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

Formalisation du problème et Proposition d'algorithme

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

24 décembre 2023

Table des matières

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 ?

2 Analyse et Formalisation

Pour résoudre le problème il faut fournir un programme à l'usager, qui va lister l'ensemble des points d'accès qu'il atteint, avec les forces des signaux correspondant. Ces informations seront utilisées pour déterminer la position de l'usager et ainsi déterminer la zone du bâtiment dans lequel il se trouve.

Considérons qu'on a repère orthonormé pour le bâtiment, Soient :

— M la position de l'usager du bâtiment qu'on cherche
— $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$: l'ensemble des n points d'accès atteignables dans le repère
— $\mathcal{Z} = \{Z_1, Z_2, \ldots, Z_m\}$: l'ensemble des m zones du bâtiment, décrit par des contraintres sur les coordonnées
— $\mathcal{W} = \{W_1, W_2, \ldots, W_n\}$: l'ensemble des forces de signaux captés par l'appareil de l'utilisateur, où W_i est la force du signal issu du point d'accès A_i
— d : la fonction permettant d'évaluer la distance à laquelle on est d'un point d'accès, en s'appuyant sur la force de son signal capté

Résoudre le problème énoncé revient à trouver le point M tel que :

$$(S) : \forall i \in 1 \ldots n, MA_i{}^2 = d^2(W_i) \tag{1}$$

De là on peut remarquer dans un repère en 2D, que si on a :

— **2 points d'accès** : on aura soit aucune solution, une ou deux solutions selon les cas (obtenables par substitution)
— **3 points d'accès** : on aura soit aucune solution, une solution ou une infinité de solution (obtenables par combinaison linéaire)
— **au delà de 4 points d'accès** : ça devient très compliqué de définir une formule claire, car on peut avoir une infinité de solution

On peut comprendre donc par analogie dans un repère en 3D, qu'il est pratiquement impossible de résoudre S en définissant une formule exacte pour tous les cas, spécialement pour les cas où on a une infinité de solutions.

Ce qui nous amène, à chercher juste une position approchée M de la position exacte tel que :

$$(S') : \forall i \in 1 \ldots n, |MA_i{}^2 - d^2(W_i)| \to 0 \tag{2}$$

Ce qui revient à résoudre le problème d'optimisation suivant :

$$\min_M f(M) = \sum_{i=1}^{n} \left(MA_i{}^2 - d^2(W_i)\right)^2 \tag{3}$$

Pour le résoudre, on peut utiliser l'algorithme de descente de gradient, qui va calculer automatiquement la position du point M cherché. Il suffira de :

2. Définir si on a un repère en 2D ou en 3D
3. Fournir Le gradient $\nabla f(M)$ de f pour la position M

$$\nabla f(M) = 4 \sum_{i=1}^{n} \left(M{A_i}^2 - d^2(W_i) \right) \overrightarrow{A_i M} \tag{4}$$

4. Donner optionnellement une position initiale M_0 du bâtiment où commencer à chercher l'usager

Ayant ainsi la position de l'usagé, il suffira de sélectionner la zone Z_j qui le contient par vérification.

3 Algorithme

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

Algorithm 1 Algorithme de recherche de position

Require: Les positions des points d'accès $\mathcal{A}$, Les zones du bâtiment $\mathcal{Z}$, Les forces des signaux $\mathcal{W}$, la fonction d, position initiale M_0

$f \leftarrow$ **definition_f**($\mathcal{A}$, $\mathcal{Z}$, $\mathcal{W}$, d)
$\nabla f \leftarrow$ **definition_gradf**($\mathcal{A}$, $\mathcal{Z}$, $\mathcal{W}$, d)
$M \leftarrow$ **descente_gradient**$(f, \nabla f, M_0)$;
$j \leftarrow 0$;
while $j \leq m$ et $M \notin Z_j$ **do**
    $j \leftarrow j + 1$;
end while
if $j \leq m$ **then**
    retourner M, Z_j ;
end if
retourner $AucuneZone$;