COMP 790-058: Multi-Agent Simulation for Crowds and Autonomous Driving

Homework #1

Assigned: Sept 20, 2017

Due: 11:59 PM, Oct 04, 2017

A. Setting up the assignment

1. Install Menge
   a. Clone or download from: https://github.com/MengeCrowdSim/Menge
   b. Installation instructions and documentation: http://gamma.cs.unc.edu/Menge/
2. Test Menge by running one of the examples. Refer to instructions at https://github.com/MengeCrowdSim/Menge#running-simulations
3. Other resources for homework:
   a. Download resources from: http://gamma.cs.unc.edu/courses/planning-f17/HW/2017/hw1-resources.zip
   b. Use Menge examples specified in "hw2-examples" for your experiments
   c. Sample code for reading XML and proximity queries can be found in "segmentDist" folder
   d. Refer to "docs" folder for documentation on the roadmap format, obstacle definition etc.
   e. Some sample roadmap results can be found in the folder "sampling-results". Your results may be different, due to sampling strategy and the associated randomness. Note that the roadmaps are named as <scene>_<#samples>_<#links>_<#version>.png

B. Computing Probabilistic Roadmap (PRM) for Global Planning

Given 2D polygonal models of obstacles, you need to compute a roadmap for a circular robot. You need to compute roadmaps for two scenes:

1) hw2-examples/bottleneck.xml
2) hw2-examples/tradeshow.xml

Restrict your sampling within a bounding box, defined by bottom left and top right points in Cartesian coordinates:

hw2-examples/bottleneck : (-30,-15) (30, 15)

hw2-examples/tradeshow : (-27, -33) (27, 33)

You need to generate roadmaps for these environments. We recommend using a uniform subdivision sampling approach for the bottleneck scenario and a pure random sampling scheme for the tradeshow.

Pure Random Sampling : In this case, a predetermined number of valid samples (configurations) are found by randomly sampling the entire scene .

Random Subdivision Sampling: order to densely sample constrained spaces, a sliding window approach may be used. In this case, a square window is moved across the scene. At each position, the pure random sampling technique is used as defined above.

You may use any other sampling strategy as well.

A key issue in generating roadmaps is the implementation of the following queries:

- CLEAR (q)
  - Checks if configuration q is collision free
- LINK(q,q')
  - Checks if the straight-line path between q and q' is collision-free

In case of circular robots, CLEAR corresponds to point-edge distance computation and LINK reduces to edge-edge distance computation. Sample code for these distance/proximity queries is available in the folder "segmentDist".

You may also implement your own version of proximity queries, or download some other code over the WWW (with acknowledgements).

For each scenario, you should generate at least 12 roadmaps:

1. Compute two sets of roadmaps, one set with robot radius = 0.1 m and one with radius = 0.2 m
2. Vary the follow parameters:
   a. At least 3 values of valid samples in the range [5000, 15000]
   b. At least 2 values of max links per node in the range [10,30]
3. Report the timing breakdown between different parts of the algorithm for each computed roadmap
4. What sampling strategies did you use? Comment on their respective advantages and disadvantages? Why is pure random sampling not ideal for the bottleneck scenario?
5. Comment on the advantages and disadvantages of a random sampling (pure or otherwise) approach for path planning in an obstacle dense scene such as the tradeshow.
6. Compare Menge simulations for the tradeshow with ORCA model (-m orca) at timestep 0.1 (-t 0.1) and radius = 0.2m
   a. Using the provided navigation mesh for global planning.
   b. Using any **one** of your computed roadmaps for global planning.
   c. Comment on the observed behaviors in terms of agent's ability to find a path to the goal using each of the two global planners. You may find it easier to evaluate behaviors by following the same agent in both simulations. In the menge visualizer, select an agent by clicking on it. Note: you can set the random seed (-r <int>) and then select the same agent in both simulations for a fair comparison.
   d. Can you think of use cases where a probabilistic roadmap may be better than a navigation mesh?

C. Comparing Local Navigation Algorithms

Before comparing local navigation algorithms, you must familiarize yourself with the basic terminology that is used to describe an "ideal" crowd simulation model. Please read section 1.1.1 (pg 4-7) of the following dissertation:

https://www.cs.unc.edu/~seanc/downloads/PedVO_dissertation.pdf

For the purpose of this assignment, we will use a subjective, qualitative assessment on the simulation's stability. For example, simulations with smoothly moving agents and fewer observed collisions are regarded as more stable.

You will now compare three local navigation models (Helbing, ORCA and PedVO) on the following two simulation scenarios:

1) hw2-examples/4square.xml
2) hw2-examples/ pedModelSwap.xml

1. For each simulation scenario,
    a. Vary the time step. Start at t=0.01 and strategically increase it until you notice qualitative differences. You may even lower the timestep to t=0.001 s for helbing.
    b. Report on the relative stability of the algorithms at the chosen time step. Your answer should clearly indicate the order of stability, and the corresponding time step. For example:

       $S^{orca} > S^{pedvo} > S^{helbing}$ at t= 0.05 s   or
       $S^{orca} > S^{pedvo} = S^{helbing}$ at t = 0.1 s
    c. Report on the computation time for each algorithm. This is the average time per step of simulation (scene update time as reported on the command window).
    d. Include videos for each algorithm at the chosen time step. You should have 3 videos for each scenario.
2. Report and compare the observed behaviors of each algorithm on the two scenarios, with varying time steps.
3. Comment on the efficiency of each algorithm, based on observations of stability across time steps and the recorded per step computation time.
4. Can you think of any quantitative metrics to define the accuracy of a crowd simulation algorithm? Describe in brief.

D. Resources and Submission Information

Notes:

- Ensure consistency in menge simulation by setting the random seed in each run (-r <int>)

Submission:

Your submission should include

1. A report
    f. The computed roadmaps (visualized similar to the included samples). Please name them using the prefix : <scene>_<#samples>_<#links>_
2. Links to videos of simulations for question 6 of part B.
3. Links to videos of six simulations, as required for question 1 of part C.

Questions regarding assignment:

Contact Sahil Narang (sahil@cs.unc.edu) regarding any questions/issues relating to this assignment.

Email the class alias (comp790-058-f17@cs.unc.edu) if you feel the discussion would benefit the rest of the class.

Resources:

- Probabilistic Roadmap Path Planning
    - Notes from Northeastern Univ. http://www.ccs.neu.edu/home/rplatt/cs5335_2015/slides/04-prm.pdf
    - Notes from Columbia Univ. http://www.cs.columbia.edu/~allen/F15/NOTES/Probabilisticpath.pdf
- ORCA
    - Van Den Berg, Jur, et al. "Reciprocal n-body collision avoidance." *Robotics research* (2011): 3-19.
- Helbing
    - Helbing, Dirk, Illés Farkas, and Tamas Vicsek. "Simulating dynamical features of escape panic." *arXiv preprint cond-mat/0009448* (2000).
- PedVO

    Curtis, Sean, and Dinesh Manocha. "Pedestrian simulation using geometric reasoning in velocity space." *Pedestrian and Evacuation Dynamics 2012*. Springer, Cham, 2014. 875-890.