EECS 568 Mobile Robotics HW3
Discrete and Continuous Counting Sensor Model

PoKang Chen

Structure

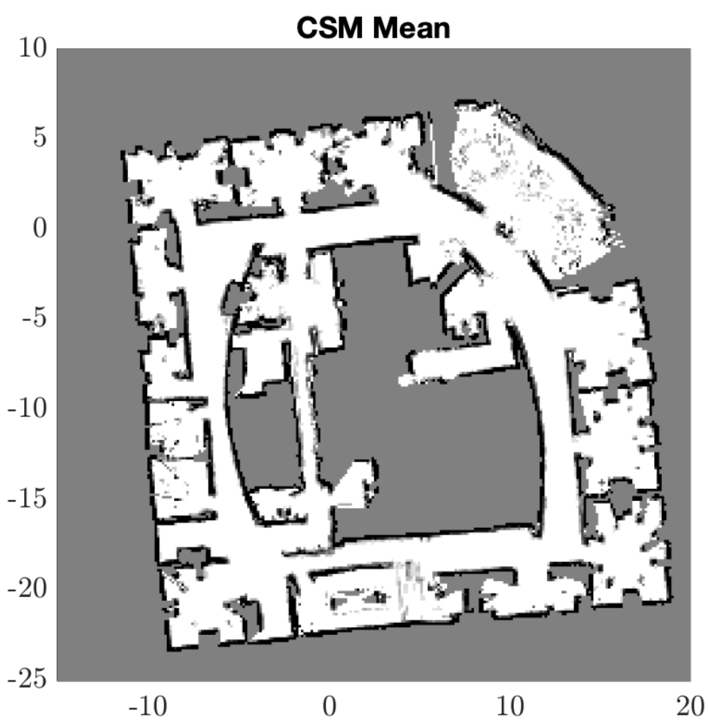
The counting sensor model (CSM) is implemented to gain the occupancy grid map of the area. To construct the map, we first utilize knnsearch to gain the nearest 16 poses(The distance is from query point to the robot location). There are 133 beams per pose of the robot. For these beam, we only use the nearest beam to renew the α for each class.

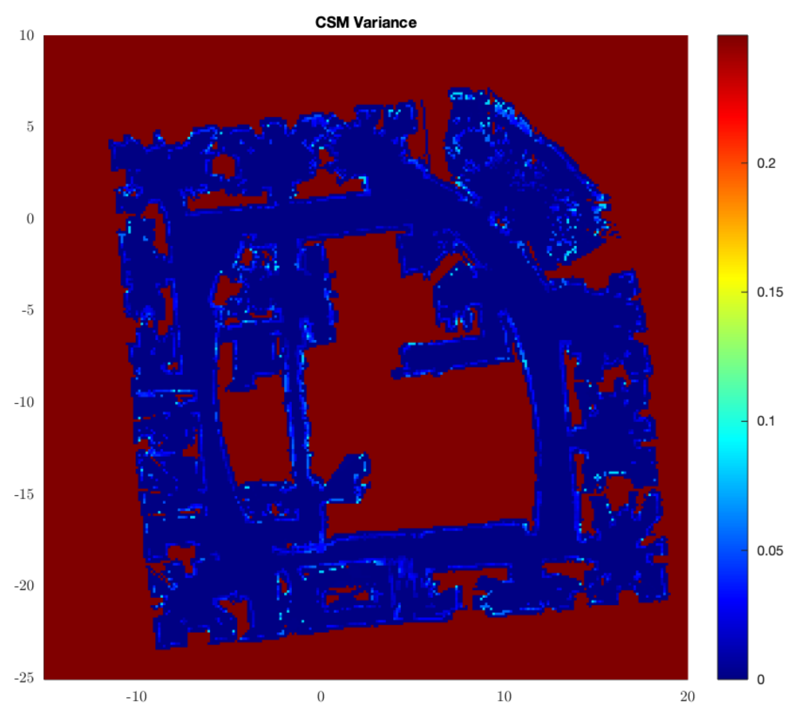
For first two task, there's only two classes for that one class represents the free space and the other represents the occupied space, noticed as α and β.
For the third and the fourth task, the semantic counting sensor model contains 7 classes with one unoccupied class, denoted $\alpha_1 \sim \alpha_7$.

