Particle system

Particle system

Particle : element at a given position + extra parameters (mass, life time, etc)

Called **particles systems** in opposition to

- Rigid bodies - Solid objects with static shape
- Deformable bodies - Continuum material that can deforms

Pro

(+) Lightweight representation (both CPU+memory)

(+) Generic flexible model (spatial deformation, no connectivity, etc)

$\Rightarrow$ highly used in CG

Cons

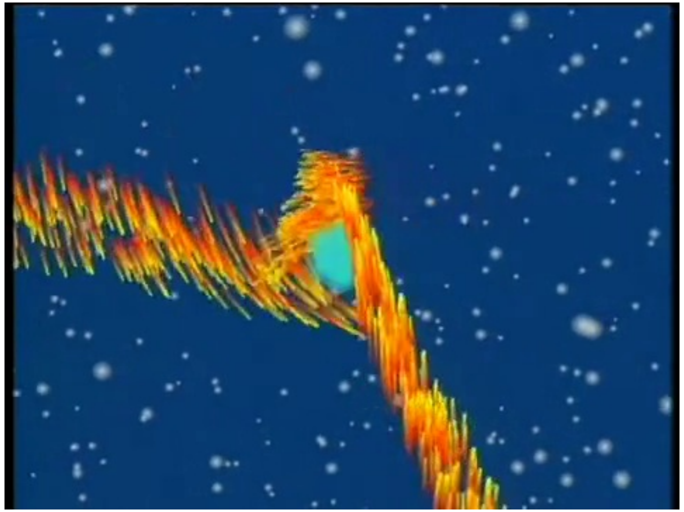
(-) Simple model from physics point of view

Particles systems History

One of the first animated model in CG

Star Trek II

[Particle systems - A Technique for Modeling a Class of Fuzzy Objects, William T. Reeves, Lucasfilm, 1983] [pdf]



Karl Sims, Particle dreams, 1988 [link]

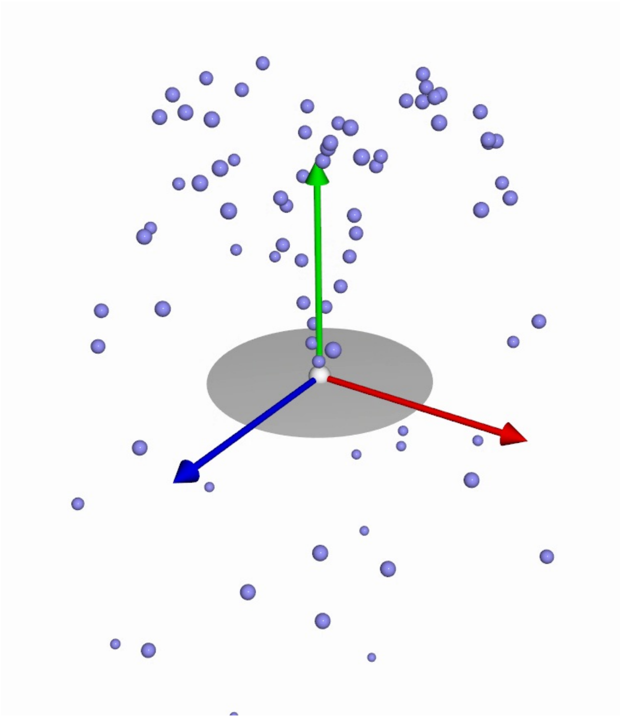
Example of Particle system

Free fall of spheres under gravity

- Geometrical representation of each particle: sphere
- Equation of motion $p(t) = \frac{1}{2}gt^2 + v_0 t + p_0$

- Initial position and speed may be placed at random position
- Each particle may have a different life time

