olivier BERDER, Pascal SCALART, Sébastien FONTAINE, Arnaud CARER

Université de Rennes 1 - IRISA
Enssat, 6 rue de Kerampont
22305 Lannion
Email: prénom.nom@irisa.fr

Résumé – La localisation en intérieur demeure un grand défi pour les applications comme le suivi sportif, les visites contextuelles (musées), la modélisation de bâtiments (*BIM*) ou les missions de drones automatisées. Les approches classiques utilisant des communications radio comme le *WiFi*, le *Bluetooth* ou le *ZigBee* ne donnent qu’une précision supérieure à 2 mètres lorsque le mobile à localiser est statique, les résultats se dégradant en mouvement. Basé sur la technologie *UWB* (*Ultra Wide Band*), un algorithme de localisation nommé *Meilleure Sélection d’Ancre pour la Trilatération* (*BAST*) en fonction de la prédiction de position et prenant en compte l’estimation du bruit est proposé. Un protocole expérimental utilisant la plateforme *Zyggie* de l’*IRISA* montre que *BAST* permet d’obtenir des résultats de 1,26 jusqu’à 4,17 fois meilleurs en termes de précision comparé aux algorithmes de faible complexité de l’état de l’art lorsque le mobile/personne est en mouvement (joueur de tennis par exemple).

Abstract – Indoor localization still remains a great challenge for applications like sports monitoring, contextual visits of museum, *Building Information Modeling* (*BIM*) or automated drone missions. Classical approaches using radio communication such as *WiFi*, *Bluetooth*, *ZigBee* only give a precision of more than 2 meters when the mobile is static, of course worse when moving. Based on *Ultra Wide Band* (*UWB*) communications, a localization algorithm named *Best Anchor Selection for Trilateration* (*BAST*) that outperforms low complexity state of the art algorithms is proposed. The anchor selection relies on position prediction and takes into account noise estimation. A wearable, light and low power *Wireless Sensor Network* (*WSN*) prototype (named *Zyggie*) including an *UWB* chipset has been developed for algorithms comparison. Experimental testbed using real cases experiments show that *BAST* can give from 1.26 up to 4.17 times better precision when the mobile/person is in movement (e.g. tennis player).

1 Introduction

Connaître la position d’un dispositif dans son environnement est une question importante pour les applications offrant des services basés sur la localisation. En environnement intérieur, certaines approches utilisent des signaux optiques tels que l’infrarouge, le laser ou le suivi vidéo (comme le système *Vicon* [1]). Les applications que nous ciblons nécessitent un système portable, faible coût et faible consommation pouvant être installé en quelques minutes. Sous ces contraintes, le système de localisation basé sur les signaux radio est le meilleur candidat. Ces dernières années, de nombreux standards radio pour la localisation ont été testés comme le *WiFi*, le *Bluetooth* ou le *ZigBee*. Ces derniers fournissent une information de localisation en se basant sur la force du signal reçu (*Received Signal Strength Indicator* : *RSSI*). Les performances sont souvent décevantes en raison du manque de précision (environ 2,5 mètres) et se détériorent encore plus lorsque le mobile est en mouvement, n’est pas en ligne de vue (*Non Line Of Sight* : *NLOS*), ou s’il y a des trajets multiples (murs, objets). Une nouvelle approche radio connue sous le nom

Ultra Large Bande (*UWB*) ayant la capacité de faire des mesures précises de temps de vol a récemment émergé. Depuis 2014, *Microsoft* a lancé une compétition de localisation en intérieur conjointement à la conférence *IPSN* [2]. Plus de la moitié des concurrents utilisent l’*UWB* et 60% d’entre eux sont dans les 10 meilleurs. Cependant, les mesures sont statiques et la plupart des prototypes présentés ne sont pas encore assez miniaturisés pour être portables. De plus, certaines solutions utilisent des plateformes différentes pour les ancres ou des ancres filaires pour synchroniser le réseau. Contrairement aux solutions mentionnées ci-dessus, notre nœud de capteur sans fil *Zyggie* est ultra compact et très léger (32 g) ce qui en fait un bon candidat pour toutes les applications où le poids et la taille sont critiques. Les ancres et le mobile ont la même architecture et la synchronisation du réseau est faite sans fil via une interface radio 2,4 GHz. La principale contribution de ce papier est le développement d’un algorithme de localisation respectant faible complexité et faible empreinte mémoire (imposées par les nœuds de capteur sans fil) nommé *Meilleure Sélection d’Ancre pour la Trilatération* (*BAST*). *BAST* permet également l’affichage en

temps réel de la position calculée sur des périphériques portables (smartphone, tablette). Une dernière contribution est que tous les résultats ont été obtenus à l'aide de cas d'expérimentation réels. Une trajectoire de marche en serpentín et une partie de tennis ont été enregistrées à plusieurs reprises pour évaluer les performances des algorithmes.

