Mobile Robots and Autonomous Vehicles

Week 5: Behavior Modeling and Learning

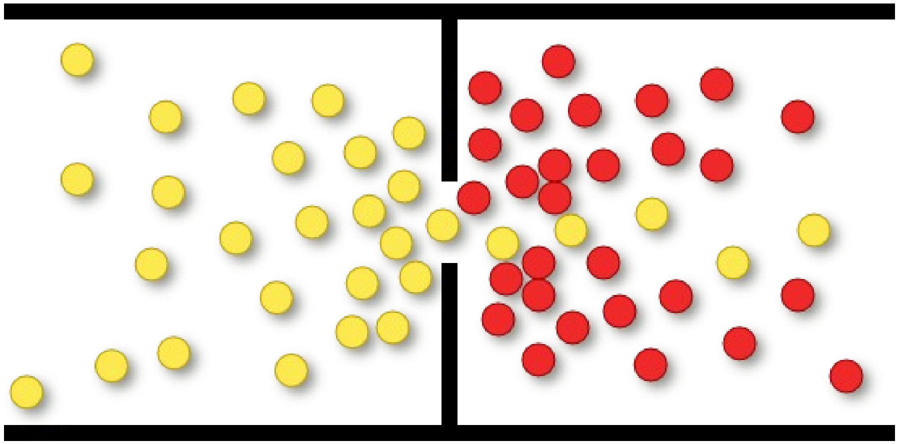
- Other approaches: Social Forces

Dizan Vasquez

informatiques mathématiques
Inria

Social Forces Model

- Reactive approach for crowd simulation
- Models interactions within people
- Does not depend on the environment
- Intended destination has to be estimated somehow



Social Forces: general algorithm

- At every time step:
  - Compute the total force for each agent i:

$$m_i \frac{dv_i}{dt} = m_i \frac{\hat{v}_i(t)\hat{e}_i(t) - v_i(t)}{\tau} + \sum_{j \neq i} f_{ij} + \sum_{w} f_{iw}$$

  - Compute acceleration and update the agent's position

Social Forces: general algorithm

Desired motion

- At every time step:
  - Compute the total force for each agent i:

$$m_i \frac{dv_i}{dt} = \boxed{m_i \frac{\hat{v}_i(t)\hat{e}_i(t) - v_i(t)}{\tau}} + \sum_{j \neq i} f_{ij} + \sum_{w} f_{iw}$$

  - Compute acceleration and update the agent's position

Social Forces: general algorithm

- At every time step:
  - Compute the total force for each agent i:

$$m_i \frac{dv_i}{dt} = \overbrace{m_i \frac{\hat{v}_i(t)\hat{e}_i(t) - v_i(t)}{\tau}}^{\text{Desired motion}} + \underbrace{\sum_{j \neq i} f_{ij}}_{\text{Other pedestrians}} + \sum_{w} f_{iw}$$

  - Compute acceleration and update the agent's position

Social Forces: general algorithm

- At every time step:
  - Compute the total force for each agent i:

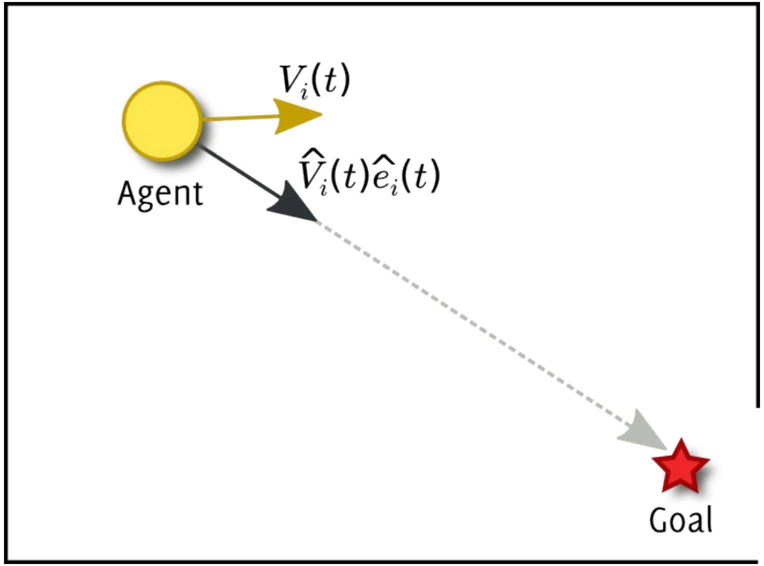
$$m_i \frac{dv_i}{dt} = \underbrace{m_i \frac{\hat{v}_i(t)\hat{e}_i(t) - v_i(t)}{\tau}}_{\text{Desired motion}} + \underbrace{\sum_{j \neq i} f_{ij}}_{\text{Other pedestrians}} + \underbrace{\sum_{w} f_{iw}}_{\text{Static obstacles}}$$