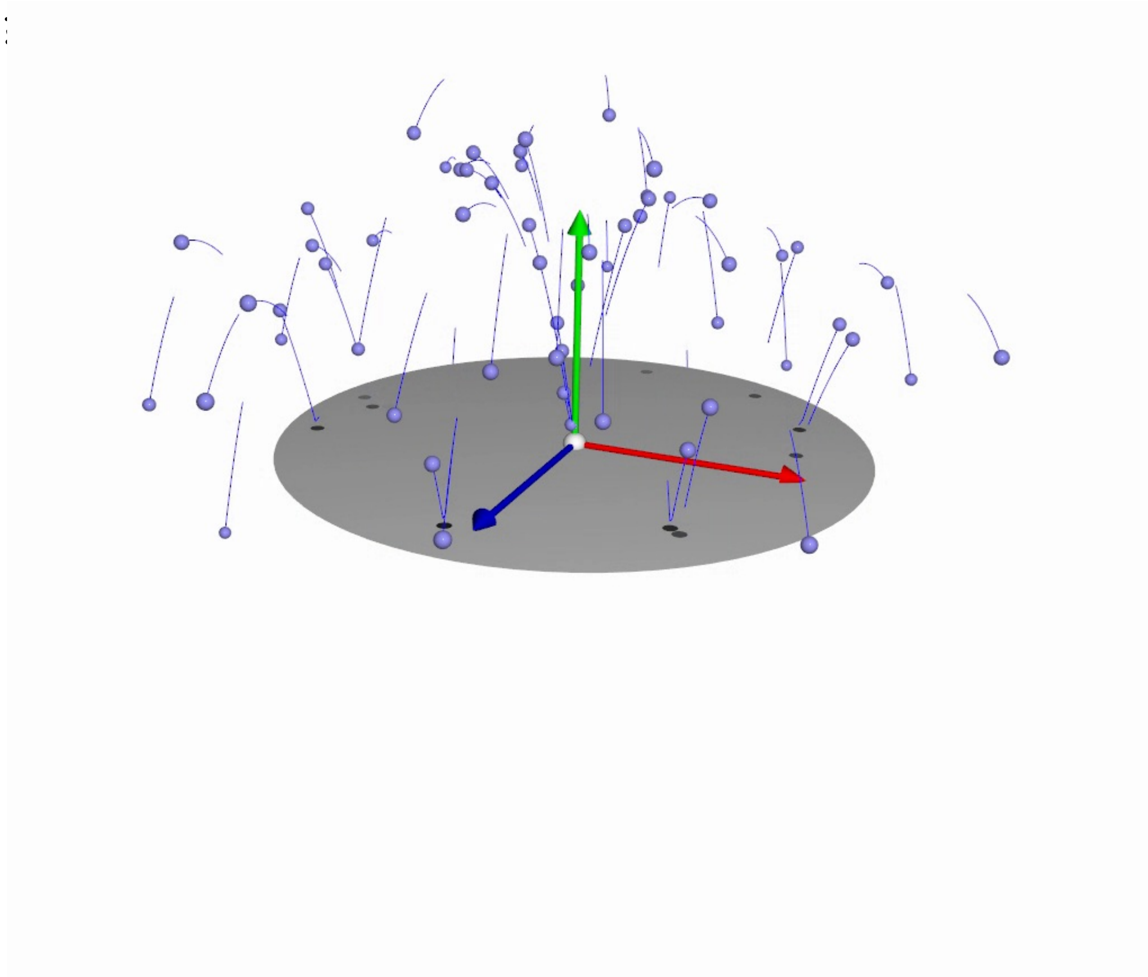
- *What are the parameters used for p_0 and v_0 in this example ?*



