CROWD SIMULAION AND ANALYSIS

Mentor: Prof. Shankar Prawesh

Group Members:

Deepanshu (210311)

Rishav Dev (210847)

Amrit Kalash (210126)

Ritaban Ghosh (210858)

Vedant Maulekhi (211156)

Shiva Shrivastava(210979)

Introduction

Background

The development of robust multi-agent navigation algorithms is critical for ensuring smooth and collision-free movement in complex environments. This project focuses on the application of the Optimal Reciprocal Collision Avoidance (ORCA) framework to achieve these goals.

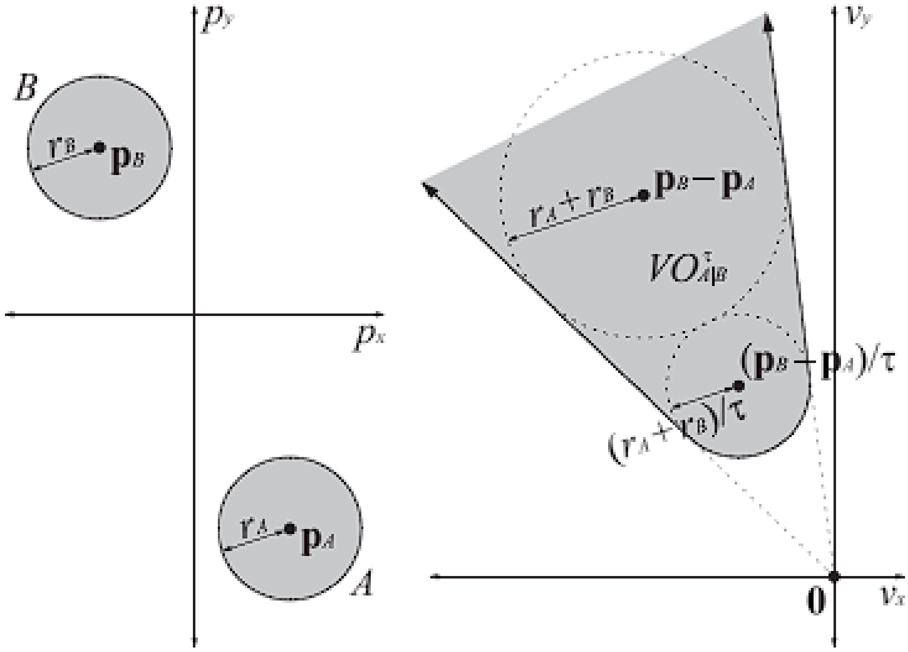
Objective

- **Develop a multi-agent navigation algorithm using ORCA** to enable smooth, collision-free movement among agents.
- **Create an entropy-based metric** for comparing real-world crowd data with simulation results to validate the accuracy of the simulator.

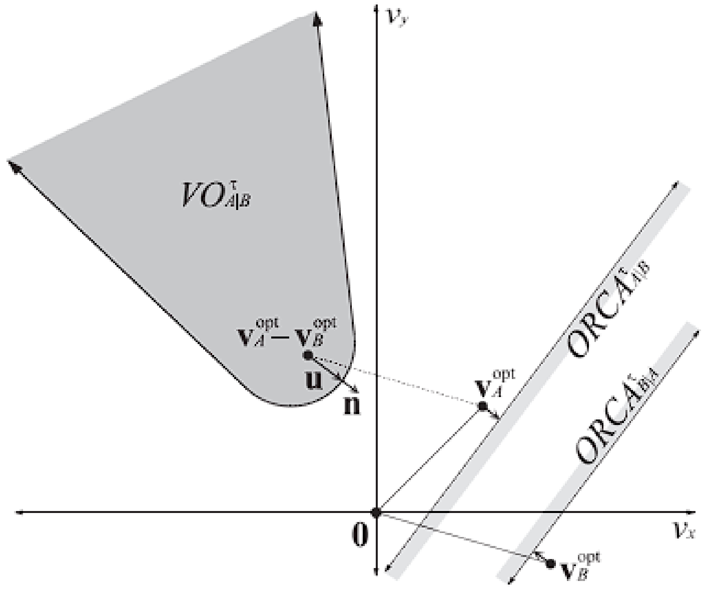
Approach

Algorithm Development

- **Encoding Advanced Physics:** The algorithm integrates advanced physics principles to ensure collision-free navigation. Linear programming techniques were used to solve the collision avoidance problem efficiently.