  - Compute acceleration and update the agent's position

Social Forces: desired motion

$$m_i \frac{\hat{v}_i(t)\hat{e}_i(t) - v_i(t)}{\tau}$$

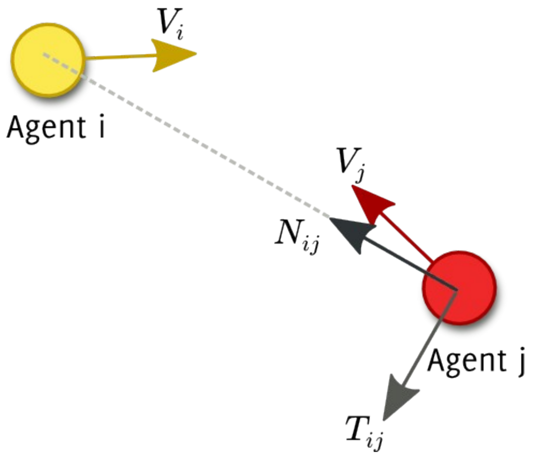
- m_i : Agent's mass
- $\hat{v}_i(t)$: Desired direction
- $\hat{e}_i(t)$: Desired speed
- $v_i(t)$: Actual velocity
- τ : Time interval



Social Forces: pedestrian interaction

$$F_{ij} = \{ae^{(r_{ij}-d_{ij})/b} + kg(r_{ij}-d_{ij})\}N_{ij} + \kappa g(r_{ij}-d_{ij})\Delta v_{ji}T_{ij}$$

- a, b, k, κ : tuning variables
- d_{ij} : agent distance
- r_{ij} : sum of agents' radius
- N_{ij} : relative direction from *j* to *i*
- $g(x)$: x if colliding, *0* otherwise
- T_{ij} : tangential direction
- v_{ji} : tangential velocity difference

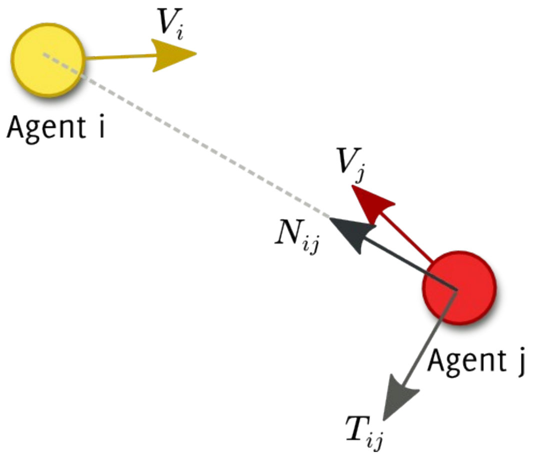


Social Forces: pedestrian interaction

$$F_{ij} = \{ae^{(r_{ij}-d_{ij})/b} + kg(r_{ij}-d_{ij})\}N_{ij} + \kappa g(r_{ij}-d_{ij})\Delta v_{ji}T_{ij}$$

Collision avoidance

- a, b, k, κ : tuning variables
- d_{ij} : agent distance
- r_{ij} : sum of agents' radius
- N_{ij} : relative direction from *j* to *i*
- $g(x)$: *x* if colliding, *0* otherwise
- T_{ij} : tangential direction
- v_{ji} : tangential velocity difference



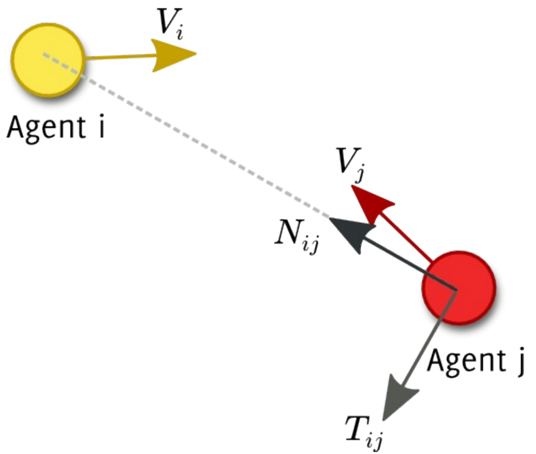
Social Forces: pedestrian interaction

$$F_{ij} = \{ae^{(r_{ij}-d_{ij})/b} + kg(r_{ij} - d_{ij})\}N_{ij} + \kappa g(r_{ij} - d_{ij})\Delta v_{ji}T_{ij}$$