Bouncing spheres

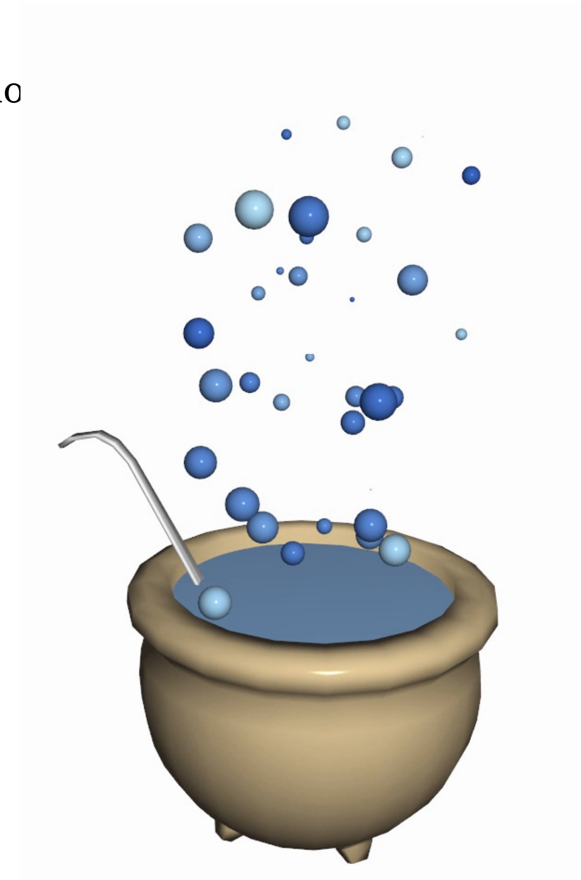
- What is the equation of motion with bouncing

    - *Consider a particle emited at time* $t = 0$
    - *At what time* t_i*, the particle touch the floor ?*
    - *What is the new speed after impact ?*



General motions

Motion equation is not restricted to physics-based equatio



- *What is the corresponding motion ?*

Billboards, Impostors, Sprites

Particles can be displayed as small images/thumbnails

In practice

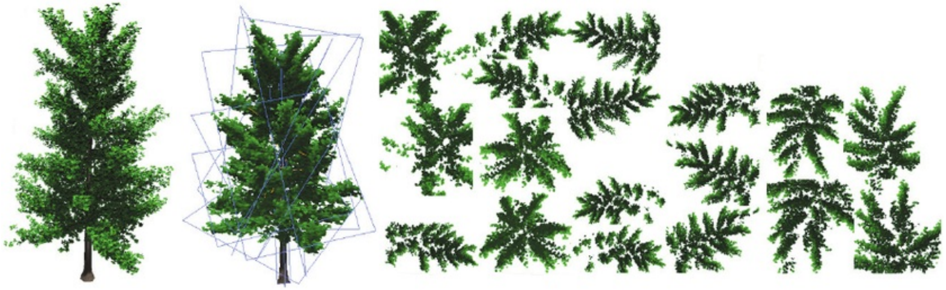
- Each particle is displayed as a quadrangle
- A texture is mapped on the quad

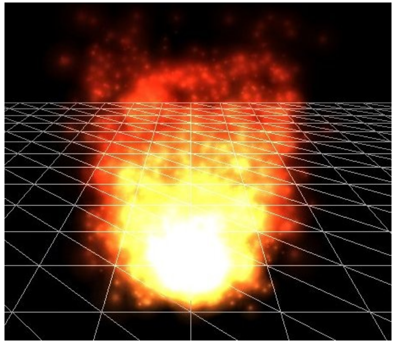
- The texture can contains transparency : quad geometry is invisible
- The quad can be facing the camera at all time (billboard)
- The texture can be animated (sprite)
- Texture can be adapted to the point of view (impostors)

