LOCAL NAVIGATION 2

FORCE-BASED BOOKKEEPING

- Social force models
  - The forces are first-class abstractions
  - Agents are considered to be mass particles
- Other models use forces as bookkeeping
  - It is merely a way to combine multiple influences on an agent

OPENSTEER

- Based on Boids (Reynold's 1987)
  - Flocking model based on three rules
    - Separation
    - Alignment
    - Cohesion
  - http://www.youtube.com/watch?v=GUkjC-69vaw
  - http://www.red3d.com/cwr/boids/

OPENSTEER

- Based on Boids (Reynold's 1987)
  - The rules are typically implemented as forces
    - Arbitrary weights define behavior
  - Linear extrapolation detects possible collisions
    - Normal forces applied to change heading
    - Poor at collision avoidance
  - http://www.youtube.com/watch?v=dKW-psERFGA
  - http://www.youtube.com/watch?v=3CRjPwb5qoI

HiDAC - Pelechano et al. 2007

- Incorporates high-order behaviors into the model
- Applies various forces
  - Attractor force
  - Wall force, Obstacle force
  - Agent force
  - Inertial force
  - Collision force
  - Fallen-agent avoidance force
- http://www.youtube.com/watch?v=KsbChtHmwfA

HIDAC

- Application of forces is based on rules
- Examples
  - When in collision, only collision force is considered
  - When “stopping” or “waiting” repulsive forces are ignored

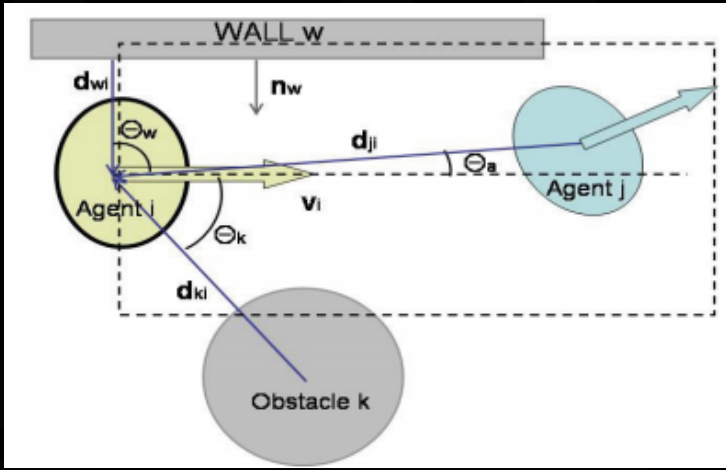
HIDAC

- Force formulation
  - "Nearby" defined by a "rectangle of influence"
  - Obstacle force

$$\mathbf{F}_{ki}^{Ob} = \frac{(\mathbf{d}_{ki} \times \mathbf{v}_i) \times \mathbf{d}_{ki}}{|(\mathbf{d}_{ki} \times \mathbf{v}_i) \times \mathbf{d}_{ki}|}$$

  - Wall force

$$\mathbf{F}_{wi}^{Wa} = \frac{(\mathbf{n}_{w} \times \mathbf{v}_i) \times \mathbf{n}_{w}}{|(\mathbf{n}_{w} \times \mathbf{v}_i) \times \mathbf{n}_{w}|}$$



From paper

HIDAC

- Apply extra rules
  - In low-speed, high-dense scenarios jittering occurs
    - The authors apply a “stopping rule”
    - Prevents responses when the forces are too strong against desired direction of travel
    - Stopping lasts for a random period of time
  - Waiting for queues (also disables responses)

AUTONOMOUS PEDESTRIANS

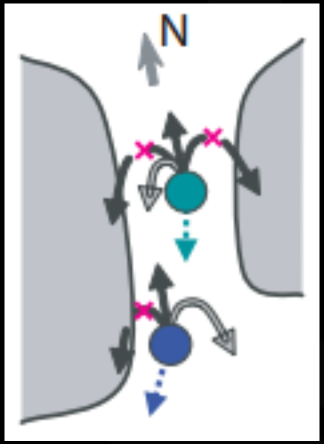
- Shao & Terzopolous, 2005
- Agent behavior based on six rules – evaluated sequentially
  - Static obstacle avoidance
  - Static obstacle avoidance with turn
  - Maintain separation
  - Avoid oncoming pedestrians
  - Avoid “dangerously” close pedestrians
  - Validate against obstacles
- http://www.youtube.com/watch?v=cqG7ADSvQ5o

AUTONOMOUS PEDESTRIANS

- Static obstacle avoidance
  - Turns preferred velocity based on nearby obstacles
  - If a great deal of turning is required, the magnitude of the preferred velocity is reduced

AUTONOMOUS PEDESTRIANS

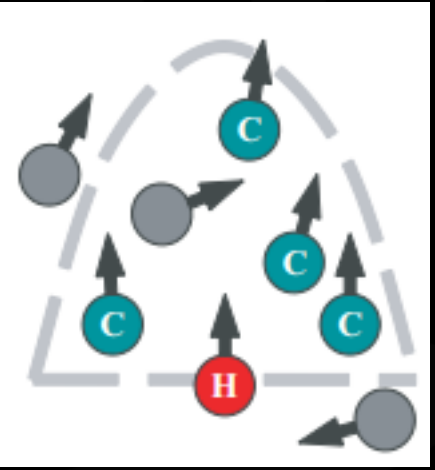
- Static obstacle avoidance with turn
  - Turning requires more than a single step (gait step, not time step)
  - Curves of increasing curvature are tested in both directions



