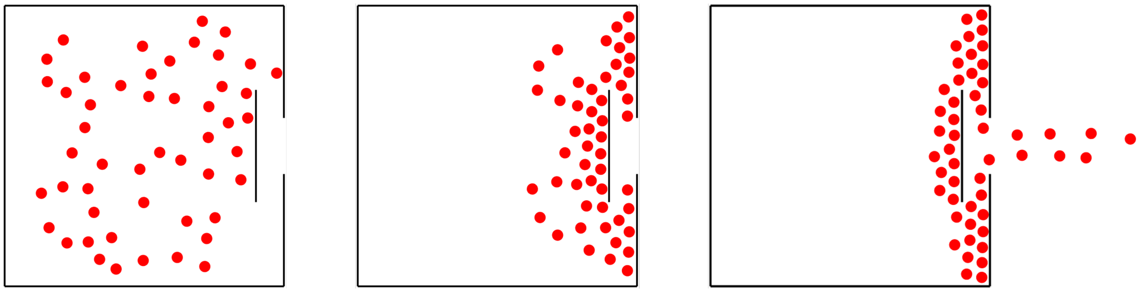
Dynamique de foule

Eric Lunéville

La dynamique de foule s'intéresse aux mouvements d'un ensemble d'individus assez nombreux pour constituer une foule et ce, dans divers contextes : sortie d'une salle, évacuation rapide d'un lieu, attroupement sur un quai de gare, ...



Afin de simuler les mouvements de foules, ils existent des modèles macroscopiques basés sur des systèmes d'équations aux dérivées partielles (Navier-Stokes, Lighthill-Whitham-Richards) ou des modèles microscopiques où le mouvement de chaque individu est considéré. Ces modèles peuvent être basés sur des agents qui se déplacent suivant des règles de comportement ou encore sur des forces "sociales" (attraction, répulsion) qui agissent sur chaque individu, à l'instar du principe fondamental de la dynamique. C'est ce dernier modèle, introduit par Helbing et Molnar, que nous proposons d'utiliser dans ce projet.

1 Modèle de forces sociales

On considère un ensemble d'individus $P_i,\ i=1,N$ constituant une foule située dans un espace bidimensionnel contraint par des murs $M_k,\ k=1,M$. Chaque mur M_k est un segment défini par les deux points extrêmes (M_k^1, M_k^2) auquel on pourra éventuellement adjoindre un vecteur représentant la normale sortante au mur $\mathbf{n}_k$ pour repérer l'intérieur d'une pièce.

A chaque individu P_i sont associés des paramètres :

- sa masse (en kg) m_i (entre 50 et 100 kg)
- le rayon du disque d'occupation r_i (entre 0.2 m et 0.5 m)
- sa vitesse de déplacement désirée w_i (entre 1 et 3 m/s)
- son temps de réaction τ_i (entre 0.3 et 0.5s)
- un point objectif $\mathbf{c}_i$ (en m)

ainsi que des données de calcul de la dynamique :

- sa position instantanée $\mathbf{p}_i$ (en m)
- sa vitesse instantanée $\mathbf{v}_i$ (en m/s)
- la force instantanée $\mathbf{f}_i$ à laquelle il est soumis (en Newton)

Le modèle de force sociale consiste à simuler la dynamique de chaque individu P_i soumis à une force d'attraction (f_i^a reliée à l'objectif) et à des forces de répulsions avec les autres individus (f_{ij}^p) et avec les murs ($f_{i\ell}^m$) :

$$(\mathcal{D}) \qquad \begin{cases} m_i \dfrac{\mathrm{d}\mathbf{v}_i(t)}{\mathrm{d}t} = \mathbf{f}_i^a(t) + \displaystyle\sum_{j\neq i} \mathbf{f}_{ij}^p(t) + \sum_{\ell} \mathbf{f}_{i\ell}^m(t) \qquad t>0 \\ \dfrac{\mathrm{d}\mathbf{p}_i(t)}{\mathrm{d}t} = \mathbf{v}_i(t) \qquad t>0 \\ \mathbf{p}_i(0) = \mathbf{p}_i^0, \mathbf{v}_i(0) = \mathbf{v}_i^0. \end{cases}$$