Collision avoidance

Non-penetration

- a, b, k, κ : tuning variables
- d_{ij} : agent distance
- r_{ij} : sum of agents' radius
- N_{ij} : relative direction from *j* to *i*
- $g(x)$: *x* if colliding, *0* otherwise
- T_{ij} : tangential direction
- v_{ji} : tangential velocity difference



Social Forces: pedestrian interaction

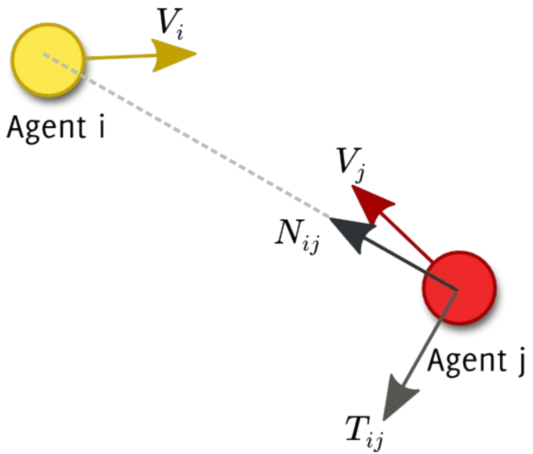
$$F_{ij} = \{ae^{(r_{ij}-d_{ij})/b} + kg(r_{ij}-d_{ij})\}N_{ij} + \kappa g(r_{ij}-d_{ij})\Delta v_{ji}T_{ij}$$

Collision avoidance

Non-penetration

Sliding force

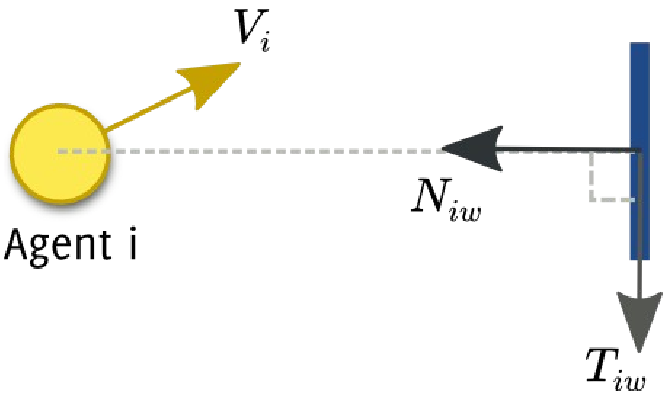
- a, b, k, κ : tuning variables
- d_{ij} : agent distance
- r_{ij} : sum of agents' radius
- N_{ij} : relative direction from *j* to *i*
- $g(x)$: x if colliding, *0* otherwise
- T_{ij} : tangential direction
- v_{ji} : tangential velocity difference



Social Forces: static interaction

$$F_{iw} = \{ae^{(r_i - d_{iw})/b} + kg(r_i - d_{iw})\}N_{iw} + \kappa g(r_i - d_{iw})(v_i \cdot T_{iw})T_{iw}$$

- a, b, k, κ : tuning variables
- d_{iw} : wall distance
- r_i : agent's radius
- N_{iw} : relative direction from wall
- $g(x)$: x if colliding, *0* otherwise
- T_{iw} : tangential direction
- v_i : agent's velocity