Le reste du papier est organisé comme suit. La Section 2 présente les algorithmes de l'état de l'art pour la localisation basée sur des ancrés. Notre nouvel algorithme (*BAST*) pour la localisation en intérieur d'un mobile est présenté en Section 3. Il est ensuite validé expérimentalement en Section 4 et comparé à d'autres algorithmes de localisation, avant de conclure.

2 Algorithmes de localisation

Un algorithme de localisation a pour but de fournir des coordonnées 2D/3D en combinant des distances d'une cible (éventuellement mobile) à des ancrés dont la position est supposée connue. 3 ancrés minimum sont nécessaires mais plus il y a d'ancrés, plus le système d'équations sera important et potentiellement meilleure sera la précision. Ce système peut être décrit pour 3 ancrés de la façon suivante :

$$\begin{aligned} d_{A1}^2 &= (x_M - x_{A1})^2 + (y_M - y_{A1})^2 + n_{A1} \\ d_{A2}^2 &= (x_M - x_{A2})^2 + (y_M - y_{A2})^2 + n_{A2} \\ d_{A3}^2 &= (x_M - x_{A3})^2 + (y_M - y_{A3})^2 + n_{A3} \end{aligned} \quad (1)$$

où les d_{Ai} sont les distances du mobile à l'ancre i , x_{Ai} et y_{Ai} sont les coordonnées de l'ancre i , n_{Ai} le bruit sur les distances mesurées et x_M , y_M les coordonnées du mobile. Ce système est la base pour la plupart des algorithmes de localisation. Parmi les plus courants on peut citer le barycentre pondéré (*WCL* pour *Weighted Centroid Localization*) décrit dans [3] qui consiste à trouver la solution \mathbf{m} de Eq. (2) :

$$\mathbf{m} = \begin{pmatrix} x_M \\ y_M \end{pmatrix} = \sum_{i=1}^N w_{Ai} \cdot \mathbf{a}_{Ai} = \sum_{i=1}^N \frac{\frac{1}{d_{Ai}} \cdot \begin{pmatrix} x_{Ai} \\ y_{Ai} \end{pmatrix}}{\sum_{j=1}^N \frac{1}{d_{Aj}}} \quad (2)$$

où \mathbf{a}_{Ai} représente les vecteurs de coordonnées des N ancrés et w_{Ai} les coefficients de pondération de chaque ancre.

La méthode *Gauss-Newton* [4] est un algorithme itératif visant à minimiser l'erreur de positionnement, en résolvant :

$$\mathbf{m}_l = \mathbf{m}_{l-1} - \mathbf{H}_{l-1}^{-1} \cdot \mathbf{g}_{l-1} = \begin{pmatrix} x_M \\ y_M \end{pmatrix} \quad (3)$$

avec l l'indice d'itération, \mathbf{H} la matrice hessienne et \mathbf{g} le vecteur de gradient. Cet algorithme a l'avantage de converger rapidement au prix d'une complexité plus grande due à la matrice hessienne.

2.1 Trilatération

L'algorithme de trilatération consiste à trouver une solution au système défini par (1). Après quelques transformations, il est possible d'écrire le système à résoudre comme $\mathbf{u} = \mathbf{V} \cdot \mathbf{m} + \mathbf{e}$ avec :

$$\begin{aligned} \mathbf{u} &= \begin{pmatrix} (d_{A1}^2 - d_{A2}^2) - (x_{A1}^2 - x_{A2}^2) - (y_{A1}^2 - y_{A2}^2) \\ (d_{A1}^2 - d_{A3}^2) - (x_{A1}^2 - x_{A3}^2) - (y_{A1}^2 - y_{A3}^2) \end{pmatrix}, \\ \mathbf{V} &= \begin{pmatrix} 2 \cdot (x_{A2} - x_{A1}) & 2 \cdot (y_{A2} - y_{A1}) \\ 2 \cdot (x_{A3} - x_{A1}) & 2 \cdot (y_{A3} - y_{A1}) \end{pmatrix} \text{ et} \\ \mathbf{e} &= \begin{pmatrix} n_{A1} - n_{A2} \\ n_{A1} - n_{A3} \end{pmatrix} \end{aligned} \quad (4)$$