Force d'attraction

On introduit la force d'attraction suivante :

$$f_i^a(t) = m_i \frac{w_i(t)\mathbf{d}_i(t) - \mathbf{v}_i(t)}{\tau_i}$$

où $\mathbf{d}_i(t)$ désigne la direction normalisée désirée. On peut, par exemple, choisir de se diriger vers un point cible $\mathbf{c}_i$:

$$\mathbf{d}_i(t) = \begin{cases} \dfrac{\mathbf{c}_i - \mathbf{p}_i(t)}{\|\mathbf{c}_i - \mathbf{p}_i(t)\|} & \text{si } \mathbf{p}_i(t) \neq \mathbf{c}_i \\ \mathbf{0} & \text{sinon.} \end{cases}$$

Force d'interaction entre individus

La force de répulsion entre deux individus a la forme suivante :

$$\mathbf{f}_{ij}^p(t) = Ae^{\frac{s_{ij}(t)}{B}}\mathbf{n}_{ij}(t) + k_1 g(s_{ij}(t))\mathbf{n}_{ij}(t) + k_2 g(s_{ij})\delta_{ij}(t)\mathbf{t}_{ij}(t)$$

avec

$$d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|,\ s_{ij} = r_i + r_j - d_{ij},\ \mathbf{n}_{ij} = \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|},\ \mathbf{t}_{ij} = (-n_{ij}^y, n_{ij}^x),\ \delta_{ij} = (\mathbf{v}_i - \mathbf{v}_j | \mathbf{t}_{ij}),\ g(x) = \max(x, 0)$$

et les paramètres suivants :

- A représente la portée de la force de répulsion entre individus et est comprise entre 0.05m et 0.2m
- B représente l'amplitude de la force de répulsion entre individus et est comprise entre 500 et 5000N
- k_1 représente l'amplitude d'une force de compression et est comprise entre 100000N et 150000N
- k_2 représente l'amplitude d'une force de friction ay contact et est comprise entre 200000N et3050000N

Force d'interaction avec les murs

La force d'interaction avec les murs a la même forme que la force d'interaction entre individus, en prenant pour point la projection du point courant sur le mur ℓ : $\mathbf{p}_{i\ell} = \Pi_{M_\ell}(\mathbf{p}_i)$. La force d'interaction s'écrit alors :

$$\mathbf{f}_{i\ell}^m(t) = Ae^{\frac{s_{i\ell}(t)}{B}}\mathbf{n}_{i\ell}(t) + k_1 g(s_{i\ell}(t))\mathbf{n}_{i\ell}(t) + k_2 g(s_{i\ell})\delta_{i\ell}(t)\mathbf{t}_{i\ell}(t)$$

avec

$$d_{i\ell} = \|\mathbf{p}_i - \mathbf{p}_{i\ell}\|,\ s_{i\ell} = r_i - d_{i\ell},\ \mathbf{n}_{i\ell} = \frac{\mathbf{p}_i - \mathbf{p}_{i\ell}}{\|\mathbf{p}_i - \mathbf{p}_{i\ell}\|},\ \mathbf{t}_{i\ell} = (-n_{i\ell}^y, n_{i\ell}^x),\ \delta_{i\ell} = (\mathbf{v}_i | \mathbf{t}_{i\ell}).$$

Discrétisation

En introduisant les instants $t_k = k\,\Delta t$, $k = 0$ à n, v_i^k une approximation de $v_i(t_k)$, le système dynamique $\mathcal{D}$ est approché par différences finies :

$$(\mathcal{D}_{\Delta t}) \qquad \begin{cases} m_i \dfrac{\mathbf{v}_i^{k+1} - \mathbf{v}_i^k}{\Delta t} = (\mathbf{f}_i^a)^k + \displaystyle\sum_{j \neq i} (\mathbf{f}_{ij}^p)^k + \sum_{\ell} (\mathbf{f}_{i\ell}^m)^k & k = 0, n-1 \\ \dfrac{\mathbf{p}_i^{k+1} - \mathbf{p}_i^k}{\Delta t} = \mathbf{v}_i^{k+1} & k = 0, n-1 \\ \mathbf{p}_i(0) = \mathbf{p}_i^0, \mathbf{v}_i(0) = \mathbf{v}_i^0. & \end{cases}$$