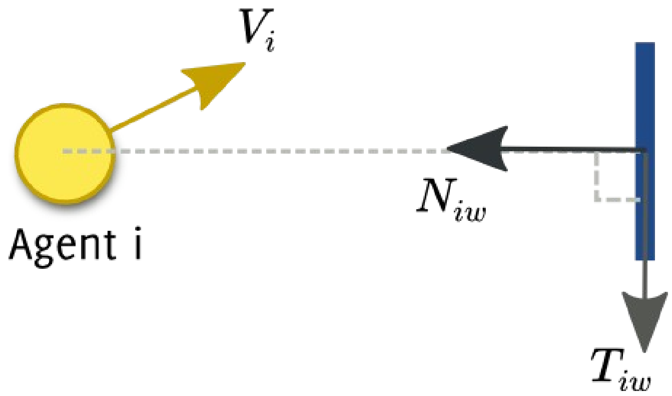


Social Forces: static interaction

$$F_{iw} = \{ae^{(r_i - d_{iw})/b} + kg(r_i - d_{iw})\}N_{iw} + \kappa g(r_i - d_{iw})(v_i \cdot T_{iw})T_{iw}$$

Collision avoidance

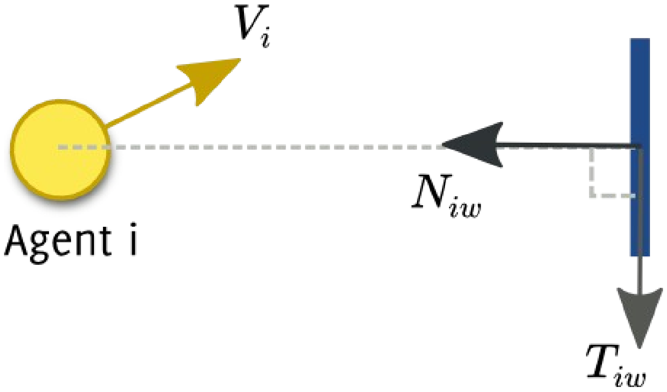
- a, b, k, κ : tuning variables
- d_{iw} : wall distance
- r_i : agent's radius
- N_{iw} : relative direction from wall
- $g(x)$: x if colliding, *0* otherwise
- T_{iw} : tangential direction
- v_i : agent's velocity



Social Forces: static interaction

$$F_{iw} = \{\underbrace{ae^{(r_i - d_{iw})/b}}_{\text{Collision avoidance}} + \underbrace{kg(r_i - d_{iw})}_{\text{Non-penetration}}\}N_{iw} + \kappa g(r_i - d_{iw})(v_i \cdot T_{iw})T_{iw}$$

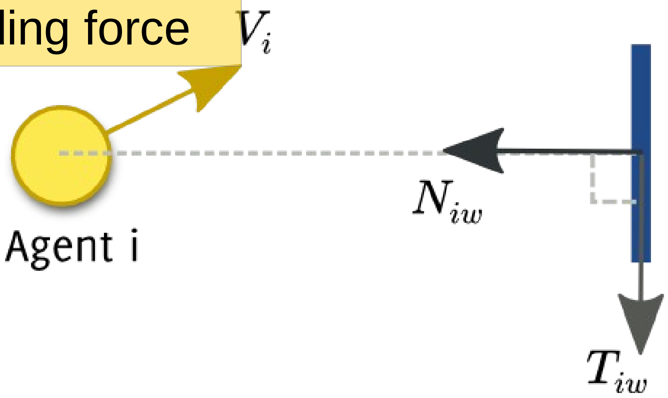
- a, b, k, κ : tuning variables
- d_{iw} : wall distance
- r_i : agent's radius
- N_{iw} : relative direction from wall
- $g(x)$: x if colliding, *0* otherwise
- T_{iw} : tangential direction
- v_i : agent's velocity



Social Forces: static interaction

$$F_{iw} = \{\underbrace{ae^{(r_i - d_{iw})/b}}_{\text{Collision avoidance}} + \underbrace{kg(r_i - d_{iw})}_{\text{Non-penetration}}\}N_{iw} + \underbrace{\kappa g(r_i - d_{iw})(v_i \cdot T_{iw})T_{iw}}_{\text{Sliding force}}$$

- a, b, k, κ : tuning variables
- d_{iw} : wall distance
- r_i : agent's radius
- N_{iw} : relative direction from wall
- $g(x)$: x if colliding, *0* otherwise
- T_{iw} : tangential direction
- v_i : agent's velocity



Social Forces vs. HMMs

HMMs

- ✔ Infers intentions
- ✔ Long-term prediction
- ✗ Requires robust data association
- ✗ Requires global perception
- ✗ Does not model person to person interaction
- ✗ Only for original environment

Social Forces

- ✔ Models person to person and person to environment interactions
- ✔ Works on any environment
- ✔ Does not require robust data association.
- ✗ Does not infer intention
- ✗ Only short-term predictions