Task 1

A. Methodology of discrete CSM

The first task utilize inverse sensor model to detect whether the hit point is within certain distance and certain geometry pattern from the query point. We then refreshed the α value and β value.





Task 2

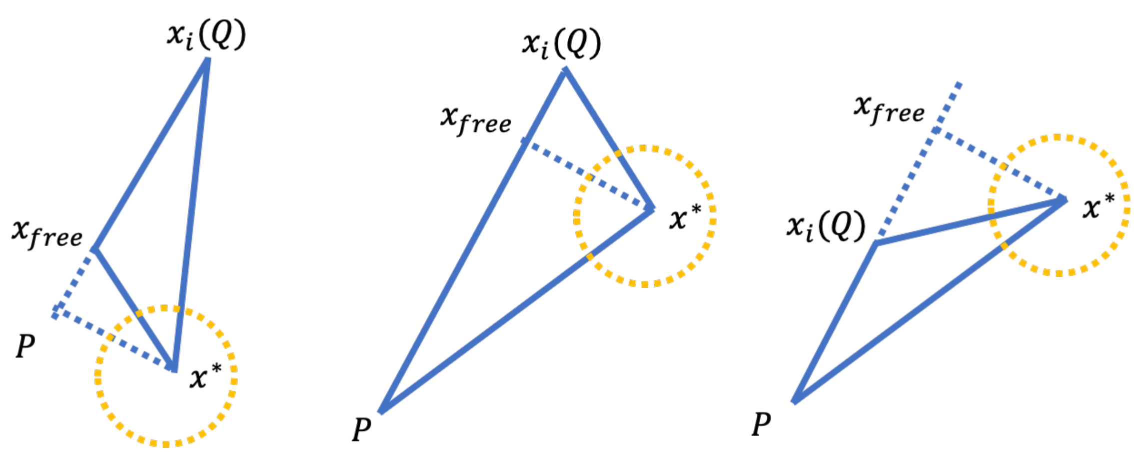
A. Methodology of continuous CSM

For continuous counting sensor model, we use the "projection model" as mentioned in the paper Learning-Aided 3-D Occupancy Mapping With Bayesian Generalized Kernel Inference. x_{free} is gained to decide upon three of the situations. We first decided whether the projection $\overrightarrow{a}$ is within $\overrightarrow{PQ}$ or not. The three situations are depicted as following:

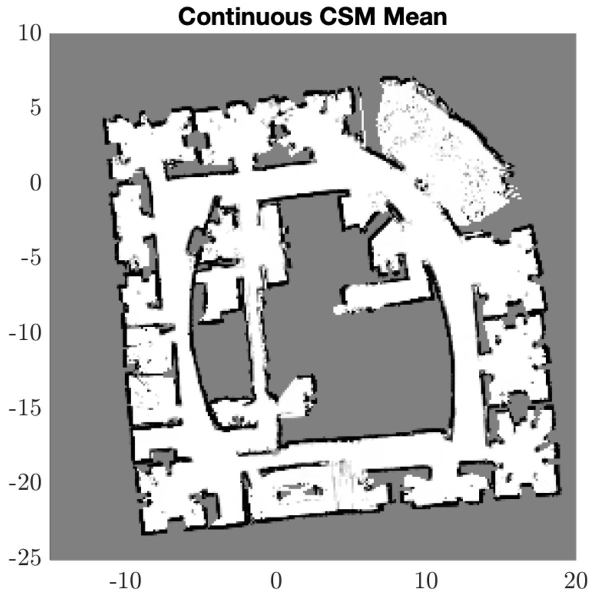
$$x_{free} = \begin{cases} P & \overrightarrow{a} < 0 \\ P + |\overrightarrow{a}|\frac{PQ}{|PQ|} & 0 \leq \overrightarrow{a} \leq \overrightarrow{PQ} \\ Q & \overrightarrow{a} > \overrightarrow{PQ} \end{cases}$$