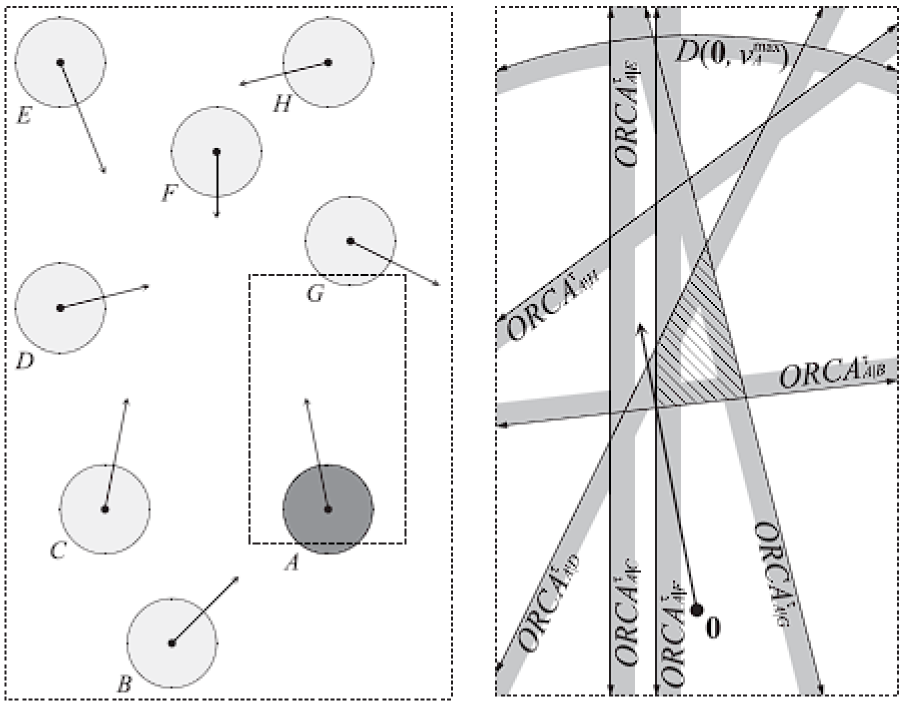
When two bodies enter a potential collision range, they determine the appropriate velocity to prevent the collision by computing their relative velocity and relative position.



The velocity obstacle $VO^{\tau}_{A/B}$(gray) can geometrically be interpreted as a truncated cone with its apex at the origin (in velocity space) and its legs tangent to the disc of radius $r_A + r_B$ centered at $p_B - p_A$. The amount of truncation depends n the value of τ; the cone is truncated by an arc of a disc of radius $(r_A + r_B)/\tau$ centered at $(p_B - p_A)/\tau$. The velocity obstacle shown here is for τ = 2.