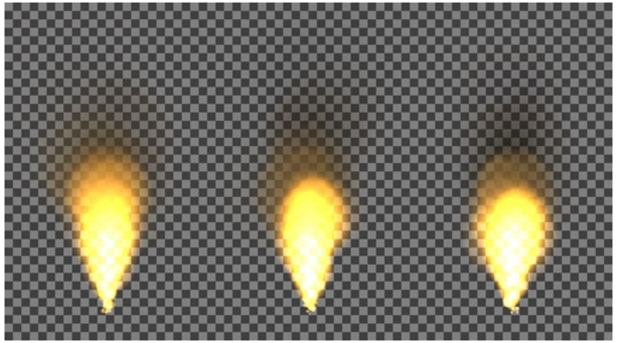
Usage of billboards

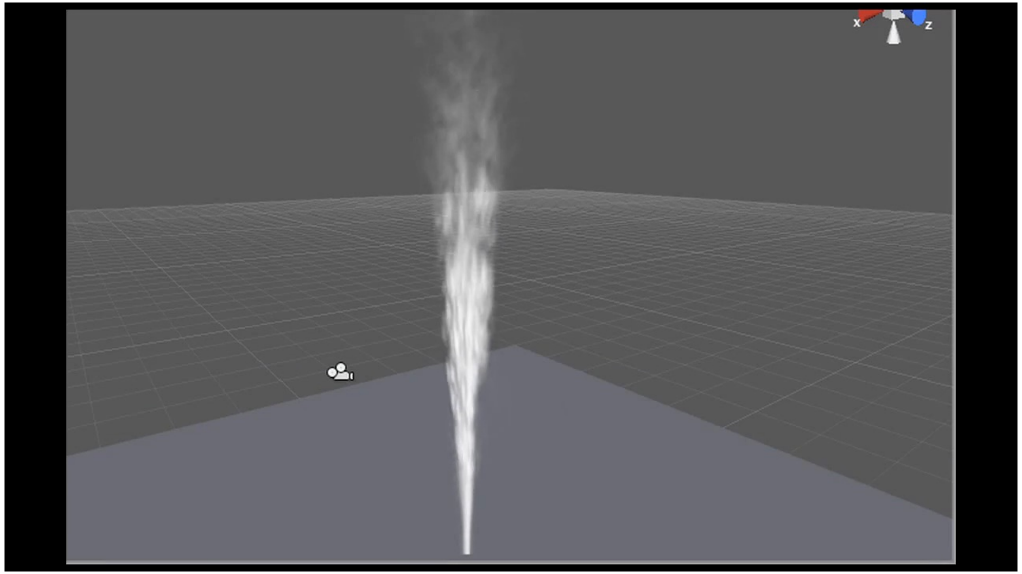
Large use of billboards for complex models

Vegetation, fire, smoke, etc.









Use case in production

How to model the *horse heads made of water* ?

The Lord of the Rings

Use case in production

How to model the *horse heads made of water* ?