Les quantités $(\mathbf{f}_i^a)^k$,$(\mathbf{f}_{ij}^p)^k$ et $(\mathbf{f}_{ik}^m)^k$ sont définis de la même manière que $\mathbf{f}_i^a(t)$,$\mathbf{f}_{ij}^p(t)$ et $\mathbf{f}_{i\ell}^m(t)$ où $\mathbf{p}_i(t)$ est remplacé par p_i^k et $\mathbf{v}_i(t)$ par v_i^k.

Dans un premier temps on propose d'utiliser l'algorithme de mise à jour suivant :

- pour $k = 0,\ n-1$
  - mise à jour des forces de tous les individus
  - pour chaque individu $i = 1, N$

    mise à jour de la vitesse $\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + \dfrac{\Delta t\, \mathbf{f}_i^k}{m_i}$

    mise à jour de la position $\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \Delta t\, \mathbf{v}_i^{k+1}$

On pourra tester dans un deuxième temps l'algorithme suivant, où on déplace les individus au fur et à mesure :

- pour $k = 0,\ n-1$
  - pour chaque individu $i = 1, N$ pris de façon aléatoire

    mise à jour de la force $\mathbf{f}_i^k$ s'exerçant sur l'individu i

    mise à jour de la vitesse $\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + \dfrac{\Delta t\, \mathbf{f}_i^k}{m_i}$

    mise à jour de la position $\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \Delta t\, \mathbf{v}_i^{k+1}$

Pour le calcul des forces s'exerçant sur un individu, dans un premier temps on considérera toutes les interactions avec tous les murs et tous les autres individus. Comme les interactions sont très faibles lorsque l'individu est loin d'un mur ou loin d'un autre individu, dans un deuxième temps, on proposera une méthode permettant de limiter le calcul des forces d'interaction aux murs et individus proches.

2 Mise en œuvre

Afin de gérer des points ou des vecteurs de dimension 2, on développera une classe `Vecteur` gérant explicitement les composantes x, y pour de meilleures performances :

```
class Vecteur{
public:
  double x=0, y=0;  //coordonnees d'un point 2D
  ...
};
```

On aura, principalement, besoin des opérateurs algébriques usuels permettant de réaliser des combinaisons linéaires, du produit scalaire (opérateur |) et de la norme. `Point` sera défini comme un alias de `Vecteur`. On aura également besoin de calculer la projection d'un point sur une droite ou sur un segment.

La classe Murs

Afin de décrire le contexte géométrique on pourra utiliser une classe `Murs` gérant une collection (`vector` ou `list`) de `Mur` que l'on supposera être des segments équipés de leurs normales sortantes si il s'agit d'un mur définissant un bor d :

```
typedef pair<Point,Point> Segment;
typedef pair&lt;Segment,Vect&gt; Mur;  // segment et normale sortante
class Murs
{ public:
  list<Mur> murs;                // liste des murs
  ...
};
```

On développera toutes les fonctions que l'on jugera utiles, en particulier :

- une fonction testant si un point se situe à l'intérieur de la pièce définie par les murs équipés de normales sortantes,
- des fonctions d'affichage de `Segment`, de `Mur` et de `Murs`,
- une fonction d'export dans un fichier permettant de dessiner le contexte géométrique dans Matlab,

La classe Individu

La classe `Individu` regroupera toutes les données attachées à un individu :

— un identifiant unique (id)
— un numéro de groupe (ng) permettant de repérer des groupes d'individus
— la masse de l'individu (m)
— le rayon du disque d'occupation (r)
— son temps de réaction (tau)
— sa vitesse désirée (w)
— son point cible (c)
— sa position courante (p)
— sa vitesse courante (v)
— la force exercée courante (f)
— l'historique de ses positions (ps)

On pourra lui adjoindre les fonctions membre calculant la forces d'attraction, la force d'interaction avec un autre individu et la force d'interaction avec un mur.