From paper

AUTONOMOUS PEDESTRIANS

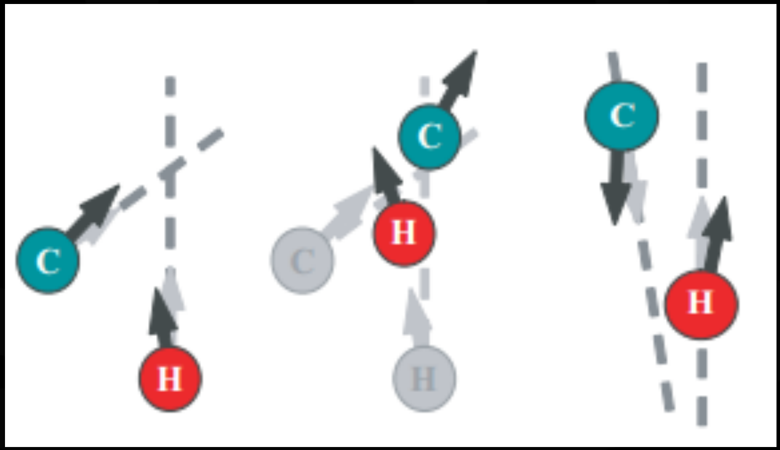
- Maintain separation
  - Only considers “temporary crowd”
    - Nearby agents moving with similar velocity
    - $f_{ij} = \frac{r_i}{|\vec{p}_{ij}| - d_{min}} \hat{p}_{ij}$
    - Got some mathematical problems



From paper

AUTONOMOUS PEDESTRIANS

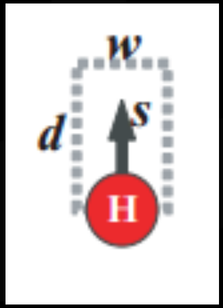
- Avoid oncoming pedestrians
  - Classifies potential collisions with non-temporary crowd members
    - Cross collisions
    - Head-on collisions
  - Considers most "imminent"
    - Turns from head-on
    - Changes speed for cross collisions



From paper

AUTONOMOUS PEDESTRIANS

- Avoid “dangerously” close pedestrians
  - Safety catch for when the previous two rules fail
  - If another pedestrian is in the safety zone:
    - Stop as quickly as possible
    - Turn away
    - Start again when it appears clear



AUTONOMOUS PEDESTRIANS

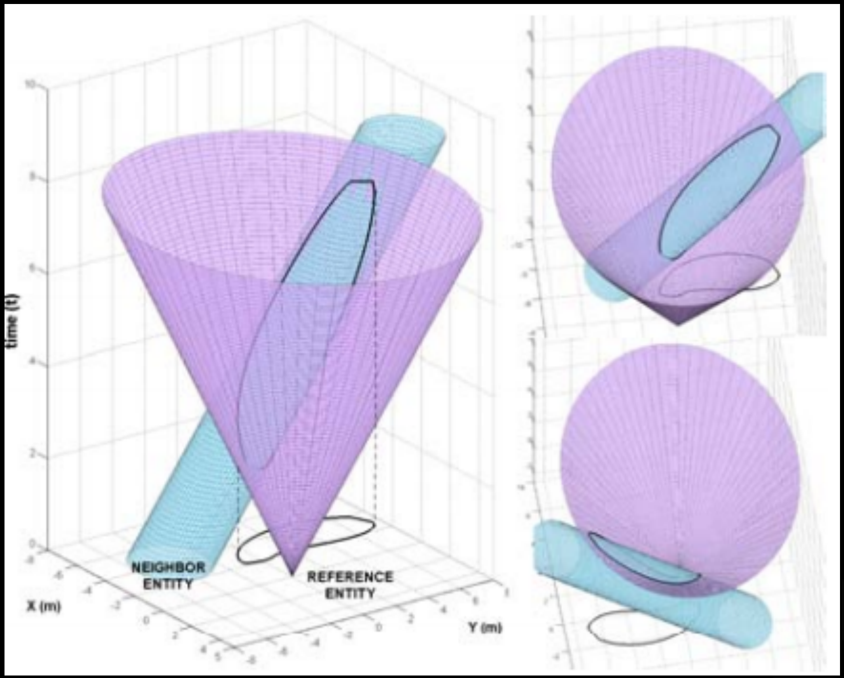
- Validate against obstacles
  - Inter-agent rules can lead to obstacle collisions
  - The current velocity is validated against obstacles
  - Throws out agent-responses
  - Applies voodoo to know when slowing should occur
    - (Not described in the paper)

