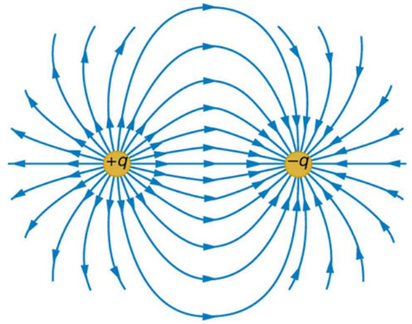
Animating crowds of characters

Interaction between particles

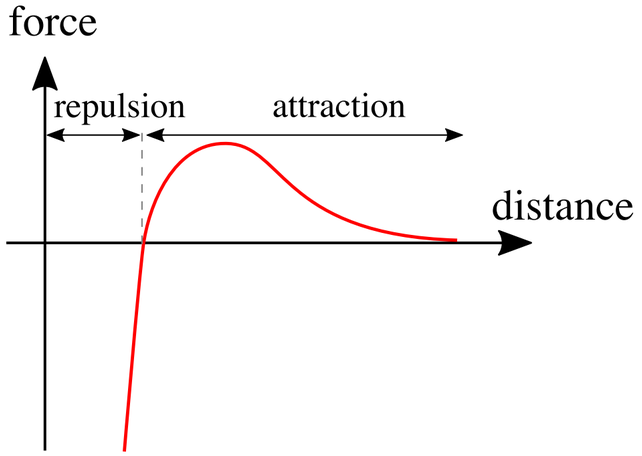
Interaction as **force field** (interact at distance)

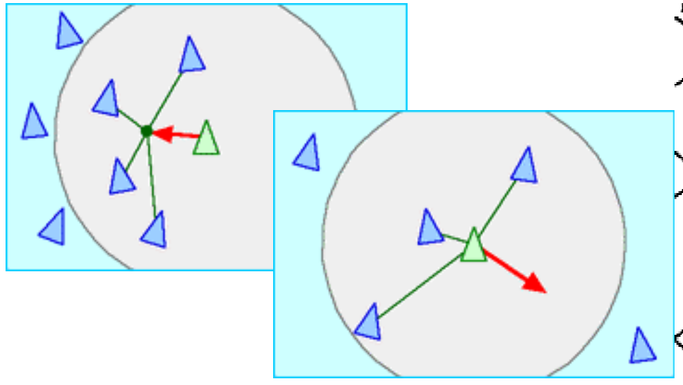


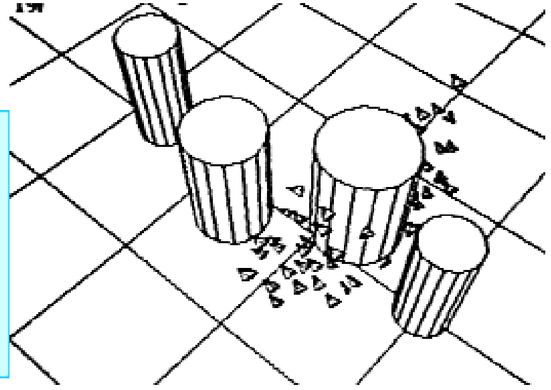
Example of usage

- Models crowd of life-like characters at large scale
  Inspired from physics particles forces (ex. Lennard-Jones potential)
  Attraction at *long-range*
  Repulsion at *short-range*

- First model: **Boids** Craig Reynolds 1987
- Extended later to human crowd modeling







Boids Model

Introduced by

[Craig Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model, SIGGRAPH 1987]

[Craig Reynolds. Steering Behaviors For Autonomous Characters. Proceedings of Game Developers, 1999]