La classe Foule

La classe `Foule` gérera principalement une collection d'individus (`vector<Individu>` ou `list<Individu>`). Cette classe devra proposer des fonctions pour générer de foules, de façon aléatoire ou déterministe suivant différents motifs dépendant du contexte géométrique. On pourra également prévoir de créer une foule à partir d'un fichier décrivant les individus, formaté de la façon suivante

```
id qx qy vx vy m r ng tau vd cx cy
...
```

Ce format pourra être également utilisé pour générer des fichiers de sortie exploités par Matlab pour représenter graphiquement l'état d'une foule à un instant donné. La classe `Foule` proposera une fonction réalisant la mise à jour des forces de tous les individus.

Lors de la création de la foule, on pourra choisir les mêmes masse, rayon, temps de réaction, vitesse désirée et point cible ou pour être un peu plus réaliste, choisir ces caractéristiques de façon aléatoire dans des intervalles donnés.

La classe Dynamique

La classe `Dynamique` collectera tous les éléments d'une simulation de la dynamique de la foule :

— un pointeur sur le contexte géométrique (`Murs`)
— un pointeur sur une foule
— les paramètres $A,\ B,\ k_1,\ k_2$ de la dynamique
— les paramètres de simulation $dt,\ nt$

Par exemple :

```
class Dynamique{
   const Murs* murs =nullptr;   // geometrie
   Foule* foule=nullptr;        // foule
  public:
   double k1=100000, k2=200000; // 100000<k1<150000 et  200000<k2<300000
   double A=2000, B=0.08;       // 500N < A <5000N, 0.05m < B <0.2m
   double dt;                   // pas de temps de la simulation
   int nbt;                     // nombre de pas de temps
   ...
```

Cette classe proposera les fonctions permettant de réaliser

— le calcul de la dynamique suivant les algorithmes indiqués
— l'export des positions et caractéristiques des individus à tout les instants soit dans un seul fichier soit dans plusieurs fichiers (un fichier correspondant à un instant) ; ces fichiers serviront à réaliser des représentations graphiques de la foule à différents instants, et à réaliser une animation (VideoWriter avec Matlab).

3 Organisation

Pour la réalisation de ce projet par 3 élèves on propose le partage du travail suivant :

— Un premier élève se consacre au développement de la classe `Murs` et des fonctions de lecture et écriture sur fichier pour exploitation par Matlab ; cet élève prendra également en charge le développement des scripts Matlab permettant d'afficher la géométrie et la foule, puis dans un second temps la réalisation d'animations.

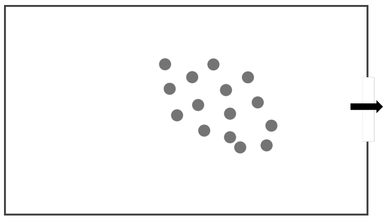
— le second élève se consacre au développement des classes `Individu` et `Foule` (génération des foules, calcul des forces)

— le troisième élève se consacre au développement de la classe `Dynamique` (export de la dynamique calculée)

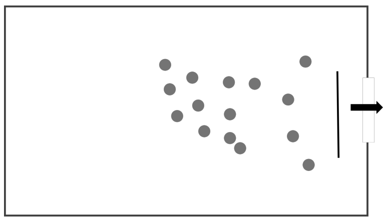
4 Validation et scénarios

Tout d'abord, on validera la dynamique sur des cas simples : 1 seul individu proche d'un mur ou se dirigeant vers un mur ou deux individus en interaction. On pourra ensuite tester différents scénarios, en considérant de plus en plus d'individus :

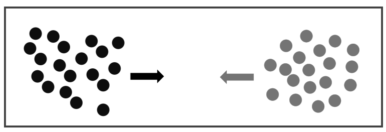
la sortie d'une salle



la sortie d'une salle obstruée



le croisement de foule



On pourra également trouver d'autres scénarios intéressants sur internet. En ce qui concerne les paramètres on trouve les indications suivantes :

contexte	A	B
normal	2000N	0.08m
flux dense	3000-5000N	0.05-0.1m
panique	5000-10000N	0.1-0.2m
personne calme	500-1500N	0.05-0.15m