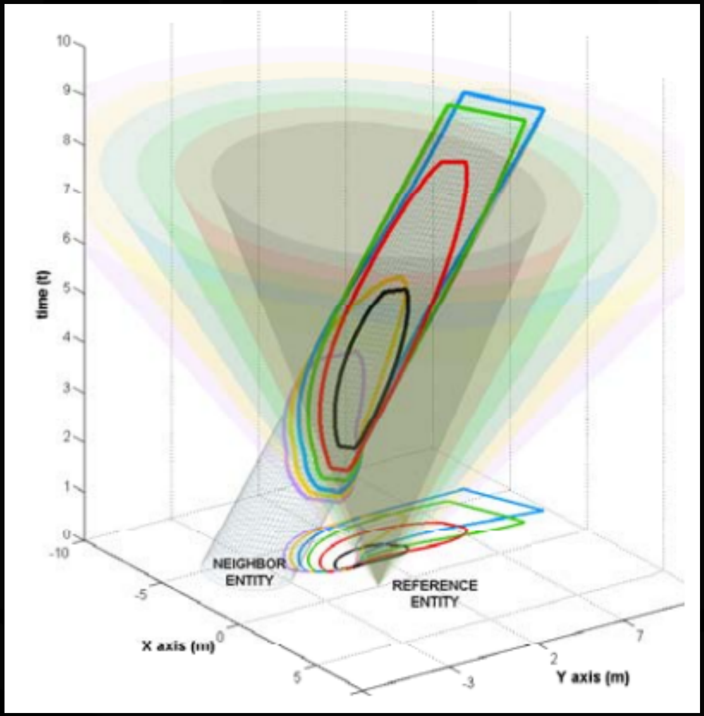
VELOCITY-SPACE MODELS

- Performs optimization in geometric space using optimization techniques
  - Here at UNC we primarily use models of this type

