

Un exemple de filtre discret : le filtre à histogramme

On considère un robot mobile se déplaçant dans une salle dont le sol est composé de dalles de couleurs, et l'on souhaite suivre sa position (x_t, y_t) à chaque instant t . L'espace est discrétisé sous forme d'une grille de dimensions $h * w$. Chaque dalle de la salle possède une couleur que le robot peut détecter grâce à un capteur. Dans un premier temps, le modèle proposé (simple) est le suivant :

- Le robot peut décider de se déplacer d'une case, horizontalement ou verticalement, ou décider de ne pas bouger. Dans le cas où il décide de se déplacer, on a constaté qu'il avait 80% de chance de réussir son déplacement, et sinon restait sur sa dalle de départ. De plus le déplacement échoue systématiquement si le robot décide de se déplacer dans un mur.
- Les capteurs, de bonne qualité, ne sont cependant pas totalement fiables, et ont 10% de chance de détecter une couleur différente que celle de la dalle où se trouve le robot.
- Le robot est placé initialement au hasard (i.e. suivant une loi uniforme sur l'ensemble des dalles) sur une des dalles de la pièce.

Nous nous proposons dans ce TD, à partir du squelette de code fourni, d'utiliser un filtre à histogramme pour évaluer les positions du robot au cours du temps.

- 1) Compléter la fonction *init_positions*. Cette fonction initialise la mesure de probabilité $p(x, y)$ décrivant la position du robot pour l'instant $t = 0$. Vous devriez alors obtenir une image uniforme de la salle, correspondant au fait que chaque position est équiprobable au départ.
- 2) On souhaite maintenant que notre robot se déplace dans la salle. Complétez la fonction *move*, qui met à jour la mesure de probabilité pour chacune des cases, en fonction de la mesure initiale, de la décision de mouvement du robot, et de la probabilité que ce déplacement ait bien réussi. Vous devriez obtenir une image composée uniquement de deux couleurs à la fin de cette étape, pouvez-vous expliquer pourquoi ?
- 3) Pour préciser notre déplacement, nous cherchons donc à prendre en compte les informations en provenance du capteur. Modifiez les valeurs *sensor_right* et *p_move* à 0.8, et complétez la fonction *sense*, afin que celle-ci mette à jour la mesure de probabilité pour chacune des dalles, suivant si la mesure observée correspond ou non à la couleur de la dalle. Vérifiez qu'après cette étape, la somme des probabilités pour l'ensemble des dalles de la salle est bien égale à 1.
- 4) Faites varier les paramètres du modèle (notamment l'incertitude sur le déplacement et la mesure), et observez les résultats. Faites passer *sensor_right* à 1 dans l'exemple fourni, qu'en concluez-vous ?