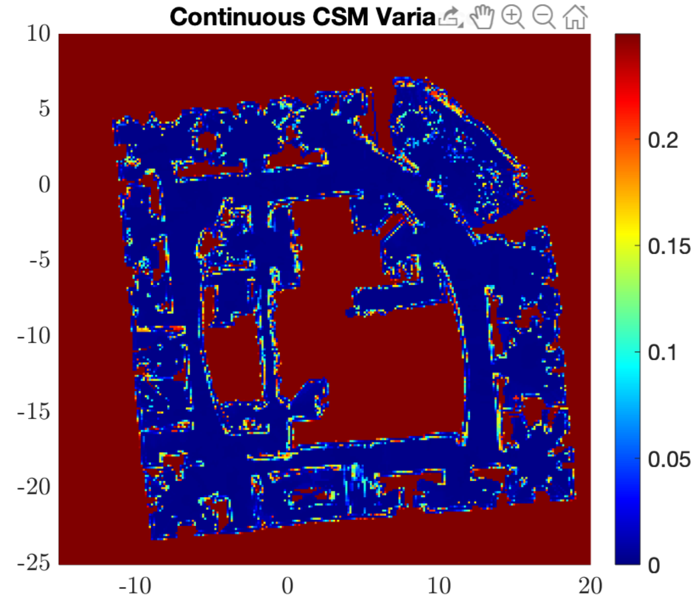
$$\overrightarrow{a} = (\frac{\overrightarrow{Px^*} \cdot \overrightarrow{PQ}}{|\overrightarrow{PQ}|})\frac{\overrightarrow{PQ}}{|\overrightarrow{PQ}|}$$

After we got the relative position of P, x^*, and $x_i(Q)$, we could then decide the x_{free} position. If x_{free} locates before P, we won't refresh any value. If x_{free} locates between $\overline{PQ}$, there are two situations we need to concern. The first one is to decide whether x_i and x^* locate within the circles centered at x^* with radius = l. The two distances d_α, d_β are accordingly $d_\alpha = ||x^* - x_i||$ and $d_\beta = ||x^* - x_{free}||$. When d_α is within l, denoted as $d_\alpha \leq l$, then we update the α with a value k_α, for which k_α is a function denoted as kernel function. Further, we would also update the β value when d_β is within l, denoted as $d_\beta \leq l$. The last condition is that when x_{free} locates after $\overline{PQ}$. We only need to decide whether d_α is within l or not. We then update α value.