VELOCITY-SPACE MODELS

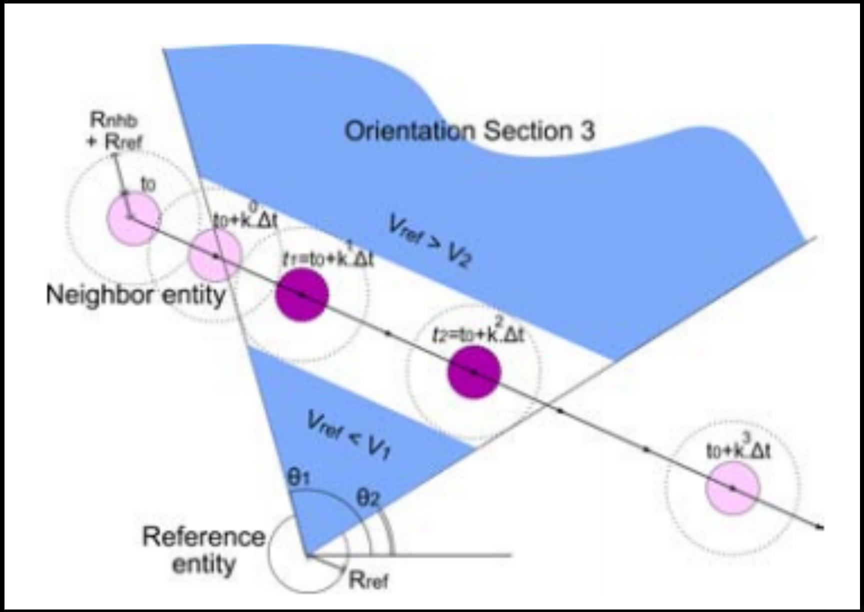
- Paris et al., 2007

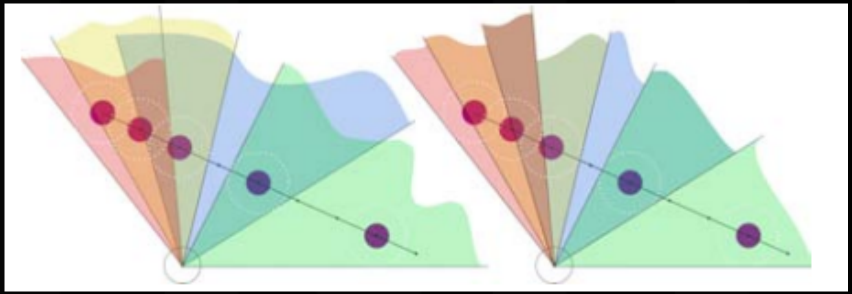




VELOCITY-SPACE MODELS

- Paris et al., 2007



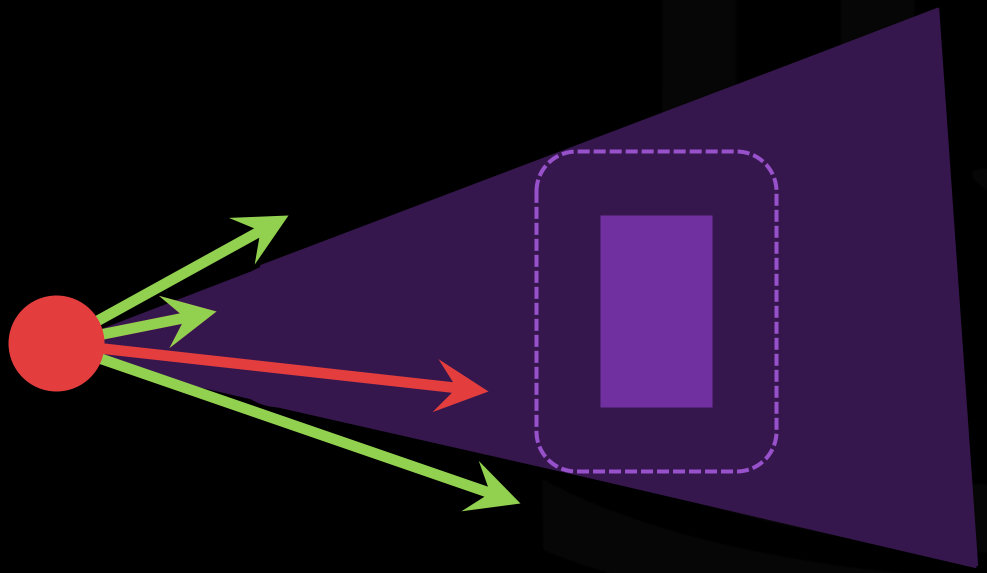


VELOCITY-SPACE MODELS

- Paris et al., 2007
  - Response is selected from the region with the lowest cost
  - Cost is minimal where:
    - Section speed is close to desired speed
    - Section orientation is close to desired direction
    - Acceleration is limited (related to previous rules)
    - Sections based on near time are more important

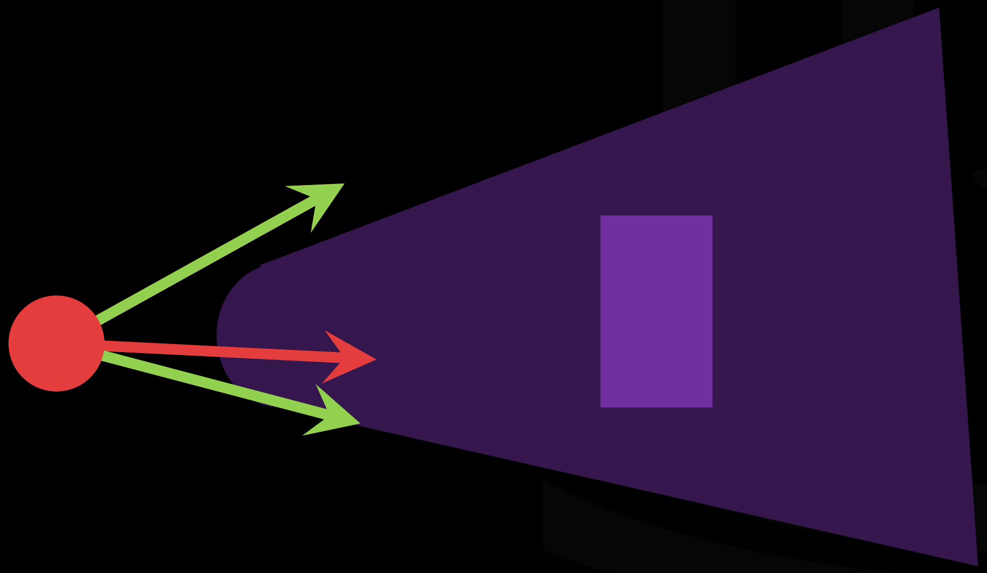
VELOCITY OBSTACLES

- A set of velocities which will lead to an inevitable collision.



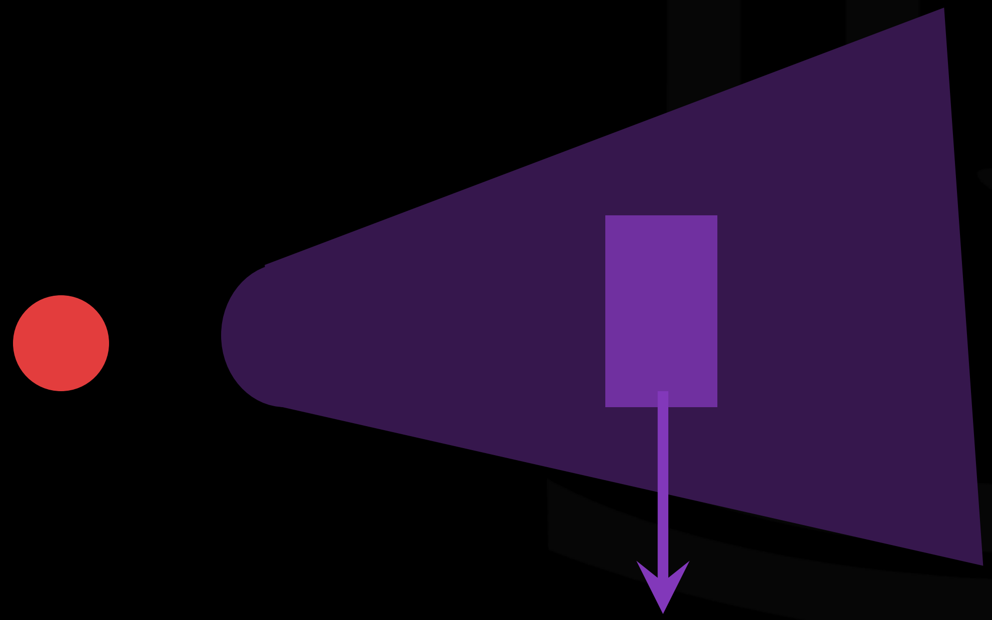
VELOCITY OBSTACLES

- Navigate by selecting "best" velocity outside of the obstacle.