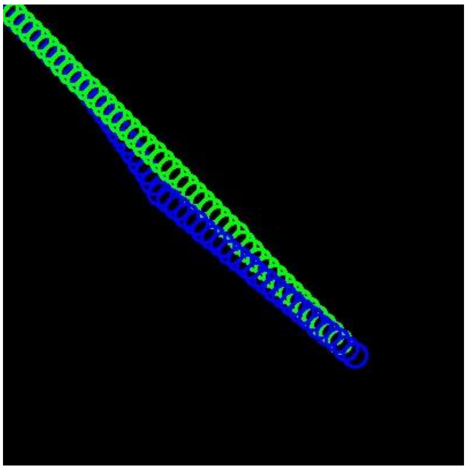
Let v_A and v_B be current the velocities of robots A and B, respectively. The definition of the velocity obstacle implies that if v_A - v_B ∈ $VO^{\tau}_{A/B}$, or equivalently if v_B - v_A ∈ $VO^{\tau}_{B/A}$, then A and B will collide at some moment before time τ if they continue moving at their current velocity. Conversely, if v_A - v_B ∈ $VO^{\tau}_{A/B}$, robot A and B are guaranteed to be collision-free for at least τ time.

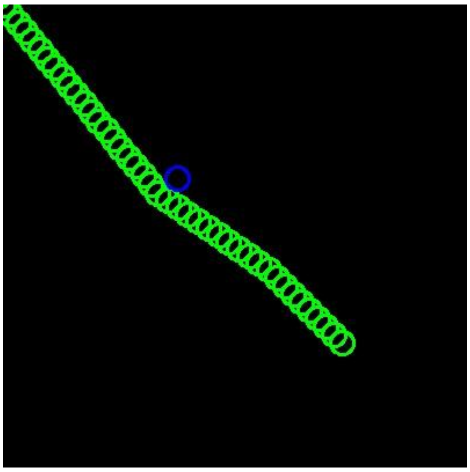
The object assesses whether its velocity falls within the $VO^{\tau}_{A/B}$. If it does, a collision is likely; if not, no collision will occur. If the velocity is within this collision area, the object then calculates the minimum velocity required to avoid the collision.



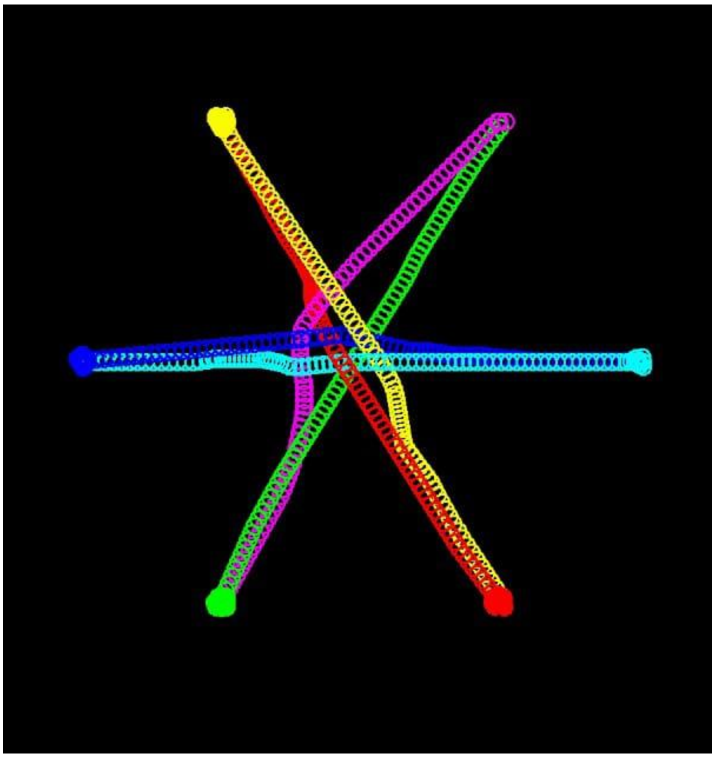
The object calculates the ORCA line for each object within its collision range, selects the optimal velocity from these lines, and then moves in the direction of the chosen velocity.

