



DÉTECTION AUTOMATIQUE DE LA POSITION D'UN USAGER
DANS UN BÂTIMENT
RAPPORT 2

Formalisation du problème et
Proposition d'algorithme

Auteur : Mbe Mbe Mindjana LOIC HENRI
Email : henrimbemindjana@gmail.com

30 décembre 2023

Table des matières

1	Problème	2
2	Analyse et Formalisation	2
3	Approche par similarité	2
4	Approche par intelligence artificielle	3

1 Problème

On a un bâtiment appartenant à un particulier, dans lequel on y a installé des points d'accès à des positions précises, pour des raisons diverses (accessibilité au réseau bâtiment par exemple). Ce bâtiment possède des zones (salles, pièces, couloir) où des usagers peuvent avoir accès. Comment peut-on faire pour qu'un usager puisse connaître, où il se trouve dans le bâtiment pour facilement s'y orienter, en se basant sur la force des signaux des points d'accès qu'il capte sur son téléphone ou son ordinateur ?

Pour répondre à cette question, on pourrait s'appuyer sur les positions exactes des points d'accès, néanmoins seul le particulier les détient. Cela limite le développement d'une solution car il faudra signer un partenariat avec le particulier pour avoir les positions des points d'accès, ce qui n'est pas forcément souhaité.

De ce fait, **comment peut-on détecter la position de l'utilisateur en s'appuyant plutôt sur un ensemble des signaux détectés à des positions précises du bâtiment ?**

2 Analyse et Formalisation

Considérons qu'un signal est le couple du point d'accès et de la force du signal et qu'on a repère orthonormé pour le bâtiment, Soient :

- M la position de l'utilisateur du bâtiment qu'on cherche
- E : l'ensemble des signaux détectés par l'utilisateur
- $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$: l'ensemble des n positions explorées du bâtiment, P_i étant la i -ème position explorée
- $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$: l'ensemble des n ensembles de signaux détectés aux positions explorées, D_i étant l'ensemble correspondant à la i -ème position explorée

Donner une position approximative M à l'utilisateur en utilisant les signaux collectés revient à :

- Soit trouver les positions où on a collecté les signaux qui ressemblent le plus à ceux perçus par l'utilisateur, puis de les pondérer pour obtenir la position M . La pondération revenant à attribuer plus d'importance aux positions les plus similaires
- Soit utiliser un modèle d'intelligence artificielle (explicable) entraîné sur l'ensemble des signaux collectés, pour prédire la position M sur la base des signaux perçus par l'utilisateur

La première option revient à définir une fonction de similarité Sim qui va permettre d'évaluer la ressemblance entre deux ensembles de signaux, et un seuil ϵ à partir duquel on va considérer que deux ensembles de signaux sont similaires.

La deuxième option revient à choisir un modèle de régression qui va être entraîné sur les données collectées et de l'utiliser pour prédire la position de l'utilisateur sur la base de ses signaux détectés. L'entraînement pourra ainsi être réalisé en amont ou non.

3 Approche par similarité

De la formulation précédente, on a l'algorithme suivant :

Algorithm 1 Algorithme de recherche de position par similarité

Require: l'ensemble des signaux détectés par l'utilisateur E , l'ensemble des n positions explorées du bâtiment \mathcal{P} , l'ensemble des n ensembles de signaux détectés aux positions explorées \mathcal{D} , fonction de similarité Sim , seuil de similarité ϵ

```
1:  $M \leftarrow \vec{0}$  ;
2:  $sum \leftarrow 0$  ;
3: for all  $i$  allant de 1 à  $n$  do
4:    $s \leftarrow Sim(E, D_i)$  ;
5:   if  $s \geq \epsilon$  then
6:      $M \leftarrow M + s \times P_i$  ;
7:      $sum \leftarrow sum + s$  ;
8:   end if
9: end for
10: if  $sum \neq 0$  then
11:   retourner  $M/sum$  ;
12: end if
13: retourner AucunePosition ;
```

La fonction de la similarité pourra être définie en utilisant l'indice de Jaccard et ou le cosinus de similarité qui sont des métriques de similarité très connus.

4 Approche par intelligence artificielle

Pour l'entraînement du modèle, on pourra utiliser un modèle comme XGBoost ; et définir comment transformer les signaux d'une position en vecteur utilisable par ce modèle, qui consiste au prétraitement des données. On utilisera par la suite ce modèle de prétraitement et ce modèle de prédiction pour prédire la position de l'utilisateur.

On obtient de là les algorithmes suivant :

Algorithm 2 Algorithme d'entraînement des modèles

Require: l'ensemble des positions explorées du bâtiment \mathcal{P} , l'ensemble des ensembles de signaux détectés aux positions explorées \mathcal{D} , modèle de prédiction $Model$

```
1:  $PreModel \leftarrow model\_pretraitement(\mathcal{P}, \mathcal{D})$ 
2:  $X, y \leftarrow PreModel(\mathcal{P}, \mathcal{D})$  ;
3:  $Model.train(X, y)$  ;
4: retourner  $Model, PreModel$  ;
```

Algorithm 3 Algorithme de recherche de position par Modèle

Require: l'ensemble des signaux détectés par l'utilisateur E , modèle de prétraitement $PreModel$, Modèle de prédiction $Model$

```
1:  $x \leftarrow PreModel(E)$  ;
2:  $M \leftarrow prediction(Model, x)$  ;
3: retourner  $M$  ;
```