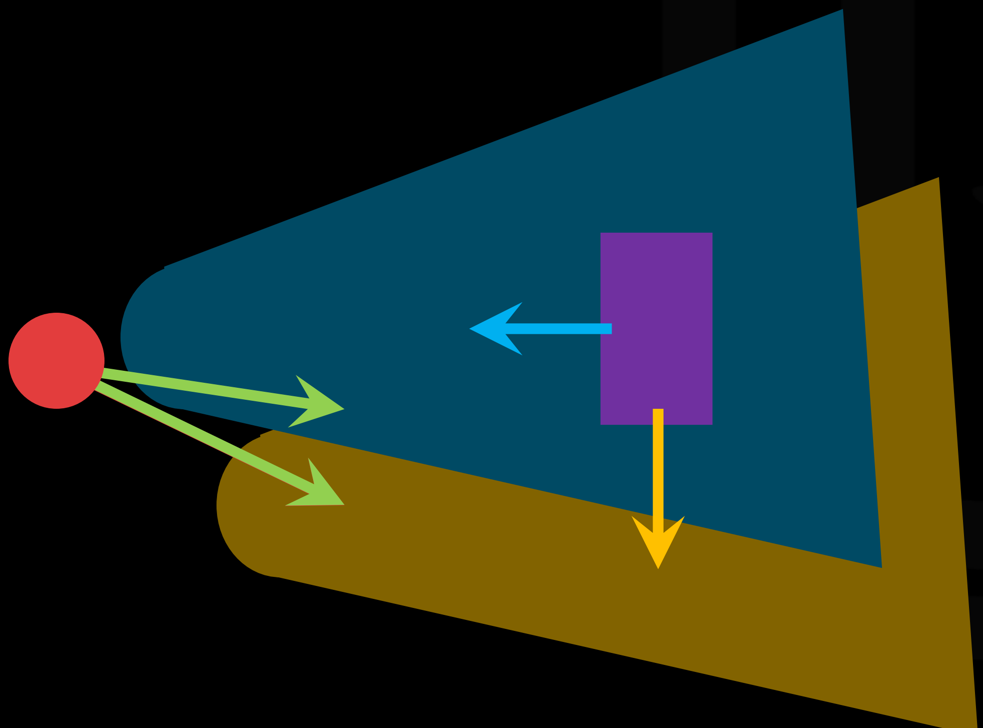
VELOCITY OBSTACLES

- Velocity obstacle for moving objects is translated by that object's velocity.
- This is the original VO formulation [Fiorini & Schiller 1998].



VELOCITY OBSTACLES

- Predicting responsive obstacles

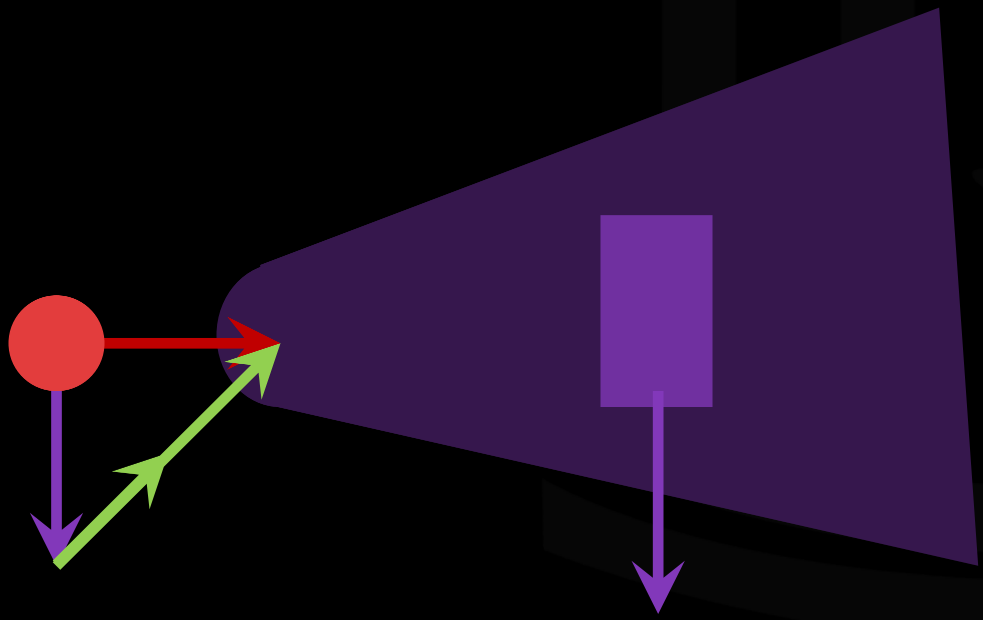


VELOCITY OBSTACLES

- Reciprocal Velocity Obstacles (RVO) - van den Berg, et al., 2008
  - Assume:
    - Each agent is responsive
    - Each agent will take an equal share to avoid collision