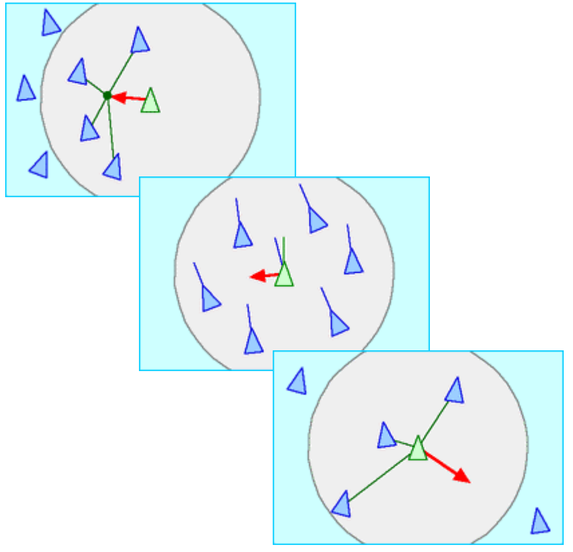
A boid is defined by its

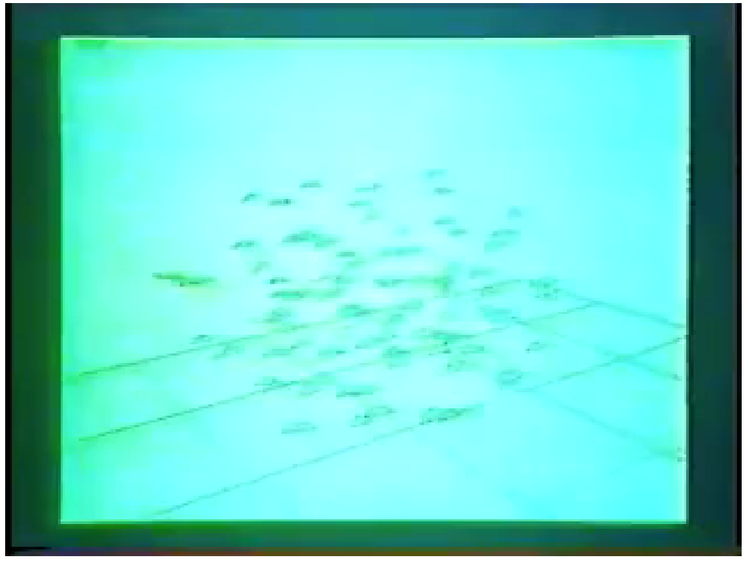
- Position
- Speed
- Forces acting on it

Three basic local *steering behaviors* to model flocks

- **Cohesion** between local particles
- **Alignment** between local particles
- **Separation** between too close particles

=> Leads to emerging global behaviors.





Original video in 1986 (C. Reynolds)

Boids Model - Basic model.

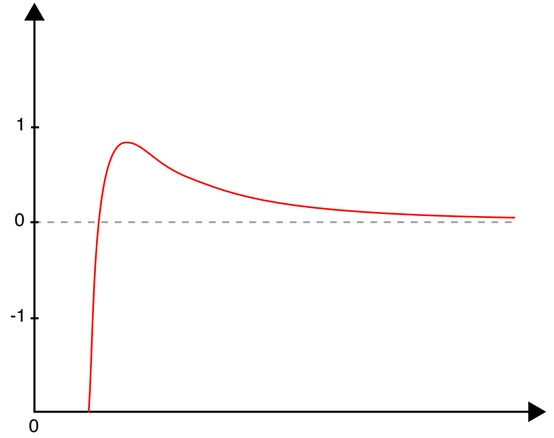
- Set random initial position/speed to N particles.

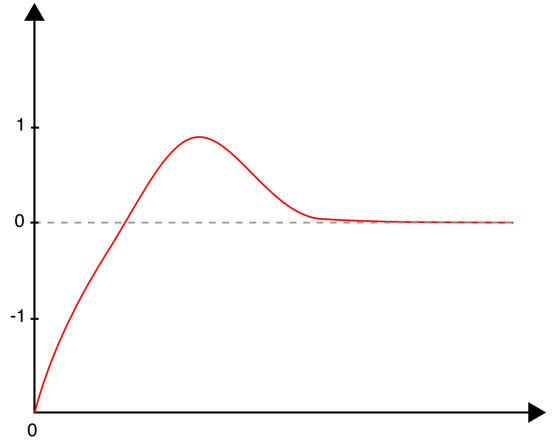
- Set attraction/repulsion force depending on pairwise distances

$$F(p_i) = \sum_j f(\|p_j - p_i\|) \frac{p_j - p_i}{\|p_j - p_i\|}$$

Example

- Inverse of distance $f(x) = \frac{\alpha_1}{x^2} - \frac{\alpha_2}{x^4}$
- Exponential/Gaussian $f(x) = \alpha_1 \exp\left(-k\left(\frac{x - x_0}{x_0}\right)^2\right) - \alpha_2 \exp\left(\frac{x}{x_0}\right)$





- Integrate position and speed through time

$$v^{t+\Delta t}(p_i) = v^t(p_i) + \Delta t \; F(p_i^t)$$
$$p^{t+\Delta t}(p_i) = p^t(p_i) + \Delta t \; v^{t+\Delta t}(p_i)$$

Boids Model - Complexity.