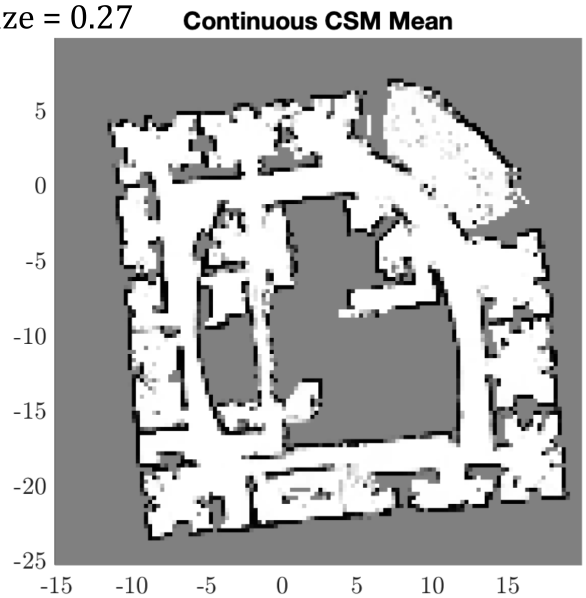


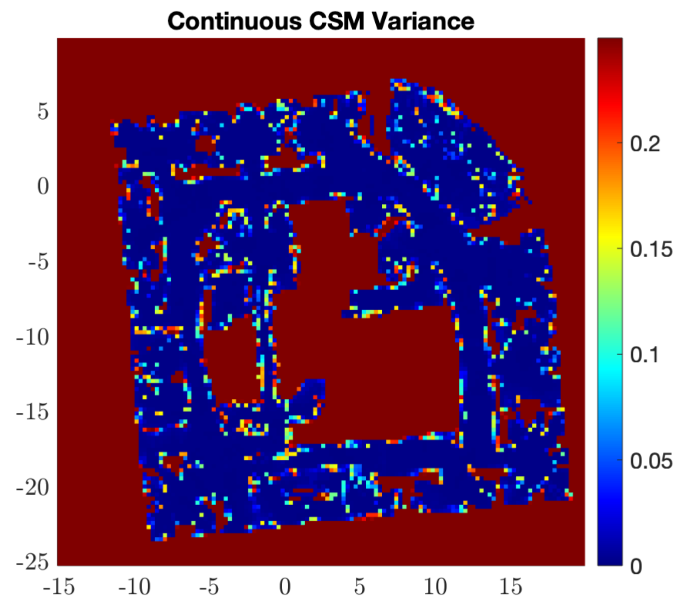
grid_size = 0.135



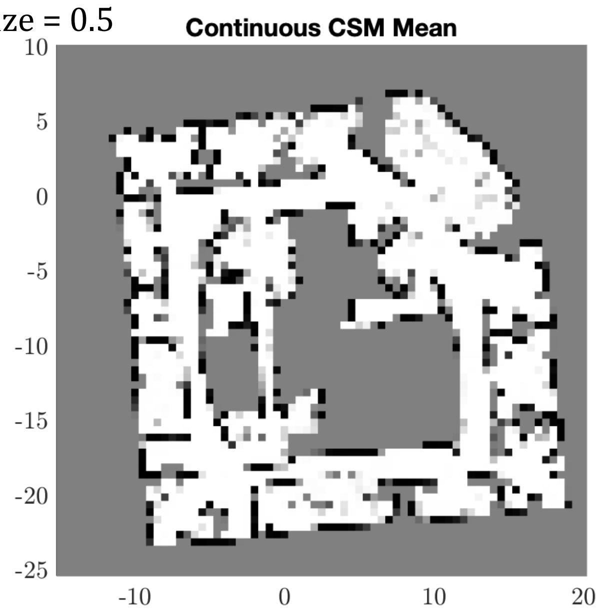


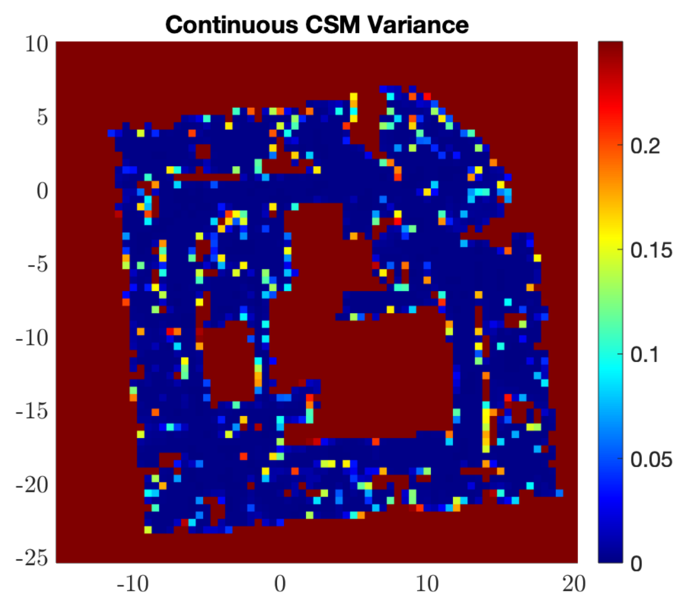
grid_size = 0.27





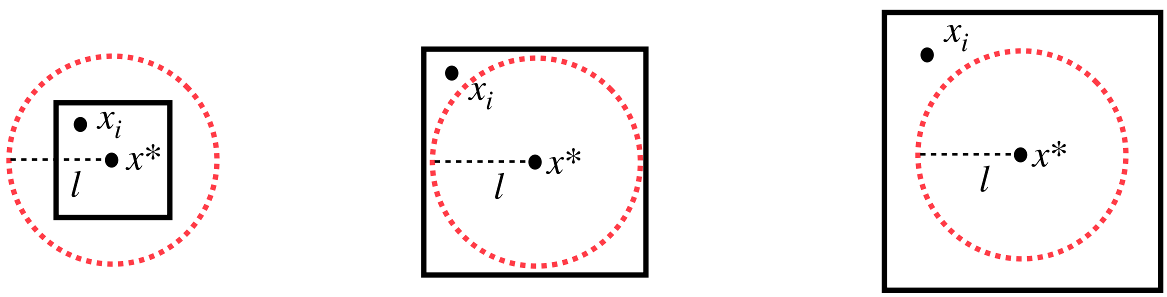
grid_size = 0.5





B. Discussion of different grid size:

The following is the six plots of grid size = [0.135, 0.27, 0.5]. With various resolution, the grid size is different and thus the area per grid covers also differs. This means that there are less query points to loop through even though the value of l doesn't change. As we set $l = 0.2$, if the grid size is larger than 0.2, the procedure might happen to lose