VELOCITY OBSTACLES

- RVO

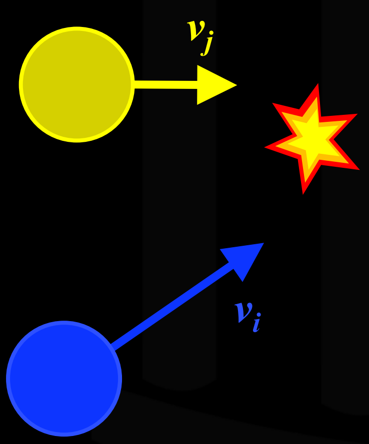


VELOCITY OBSTACLES

- RVO
  - It still assumes that it accurately predicts the other agent's future velocity
  - If the other agent has OTHER constraints that prevent it from taking the expected velocity, the assumption is broken
  - That brings us to Optimal Reciprocal Collision Avoidance (ORCA) – van den Berg, et al., 2009

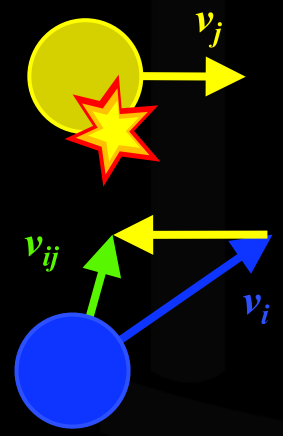
OPTIMAL RECIPROCAL COLLISION AVOIDANCE (ORCA)

- Identify a collision
  - Linear extrapolation (constant velocity)



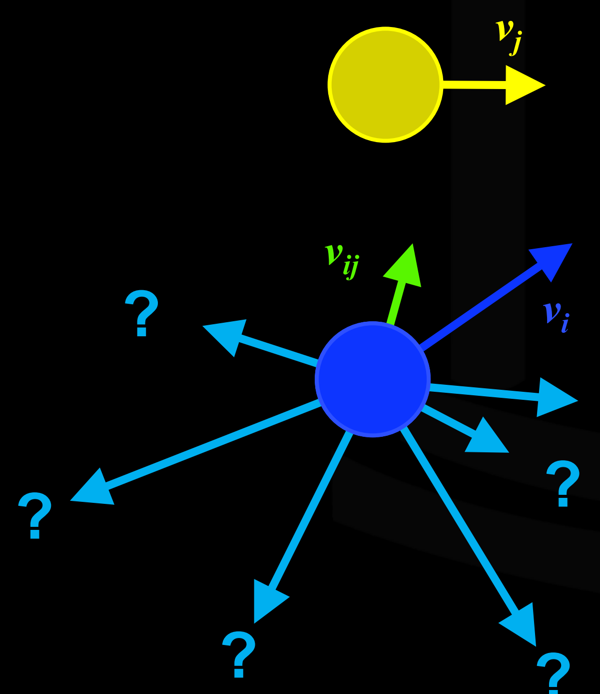
OPTIMAL RECIPROCAL COLLISION AVOIDANCE (ORCA)

- Identify a collision w.r.t. relative velocity and position
  - Linear interpolation (constant velocity)



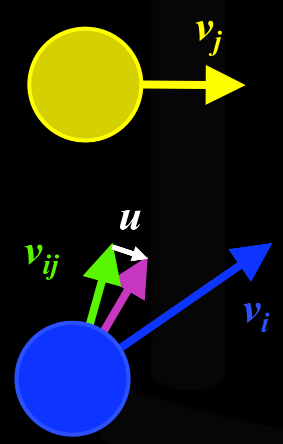
OPTIMAL RECIPROCAL COLLISION AVOIDANCE (ORCA)

- Find alternate, collision-free relative velocity
  - Which one?



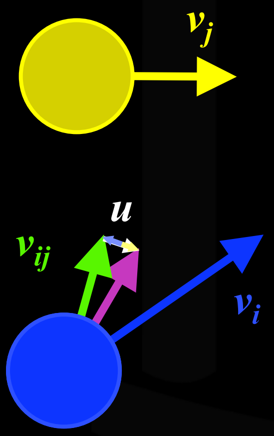
OPTIMAL RECIPROCAL COLLISION AVOIDANCE (ORCA)

- ORCA finds the relative velocity that requires the *smallest* change to the current relative velocity
  - $\boldsymbol{u}$ is the change vector