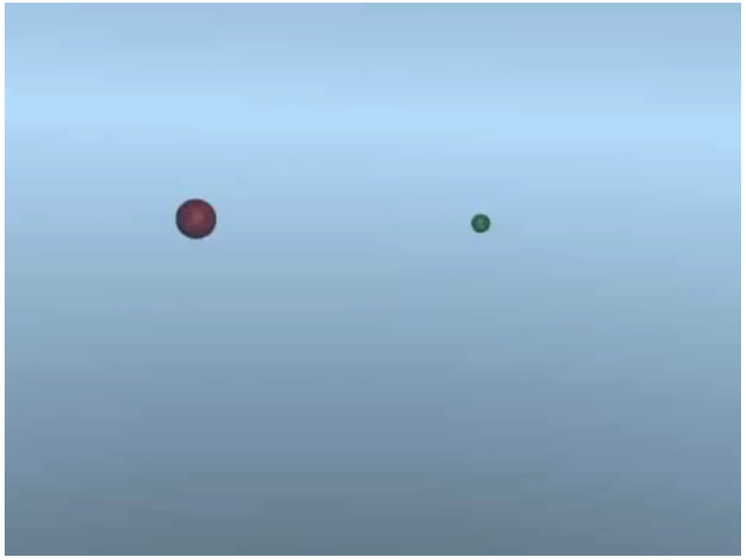
Trivial implementation:

```
struct particle { vec3 p, v, f; };

std::vector<particle> boids;

// Initialize N boids ...
// ...

// compute pairwise force
for(int i=0; i<N; ++i)
{
    for(int j=0; j<N; ++j)
    {
        if( i!=j )
        {
            const vec3& pi = boids[i].p;
            const vec3& pj = boids[j].p;

            boids[i].f += force( norm(pi,pj)) * (pi-pj)/norm(pi-pj);
        }
    }
}

// integration
for(int i=0; i<N; ++i)
{
    boids[i].v = boids[i].v + dt * boids[i].f;
    boids[i].p = boids[i].p + dt * boids[i].v;
}
```

- What is the complexity (wrt. N) of this algorithm ?
- Can you think of a way to be more efficient for large N ?

Boids Model - Usage and limitations

- Well adapted to flocks (birds, fishes - looking behavior)
- Display particles using 3D animated model



Additional behaviors

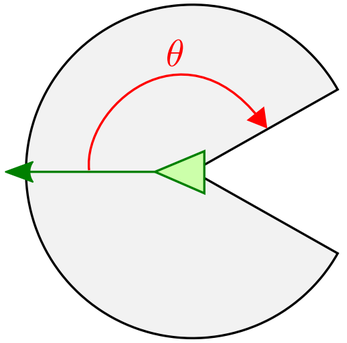
- Objective position/speed value
- Constraints: Obstacle avoidance, limited velocity
- Pursue and evade target/other particles - follow the leader, predators, etc.

- *Is collision between particles possible ?*
- *Human displacement are mostly guided by vision, what key element is missing in the basic boids force-based model ?*

Boids Model - Extensions

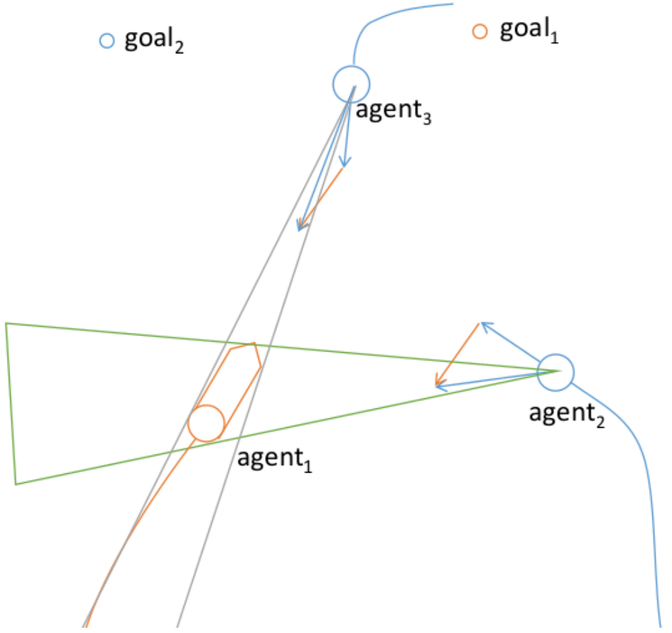
Improving force computation

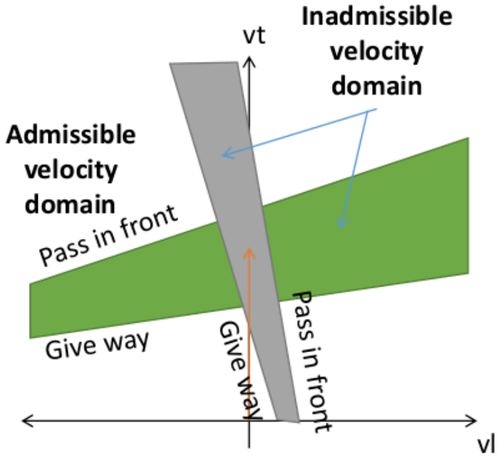
- Vision-based force computation (angle, speed) : limited view angle