Results of ORCA





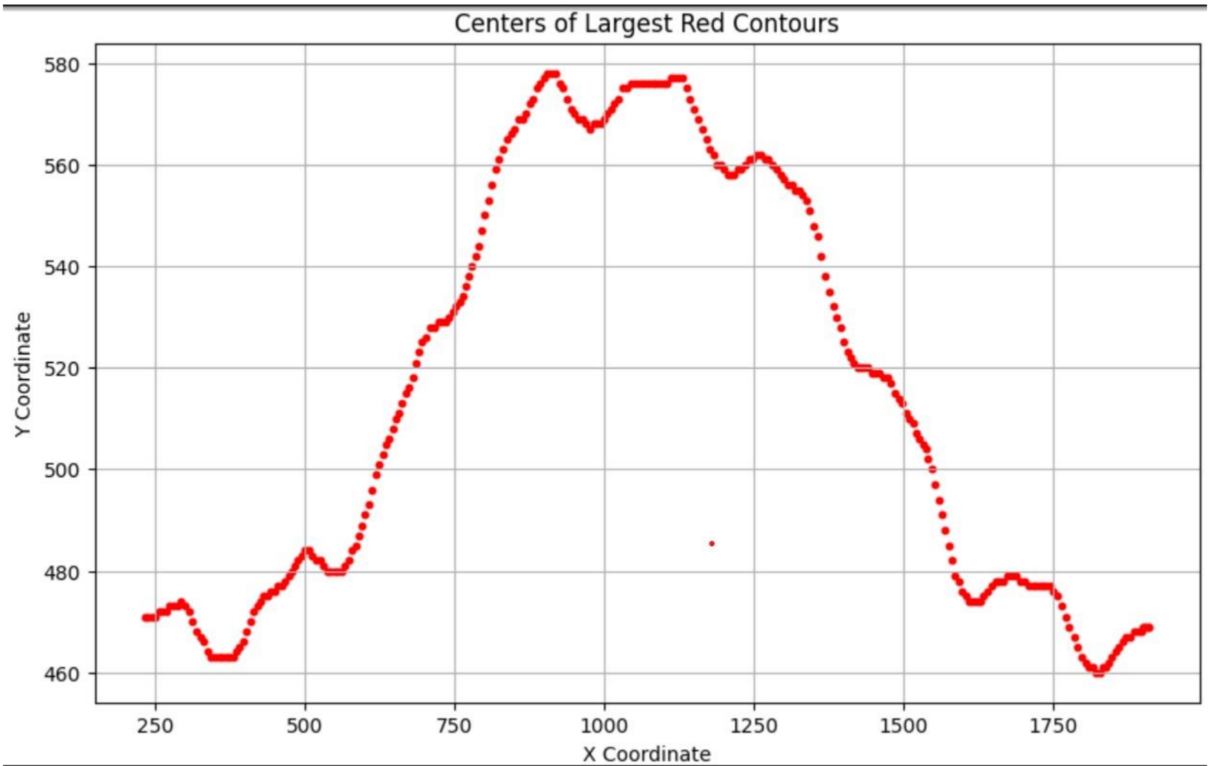
- The model successfully demonstrated collision avoidance in various scenarios. In the first image, two individuals are shown effectively avoiding a head-on collision as they approach each other. In the second image, a person is seen navigating around a stationary object, avoiding a potential collision.



The above image depicts how six persons located symmetrically avoids collision when they all want to exchange their positions with the opposite person.

Data Extraction and Processing

- **Data Extraction:** Moving objects were tracked using contour detection in video footage captured from a lab setup. OpenCV was utilized for this purpose.



Entopy Metric

- **State Prediction:** Bayesian inference and the Expectation-Maximization (EM) algorithm were applied to predict simulation states from noisy crowd data.

  - The EM algorithm was applied to estimate unknown parameters from incomplete or noisy crowd data. The algorithm iteratively performed two steps: the **Expectation step (E-step)**, where it calculated the expected value of the log-likelihood function, and the **Maximization step (M-step)**, where it updated the parameters to maximize the expected log-likelihood.
  - This approach enhanced the predictive accuracy of simulation states by providing a structured method for handling missing data or uncertainties within the observed crowd movements.
  - By leveraging the EM algorithm, the model was able to refine state estimations, resulting in a more accurate representation of real-world crowd dynamics.**2.3 Enhanced State Estimation**

- **Ensemble Kalman Smoothing:** Implemented to improve state estimation robustness in high-dimensional environments, which enhances the accuracy of the simulations.