some important sample characteristic. As the situation in the plots for $l = 0.27, 0.5$, the resolution is getting worse and several areas are unidentified as the wall or the occupied areas. As depicted in the following graph, the square area is the grid we are dealing with which is centered by the query point. If there exists nearest beam $|\overrightarrow{Px_i}|$ for that x_i locates at certain position within the grid but not in the circle of $r = l$, then it wouldn' renew the α value. This explains why some of the expected boundary cells are classified as free space.



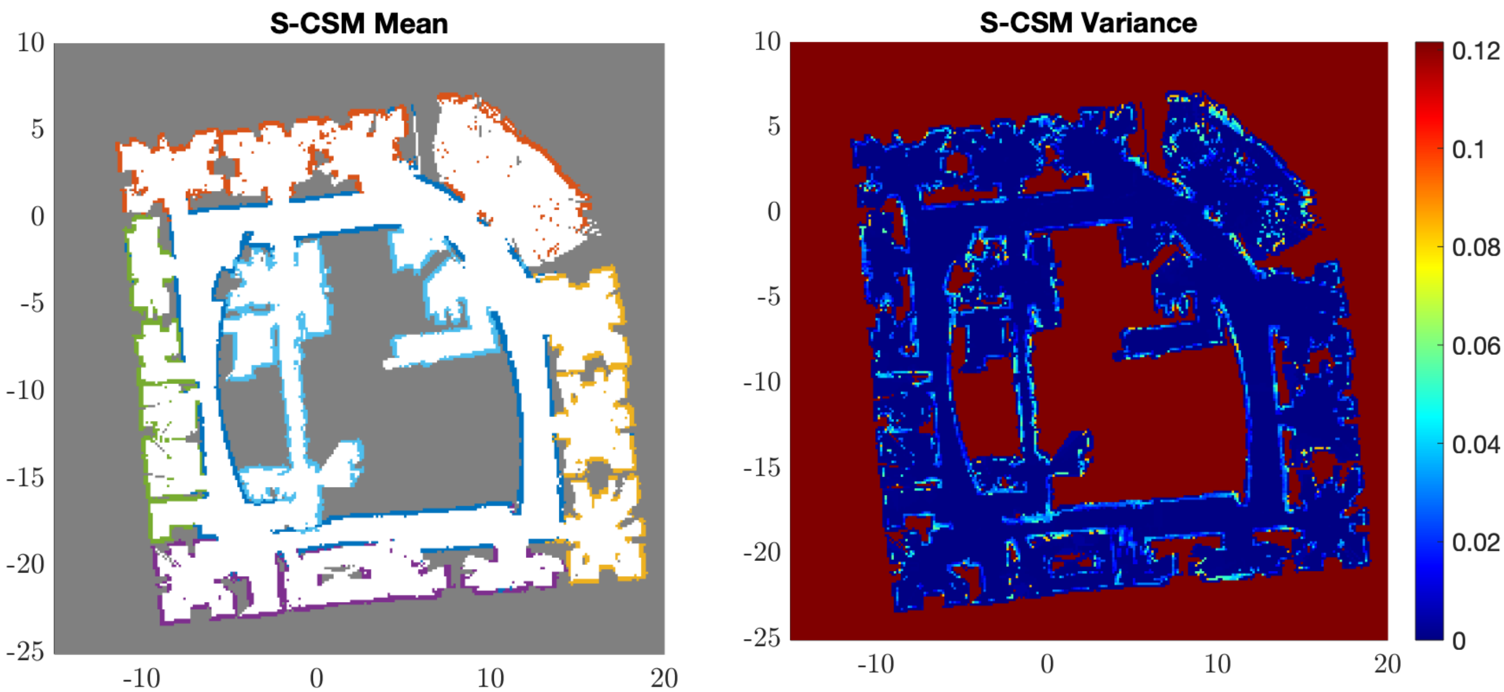
Grid size = 0.135 Grid size = 0.27 Grid size = 0.5