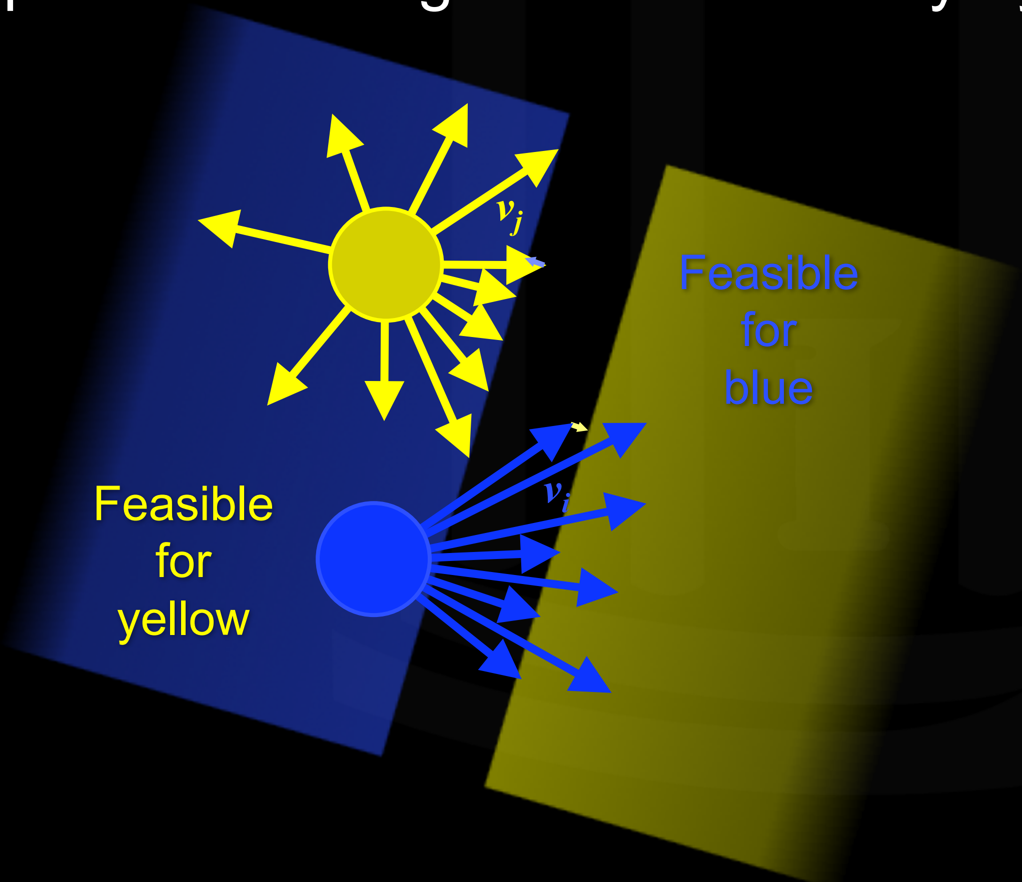
OPTIMAL RECIPROCAL COLLISION AVOIDANCE (ORCA)

- Share the displacement *equally* between the two agents



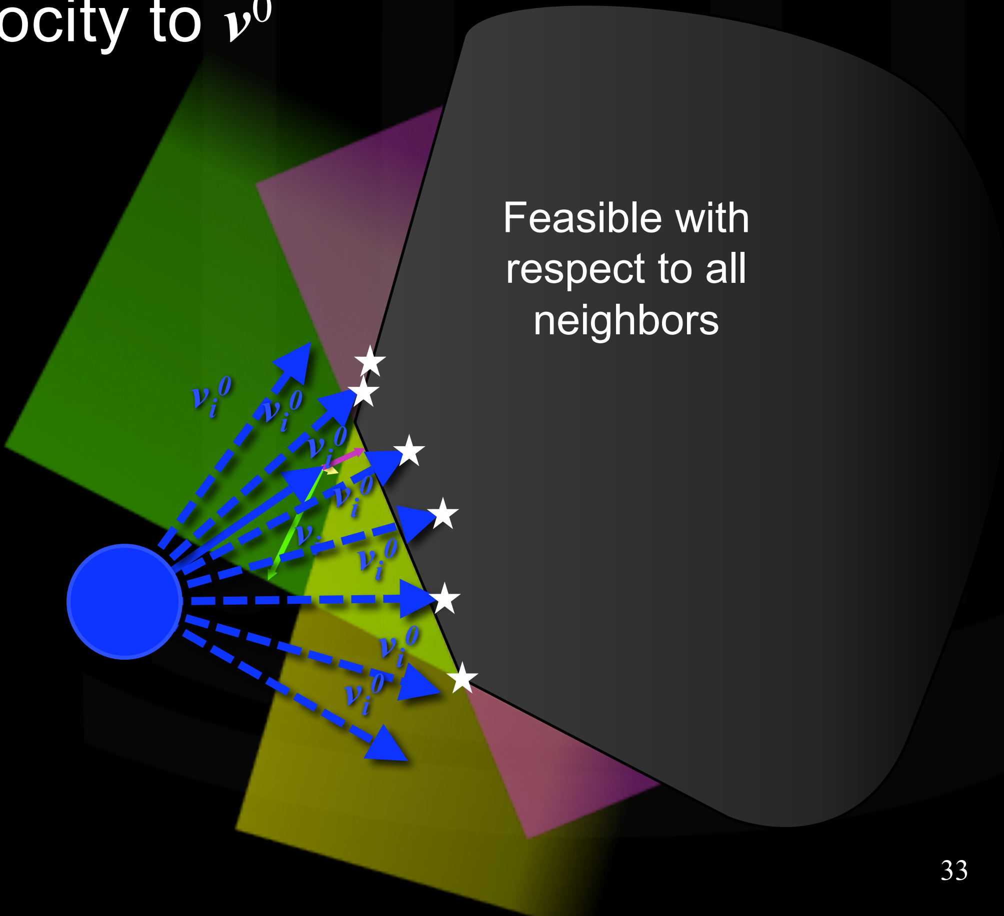
OPTIMAL RECIPROCAL COLLISION AVOIDANCE (ORCA)

- The change in velocity is enforced with a half-plane constraint
- All feasible pairs will change relative velocity by at least u