  - The EnKF was used to improve the robustness of state estimation in high-dimensional environments. This method combined multiple ensemble members to represent the distribution of possible states, effectively handling non-linearities and uncertainties.

- The filter continuously updated the simulation states by assimilating real-world data, which ensured that the predicted motion of agents closely followed observed trajectories.
- The EnKF provided smoother state transitions, improving the overall reliability of the multi-agent navigation system, especially in complex and dynamic environments.

Results of entropy metric

The following results were obtained for the entropy metric after repeated iterations of convergence:

Entropy metric after iteration 1 = 1.3282816400292399

Entropy metric after iteration 2 = 1.2035543653604572

Entropy metric after iteration 3 = 1.3557207670255824

Entropy metric after iteration 4 = 1.1621464360649405

Entropy metric after iteration 5 = 1.3434912647298427

Entropy metric after iteration 6 = 1.5600105913876128

Entropy metric after iteration 7 = 1.7458741049859283

Entropy metric after iteration 8 = 1.069533995566994

Entropy metric after iteration 9 = 1.2515555498264348

Entropy metric after iteration 10 = 1.2964829808738614

Entropy metric after iteration 11 = 1.2635641513339149

Entropy metric after iteration 12 = 1.6932092928684765

Entropy metric after iteration 13 = 1.248366493495463

Entropy metric after iteration 14 = 1.4039417270903414

Entropy metric after iteration 15 = 1.3236598733198406

Average of the last 15 entropy metrics: 1.3499595489305953

Entropy metric after iteration 16 = 1.3132330210837102

Average of the last 15 entropy metrics: 1.3489563076675601

Entropy metric after iteration 17 = 1.1184210743369132

Average of the last 15 entropy metrics: 1.3432807549326573

Entropy metric after iteration 18 = 1.1780132314023692

Average of the last 15 entropy metrics: 1.3314335858911093

Entropy metric after iteration 19 = 0.9020433942902591

Average of the last 15 entropy metrics: 1.3140933831061308

Entropy metric after iteration 20 = 1.0272273070093387

Average of the last 15 entropy metrics: 1.2930091192580975

Entropy metric after iteration 21 = 1.7751476596921227

Average of the last 15 entropy metrics: 1.3073515904783979

Entropy metric after iteration 22 = 1.6290828932027104

Average of the last 15 entropy metrics: 1.29956550969285

Entropy metric after iteration 23 = 1.0992605188725406

Average of the last 15 entropy metrics: 1.3015472779132198

Entropy metric after iteration 24 = 1.454625169358133

Average of the last 15 entropy metrics: 1.3150852525486665

Entropy metric after iteration 25 = 1.4232556315679181

Average of the last 15 entropy metrics: 1.323536762594937

Entropy metric after iteration 26 = 1.3090885420098912

Average of the last 15 entropy metrics: 1.3265717219733353

Simulation Performance

- The entropy-based metric achieved a score of **1.32**, indicating strong performance of the simulator. This score reflects the accuracy of the simulation in replicating real-world crowd dynamics.

Validation

- The developed algorithm and metric were validated against real-world data, demonstrating reliable and accurate performance. This validation underscores the effectiveness of the approach and the robustness of the state estimation.

Metric Validation

- The entropy metric offers a valuable tool for assessing the accuracy of simulations. The achieved score of 1.32 is indicative of strong alignment between simulated and real-world data.

Conclusion

The project successfully developed a multi-agent navigation algorithm using the ORCA framework, achieving a high level of accuracy in simulation performance. The use of advanced techniques in data extraction, state prediction, and estimation significantly contributed to the robustness of the solution.