The Lord of the Rings

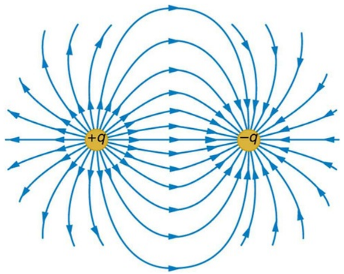
Use case in production

Full Making-Of

The Lord of the Rings

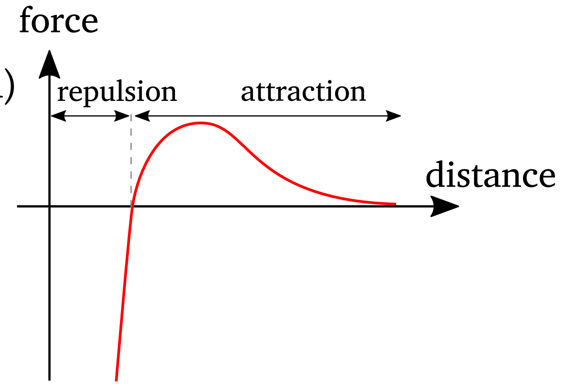
Interaction between particles

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

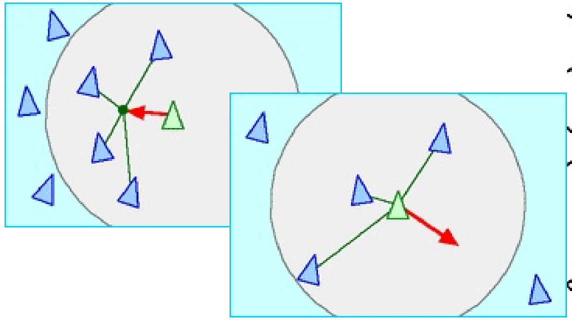


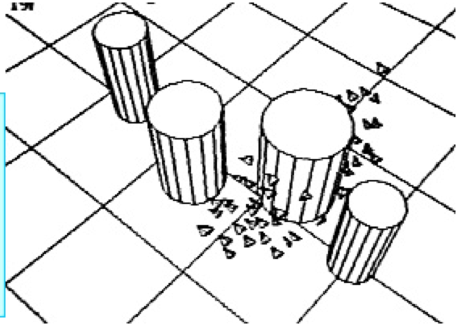
Example of usage

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

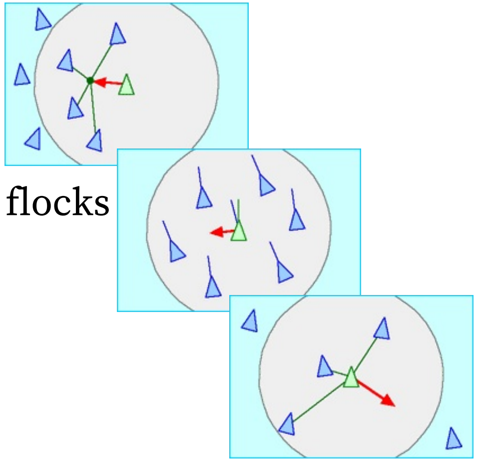
A boid is defined by its

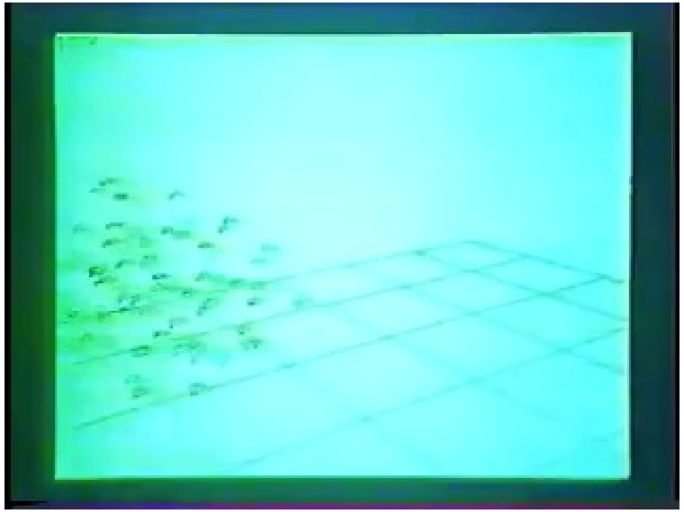
- Position
- Speed
- Forces acting on it

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.

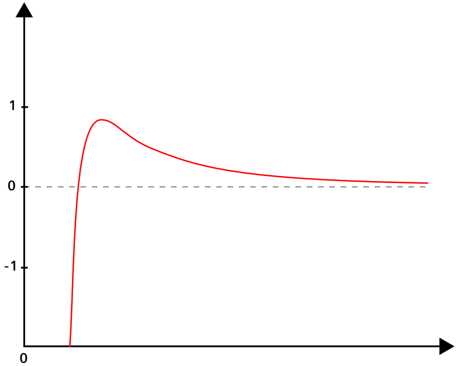
- Set random initial position/speed to N particles.

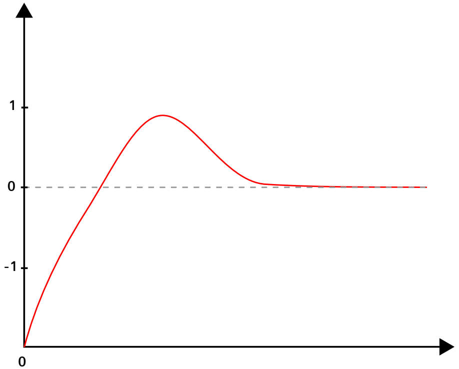
- Set attraction/repulsion force depending on pairwise distances

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

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.

Simple implementation:

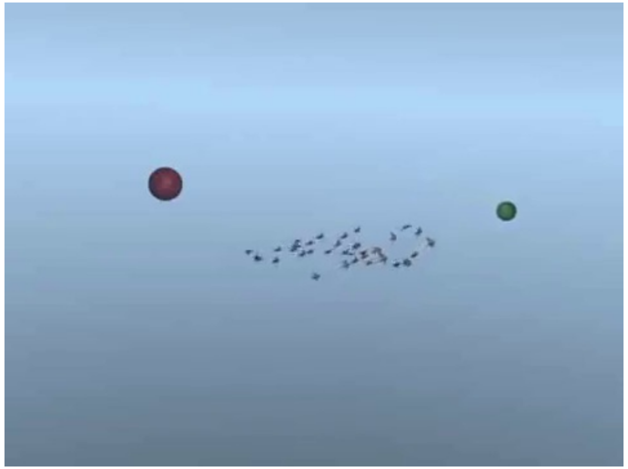
```
struct particle { vec3 p, v, f; };
std::vector<particle> boids;
// Initialize N boids ...
// ...

// 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

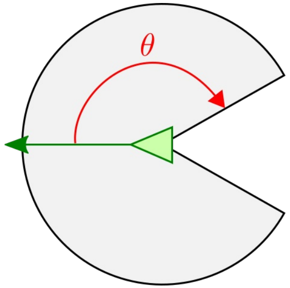
- Objective position/speed value
- Constraints: Obstacle avoidance, limited velocity
- Pursue and evade target/other particles - follow the leader, predators, etc.

- *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

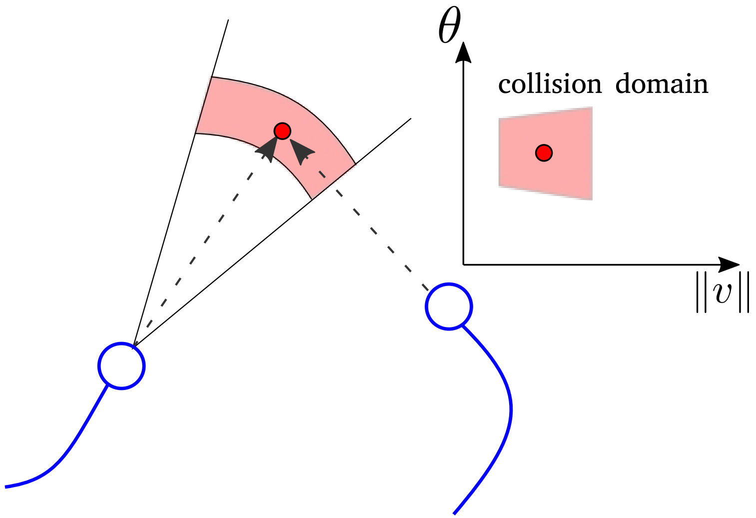
Improving force computation

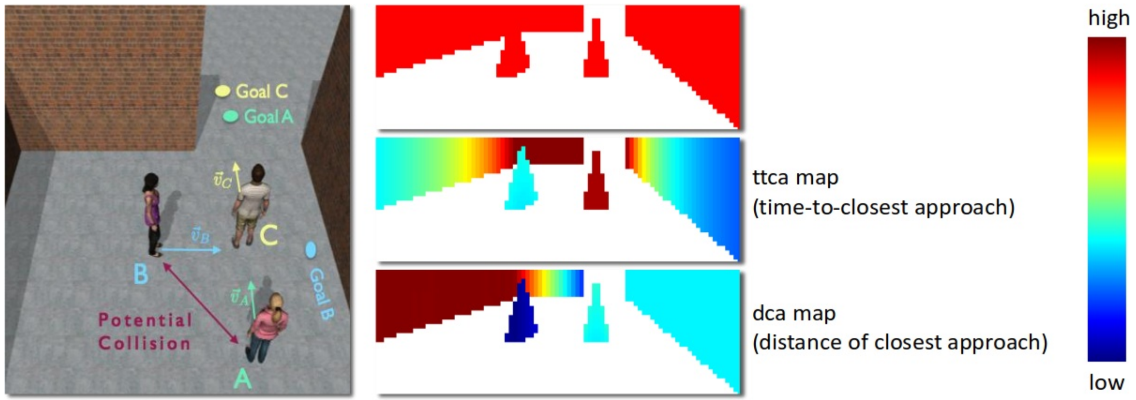
- Vision-based force computation (angle, speed) : limited view angle



Improving collision avoidance

- Velocity-based
- Vision-based





Julien Pettré, Inria Rennes

[J. Ondrej et al. A synthetic-vision based steering approach for crowd simulation, SIGGRAPH 2010]