Improving collision avoidance

- Velocity-based





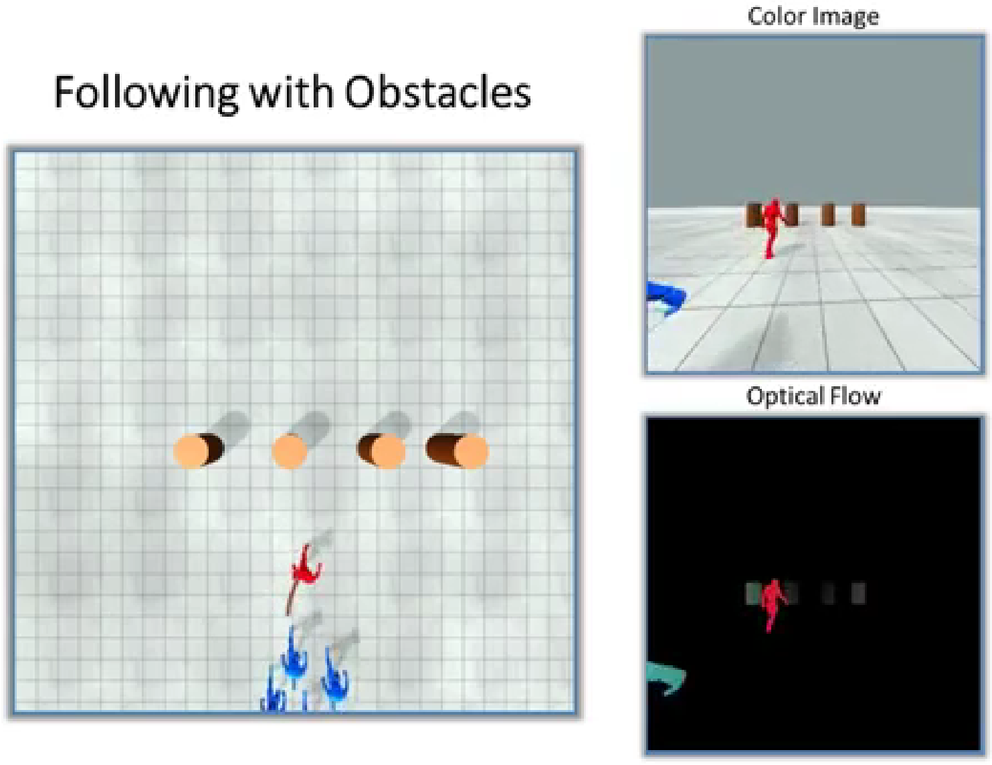
[Experiment-based Modeling, Simulation and Validation of Interactions between Virtual Walkers. J. Pettré et al. SCA 2009]

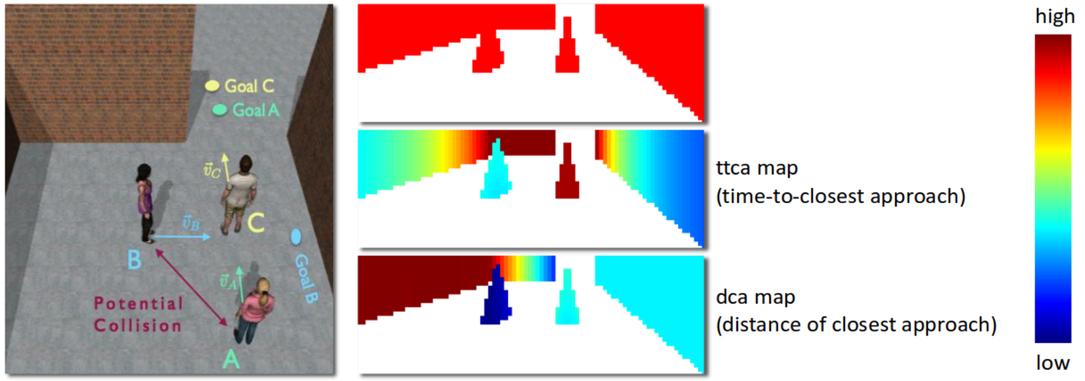
Vision based displacement

Render vision for each individual agent

- Compute optical flow u.
- Obstacles are approaching when $div(u) > 0$

Best results, but high computational cost.





[A Synthetic-Vision Based Steering Approach for Crowd Simulation. J. Ondrej et al. SIGGRAPH 2010]

[Character navigation in dynamic environments based on optical flow. A. Lopez et al. EG 2019]

Example of crowd simulation software

Golaem

In Rennes