C. Continuous & Discrete CSM:

The difference between continuous and discrete is that continuous method implemented kernel function to determine how big per increment should α and β value increase when falling into the category. The kernel would decrease to zero when the length between $d_\alpha = ||x^* - x_i||$ is bigger than l . When it falls within 0 and l, the value of the kernel would be within 0 and 1. The continuous method constructs a specific circular area for which the value gradually decreases as it gets further from the query points. The gradual increasing process of the kernel value makes the biggest difference between continuous and discrete counting sensor model. Further, the variance of continuous model is obvious and distinguishable and there are also some blurring area at the edge of the walls. The variance difference showed around the walls of variance CSM plot depicts the continuous situation which are more similar to reality world and could better indicate where the true walls are.

Task 3

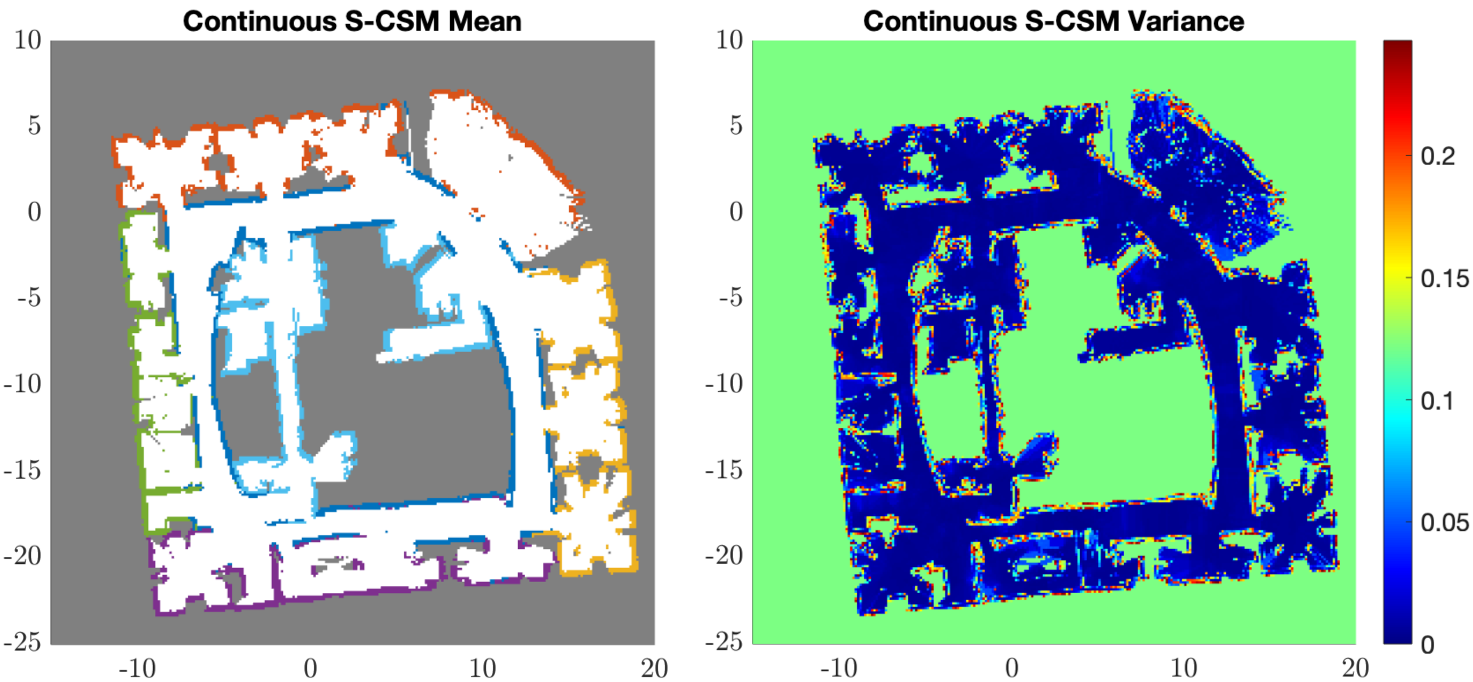
A. Methodology of semantic discrete CSM:



The semantic discrete CSM works just like the discrete CSM excepts that the map not only contains the class of the occupied and unoccupied, the occupied areas are classified into several categories as north rooms, west rooms, east rooms, south rooms, middle rooms, and hallways. In total, there are seven categories including the free space.