Pour cet exemple l'ancre numéro 1 a été choisie comme référence et apparaît donc deux fois dans les équations. En posant :

$$\mathbf{v}_1 = \begin{pmatrix} 2 \cdot (x_{A2} - x_{A1}) \\ 2 \cdot (x_{A3} - x_{A1}) \end{pmatrix} \text{ et } \mathbf{v}_2 = \begin{pmatrix} 2 \cdot (y_{A2} - y_{A1}) \\ 2 \cdot (y_{A3} - y_{A1}) \end{pmatrix} \quad (5)$$

Les règles de *Cramer* donnent une solution qui minimise l'erreur quadratique moyenne (*MSE* pour *Mean Square Error*) pour obtenir :

$$x_M = \frac{\det(\mathbf{u} \quad \mathbf{v}_2)}{\det(\mathbf{V})} \text{ and } y_M = \frac{\det(\mathbf{v}_1 \quad \mathbf{u})}{\det(\mathbf{V})} \quad (6)$$

Les ancrés ne doivent pas être alignés pour éviter un dénominateur nul. Il existe d'autres voies mathématiques pour résoudre le système comme les moindres carrés (*LS* pour *Least Squares*) ou la borne de *Cramer Rao* (*CRB* pour *Cramer-Rao Bound*) [5], mais comme nous visons une implémentation efficace en énergie, seule la méthode de *Cramer* a été impléentée.

2.2 Multilatération

La multilatération [6] utilise les développements vus précédemment pour la trilatération et les applique à toutes les combinaisons de 3 ancrés non-alignés. Pour N ancrés, on obtient C_3^N combinaisons, et la solution est la moyenne de ces points. Malgré la précision promise, le point faible de cette approche est la complexité, qui augmente avec le nombre d'ancrés.

Il existe d'autres variantes de trilatération/multilatération mais elles sont soit trop complexes, comme [7] qui utilise la logique floue (mais a besoin de phases d'apprentissages, de filtres récursifs avec mise à jour des coefficients) ou [8] basées sur le filtrage bayésien (couplé à une estimation par maximum de vraisemblance), soit trop coûteuses, comme *DV-Hop* qui propose d'utiliser 200 ancrés réparties aléatoirement sur 100 m^2 pour obtenir seulement 0,85 mètres de précision [9].

Algorithm 1 Algorithme BAST

Entrée : $\mathbf{d}_A, \mathbf{a}_A, \mathbf{m}_{k-1}, \mathbf{m}_{k-2}$ **Sortie :** \mathbf{m}_k

1: $d_A \leftarrow Filter(d_A)$	▷ Filtrage des distances éronnées
2: $\hat{\mathbf{m}} \leftarrow Predictor(\mathbf{m}_{k-1}, \mathbf{m}_{k-2})$	▷ Estimation des coordonnées du mobile
3: for $i \leftarrow 1, NbAnchors$ do	
4: $A_1 \leftarrow \{d_{A_i}; \mathbf{a}_{A_i}\}$	▷ Création de la référence
5: $(A_2, A_3) \leftarrow NearestNonAlignedAnchors(\mathbf{d}_A, \mathbf{a}_A)$	▷ Sélection des 2 ancrs non-alignées les plus proches
6: $\mathbf{m}_i \leftarrow Trilateration(A_1, A_2, A_3)$	▷ Calcul de trilatération
7: $\delta_{MA_i} \leftarrow EuclidianDistance(\mathbf{m}_i, \hat{\mathbf{m}})$	▷ Calcul de ED
8: end for	
9: $\mathbf{m}_{min} \leftarrow GetMobile(min(\delta_{MA}))$	▷ Choix des coordonnées minimisant ED
10: $(A_1, A_2, A_3) \leftarrow GetTriplet(min(\delta_{MA}))$	▷ Triplet d'ancres minimisant ED
11: $\mathbf{n} \leftarrow Noise(A_1, A_2, A_3, \mathbf{m}_{min})$	▷ Calcul du bruit
12: $\mathbf{d}_{ATriplet} \leftarrow Distances(A_1, A_2, A_3) - \mathbf{n}$	▷ Soustraction du bruit
13: $(A_1, A_2, A_3) \leftarrow UpdateDistances(\mathbf{d}_{ATriplet})$	▷ Mise à jour du triplet
14: $\mathbf{m}_k \leftarrow Trilateration(A_1, A_2, A_3)$	▷ Trilatération finale

3 Meilleure Sélection d'Ancre pour la Trilatération (*BAST*)

L'algorithme *BAST* sélectionne un triplet d'ancres en prenant en compte la prédiction de la position du mobile, et le bruit de mesure. Ces différentes étapes sont décrites dans Alg. 1 et résumées ci-dessous.