5 Extensions

Prise en compte d'une dépendance angulaire

Dans sa forme originelle, le modèle de Helbing considère des interactions isotropes (identiques dans toutes les directions). On peut prendre un effet directionnel en indiquant que les agents réagissent plus fort à ce qui se trouve devant. A cet effet, on multiplie la force d'interaction par un poids angulaire :

$$\omega(\varphi_{ij}) = \lambda + (1 - \lambda)\frac{1 + \cos\varphi_{ij}}{2}$$

où φ_{ij} représente l'angle de vision :

$$\varphi_{ij} = \widehat{\mathbf{v}_i\,\mathbf{d}_{ij}} \qquad (\mathbf{d}_{ij} = \mathbf{p}_j - \mathbf{p}_i).$$

Noter que $\cos\varphi_{ij} = (\mathbf{v}_i|\mathbf{d}_{ij})$. Le paramètre $0 \leqslant \lambda \leqslant 1$ représente les effets de réaction arrière ($\lambda = 0$ aucun effet de réaction arrière, $\lambda = 1$ redonne le comportement isotrope). En pratique on choisit $\lambda = 0.5$.

Prise en compte d'une anticipation

La prise en compte d'une anticipation dans le modèle de Helbing consiste à intégrer les positions futures, vitesses futures et temps probable avant collision dans les forces sociales. Ce qui rend le modèle beaucoup plus proche du comportement humain réel. On conserve l'expression des forces d'interactions entre individu et avec les murs, mais en remplaçant les distances par des distances anticipées (temps d'anticipation $\widetilde{\Delta t}$) :

$$\widetilde{\mathbf{p}}_i = \mathbf{p}_i + \widetilde{\Delta t}\,\mathbf{v}_i, \quad \widetilde{\mathbf{d}}_{ij} = \widetilde{\mathbf{p}}_i - \widetilde{\mathbf{p}}_j,\ \widetilde{d}_{ij} = \|\widetilde{\mathbf{d}}_{ij}\|,\ \widetilde{\mathbf{n}}_{ij} = \frac{\widetilde{\mathbf{d}}_{ij}}{\widetilde{d}_{ij}},\ \widetilde{s}_{ij} = r_i + r_j - \widetilde{d}_{ij}.$$

On peut également corriger la force d'interaction entre les individus i, j en introduisant un facteur d'amortissement :

$$\widetilde{\mathbf{f}}^p_{ij}(t) = Ae^{\frac{\widetilde{s}_{ij}}{B}}e^{-\frac{\tau_{ij}}{\tau_0}}\widetilde{\mathbf{n}}_{ij}(t) + k_1 g(\widetilde{s}_{ij}(t))\widetilde{\mathbf{n}}_{ij}(t) + k_2 g(\widetilde{s}_{ij})\delta_{ij}(t)\widetilde{\mathbf{t}}_{ij}(t)$$

où

$$\tau_{ij} = \frac{-\left(\Delta\mathbf{p}_{ij}|\Delta\mathbf{v}_{ij}\right)}{\|\Delta\mathbf{v}_{ij}\|^2}, \qquad \Delta\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j, \qquad \Delta\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j.$$

Objectifs intermédiaires

Afin d'éviter des mouvements erratiques du fait d'une trop grande distance ou un trop grand nombre d'obstacles, on peut ajouter des objectifs intermédiaires dans la simulation. En pratique, il peut s'agir de points de "passage" $M_j, j = 1, J$ connus au démarrage de la simulation et définissant une trajectoire idéale. Ainsi, le point cible $\mathbf{c}_i$ de l'individu i est mis à jour au fur et à mesure de la simulation. On change de point cible dès lors que l'individu a atteint un voisinage de sa cible courante.

Références

Helbing, D., Molnár, P. (1995). Social force model for pedestrian dynamics. Physical Review E, 51(5), 4282–4286. (https ://arxiv.org/pdf/cond-mat/9805244)
Helbing, D., Farkas, I., Vicsek, T. (2000). "Simulating dynamical features of escape panic." Nature, 407, 487–490. (https ://arxiv.org/pdf/cond-mat/0009448)
Helbing, D., Johansson, A. (2009). "Pedestrian, crowd and evacuation dynamics." In Encyclopedia of Complexity and Systems Science. (https ://arxiv.org/pdf/1309.1609)