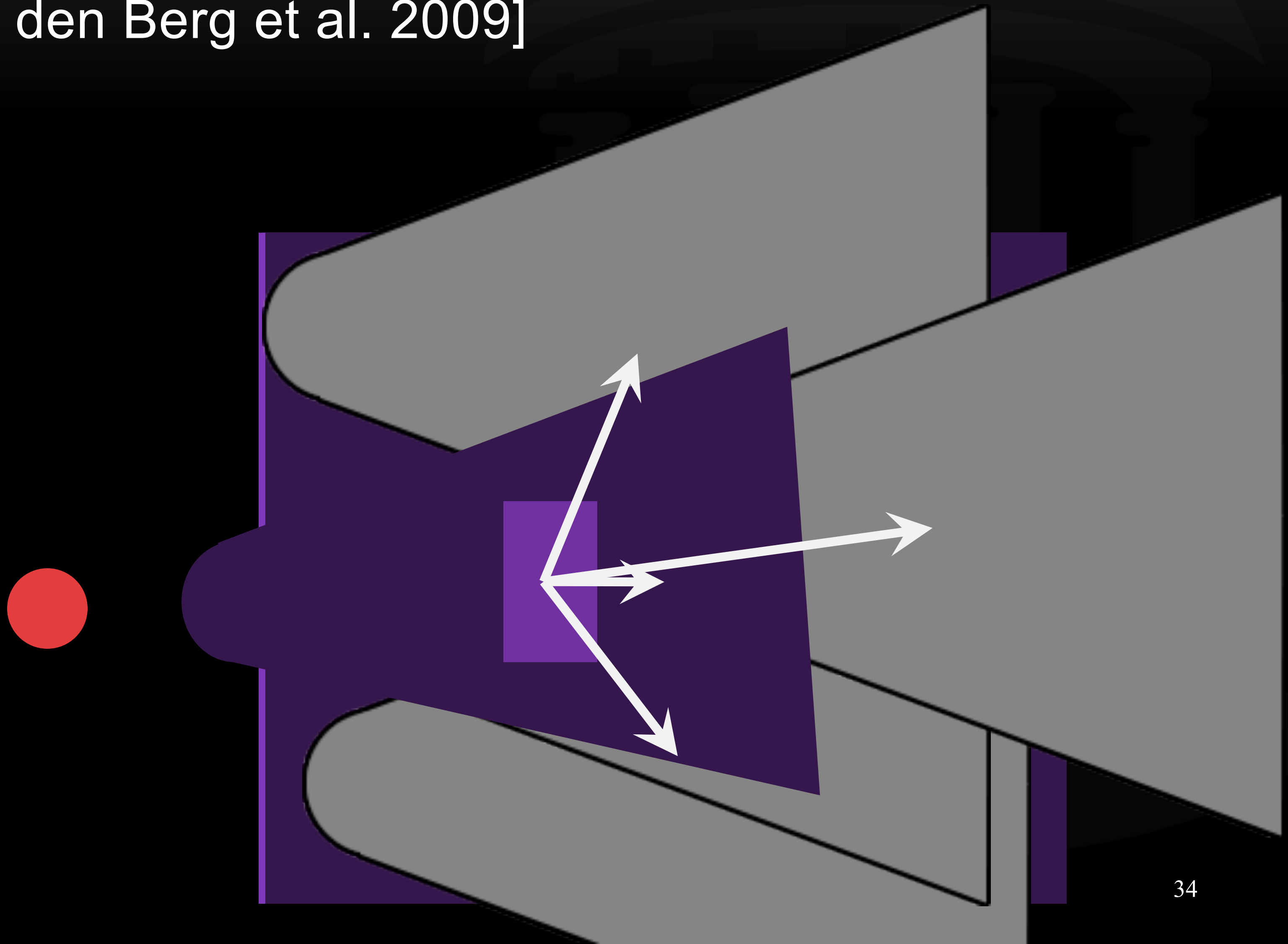
OPTIMAL RECIPROCAL COLLISION AVOIDANCE (ORCA)

- Multiple neighbors form multiple, simultaneous constraints
- Nearest feasible velocity to $\boldsymbol{v}^0$



OPTIMAL RECIPROCAL COLLISION AVOIDANCE (ORCA)

- [van den Berg et al. 2009]



VISION-BASED

- Ondrej et al., 2010
  - Based on planning in "vision" space
    - Similar to optical flow
    - Detecting how quickly things change size and heading
  - http://www.youtube.com/watch?feature=player_embedded&v=586qhaDwr24

AGGREGATE CROWDS

- Narain, et al., 2009
  - Solves for velocity based on density constraints
  - Creates velocity and density fields
    - Projects preferred velocity onto the field and solves the flow such that maximum density is never exceeded
  - http://www.youtube.com/watch?v=pqBSNAOsMDc
  - In principle, still similar to previous pedestrian models

CONTINUUM CROWD

- Treuille et al., 2006
  - Does *not* use the global-local decomposition
  - Solves globally at each time step w.r.t. dynamic entities
  - http://www.youtube.com/watch?v=lGOvYyJ6r1c

CONTINUUM CROWD

- Treuille et al., 2006
  - Computes a "unit-cost" field

$$\int_P C\,ds, \quad \text{where} \quad C \equiv \frac{\alpha f + \beta + \gamma g}{f}$$

  - Minimizes
    - Path length
    - Travel time
    - Discomfort
  - A true potential field model

CONTINUUM CROWD

- Treuille et al., 2006
  - Assumes limited number of unique groups
    - Groups share
      - Goal
      - Preferred speed
      - Discomfort fields

QUESTIONS?