Premièrement, un filtrage médian va remplacer les valeurs éronnées du vecteur d_A dues à des mesures de temps de vol impossibles ou perturbées par le transceiver *UWB*. Ensuite les coordonnées du mobile sont estimées grace à un prédicteur linéaire (en assumant une vitesse constante) par l'Eq. (7).

$$\hat{\mathbf{m}} = 2 \times \mathbf{m}_{k-1} - \mathbf{m}_{k-2} \quad (7)$$

où \mathbf{m}_{k-1} et \mathbf{m}_{k-2} sont les deux dernières coordonnées calculées par *BAST*.

L'objectif de la boucle *for* est alors de calculer la *Distance Euclidienne* (*Euclidian Distance* : ED) entre \mathbf{m}_i et $\hat{\mathbf{m}}$ pour $NbAnchors$ triplets. Le triplet correspondant à l' ED minimale δ_{MA} est alors sélectionné pour l'estimation du bruit \mathbf{n} par (8).

$$\mathbf{n} = \begin{pmatrix} d_{A1} - \sqrt{(x_{\mathbf{m}_{min}} - x_{A1})^2 + (y_{\mathbf{m}_{min}} - y_{A1})^2} \\ d_{A2} - \sqrt{(x_{\mathbf{m}_{min}} - x_{A2})^2 + (y_{\mathbf{m}_{min}} - y_{A2})^2} \\ d_{A3} - \sqrt{(x_{\mathbf{m}_{min}} - x_{A3})^2 + (y_{\mathbf{m}_{min}} - y_{A3})^2} \end{pmatrix} \quad (8)$$

où \mathbf{m}_{min} correspond aux coordonnées du mobile avec l' ED minimale. Le bruit est finalement soustrait au vecteur des distances et une dernière trilatération permet d'obtenir les coordonnées du mobile \mathbf{m}_k .

4 Résultats expérimentaux

Pour valider cet algorithme et le comparer à l'état de l'art, des campagnes de mesures ont été menées grâce à

la plateforme de réseaux corporels *Zyggie* développée par l'*IRISA*. Ultra compacte (63 mm par 37 mm) et légère (32 g avec batterie) et par conséquent idéale pour toutes applications où poids et taille sont critiques, elle se compose comme suit :

- un micro-contrôleur *ARM CortexM4* qui centralise les traitements du nœud et contrôle le composant radio *UWB*,
- un micro-contrôleur *ATMEGA2564RFR2* qui gère la radio 2,4 GHz pour les échanges de données et la synchronisation du réseau,
- un composant radio *DecaWave DW1000 UWB* [10] pour les mesures de temps de vol,
- une centrale inertielle (*InvenSense Inertial Measurement Unit* : *IMU*).

Deux scénarios différents ont été établis pour valider notre proposition et, ont été rejoués 10 fois.

1. Marche en serpent in dans une grande pièce (26,4 m par 9,2 m) avec 6 ancrs
2. Match de tennis avec comparaison *Vicon* (système de vision haute précision) pour estimer les erreurs

La figure 1 permet de valider qualitativement la pertinence de notre algorithme. Elle représente la trajectoire estimée du mobile par *BAST* dans le contexte du serpent in, et on peut y constater que l'algorithme est très peu mis en défaut, même lors de changements brusques de trajectoire. Une telle représentation n'est pas possible pour le tennis, où les déplacements sont bien plus confus, mais une analyse comparative quantitative est en revanche possible.

Les tables 1 et 2 présentent les performances des algorithmes précédemment définis en terme d'erreur moyenne ϵ et de gain apporté par *BAST* $\Delta\epsilon$, d'écart-type σ et de gain associé $\Delta\sigma$, et le temps de calcul T_{comp} . Les indices *S* et *T* indiquent serpent in et tennis, respectivement.

L'algorithme *WCL* est le moins bon, car toutes les ancrs essaient d'attirer la solution vers elles, ce qui fait que les

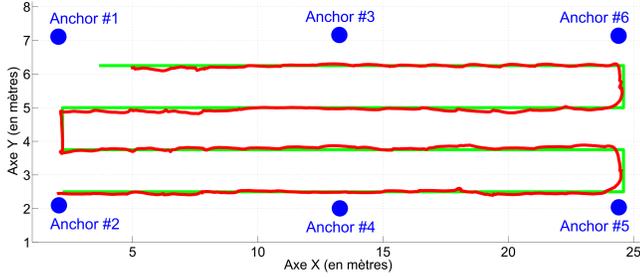


FIGURE 1 – Marche en serpentin pour *BAST* : ancres en bleu, trajectoire du mobile en rouge et idéale en vert

TABLE 1 – Résultats pour marche en serpentin

	WCL	Gauss-Newton	Trilatération	Multilatération	<i>BAST</i>
ϵ_S (m)	1,15	0,08	0,09	0,17	0,06
σ_S (m)	0,92	0,07	0,18	0,19	0,06
$\Delta\epsilon_S$	$\times 19,1$	$\times 1,33$	$\times 1,50$	$\times 2,83$	
$\Delta\sigma_S$	$\times 15,3$	$\times 1,17$	$\times 1,14$	$\times 3,17$	
T_{compS} (ms)	0,26	0,32	0,26	0,32	0,59

coordonnées estimées ne sont pas sur la trajectoire réelle. Dans les deux contextes, la trilatération fonctionne correctement, mais l’erreur peut vite augmenter si le mobile n’est pas dans le triangle virtuel formé par les ancres sélectionnées. Comme la multilatération est obtenue par la moyenne de toutes les trilatérations, le phénomène précédent est également pénalisant. *Gauss-Newton* donne des résultats quasiment identiques à *BAST* pour le contexte du serpentin. En revanche, lorsque les mouvements sont plus rapides, comme pour le tennis, *BAST* supplante tous les autres algorithmes en apportant une amélioration entre 1,26 et 4,17, au prix d’un temps de calcul doublé, mais encore largement acceptable pour des exécutions temps réel.

TABLE 2 – Résultats pour le contexte tennis

	WCL	Gauss-Newton	Trilatération	Multilatération	<i>BAST</i>
ϵ_T (m)	4,50	1,51	2,11	1,36	1,08
σ_T (m)	0,96	1,72	1,68	0,98	0,76
$\Delta\epsilon_T$	$\times 4,17$	$\times 1,40$	$\times 1,95$	$\times 1,26$	
$\Delta\sigma_T$	$\times 1,26$	$\times 2,26$	$\times 2,21$	$\times 1,29$	
T_{compT} (ms)	0,26	0,32	0,26	0,32	0,61

5 Conclusion

Nous avons présenté un nouvel algorithme de localisation en intérieur précis et facilement implémentable sur des nœuds de capteurs. Cet algorithme se base sur une estimation du temps de vol par des transceivers *UWB*, et utilise la prédiction de position pour sélectionner les ancres nécessaires à la trilatération. Grâce à la plateforme *Zyggie* des campagnes de mesures ont pu démontrer toute la pertinence de notre approche dans deux contextes différents, avec des gains de précision entre 1,26 et 4,17.

Références

- [1] “Vicon system.” [Online]. Available : <https://www.vicon.com/>
- [2] “Microsoft indoor localization competition - ipsn 2016.” [Online]. Available : <https://www.microsoft.com/en-us/research/event/microsoft-indoor-localization-competition-ipsn-2016/>
- [3] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, “Weighted centroid localization in zigbee-based sensor networks,” in *IEEE International Symposium on Intelligent Signal Processing (WISP)*, Oct 2007, pp. 1–6.
- [4] J. Nocedal and S. Wright, *Numerical Optimization : 10.3 Algorithms for Nonlinear Least-Squares Problems*. Springer New York, 2006.
- [5] B. Yang and J. Scheuing, “Cramer-rao bound and optimum sensor array for source localization from time differences of arrival,” in *the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, March 2005, pp. 961–964.
- [6] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *the 9th International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2003, pp. 81–95.
- [7] H. Chenji and R. Stoleru, “Toward accurate mobile sensor network localization in noisy environments,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 6, pp. 1094–1106, June 2013.
- [8] D. Dardari, P. Closas, and P. M. Djuric, “Indoor tracking : Theory, methods, and technologies,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, April 2015.
- [9] C. Jo and C. Lee, “Multilateration method based on the variance of estimated distance in range-free localisation,” *Electronics Letters*, vol. 52, no. 12, pp. 1078–1080, 2016.
- [10] “Decawave uwb chipset.” [Online]. Available : <http://www.decawave.com/products/dw1000>