Task 4

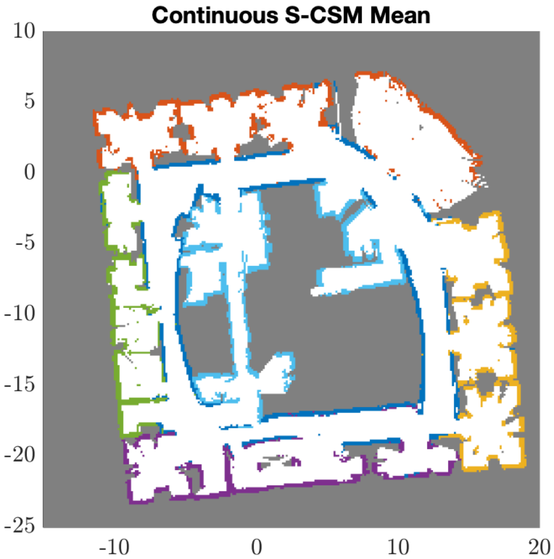
A. Methodology of semantic continuous CSM:

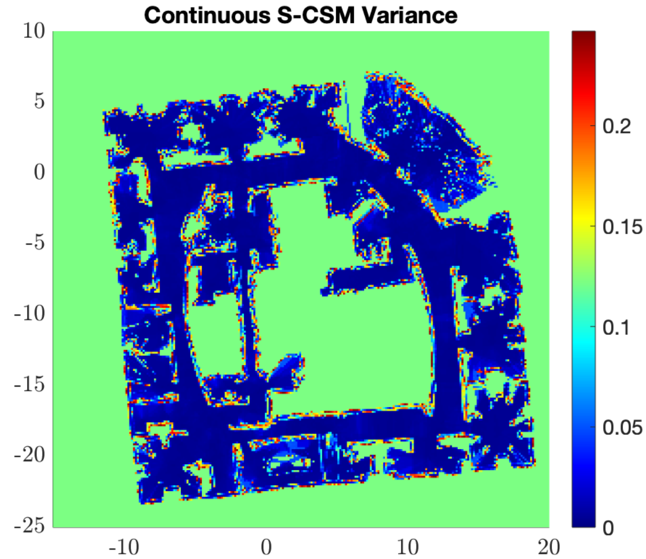


The continuous semantic CSM is also just like the one in task 2. It depicted the different categories of the occupied areas with six different classes. The continuous one better indicates the boundary areas. The reason that the background area of continuous semantic CSM variance is green is that the default value of alphas are set as 0.001 and there are 7 categories. $V \approx 0.12$ when we didn't renew any value of variance.

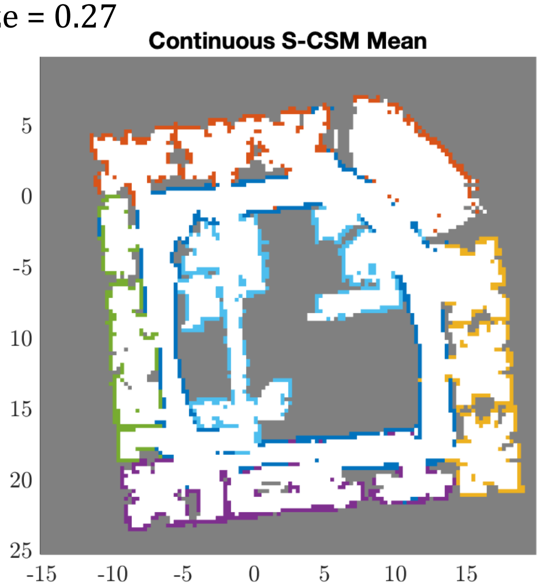
B. Discussion of different grid size:

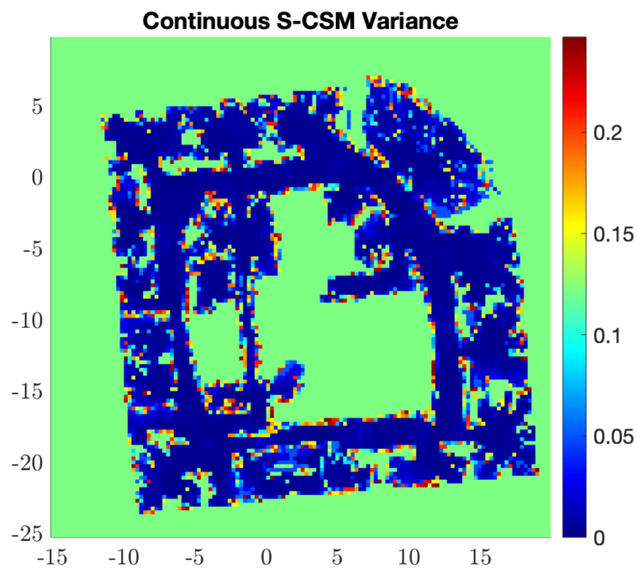
grid_size = 0.135



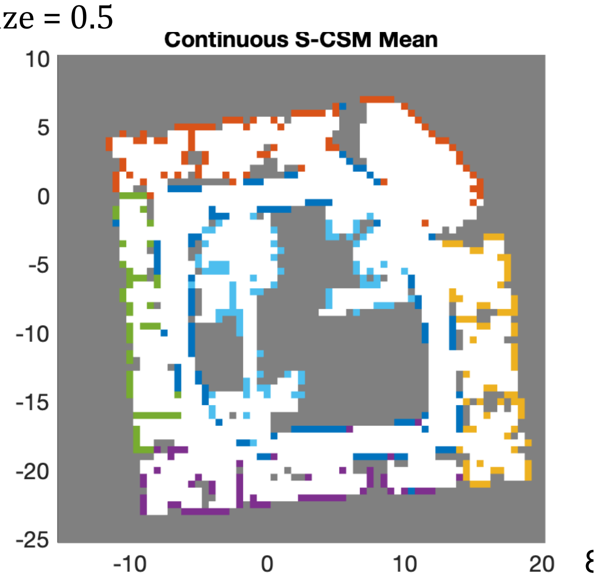


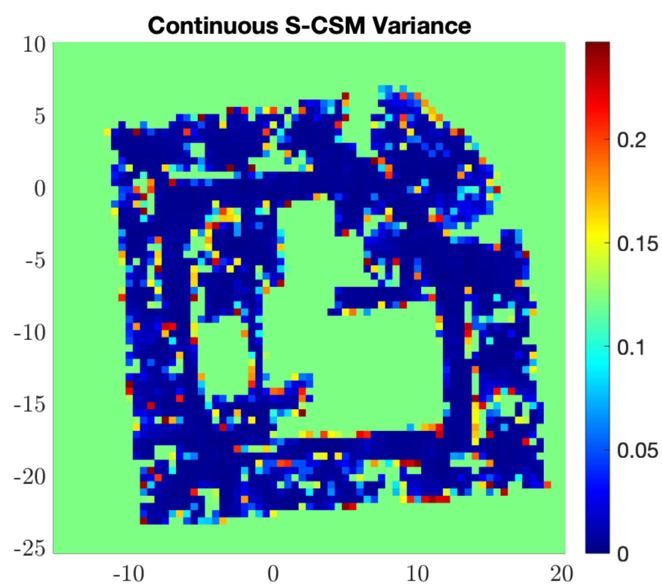
grid_size = 0.27





grid_size = 0.5





The resolution of the map would also influence the depiction of the true objects as mentioned in Task 2. As the grid size increases, it would then miss some of the important points of x_i, which is depicted in the figure of Task2. This could explain why the resolution would influence whether it could successfully and throughly detect the expected boundary. Thus, the mean and variance plots would both miss some important information when the grid size increases from 0.135 to 0.5.

C. Continuous & Discrete semantic CSM:

The continuous semantic CSM would be better than the discrete one for that the biggest difference is the plot of variance. The boundary of discrete semantic CSM variance plot are more blur and undistinguishable than the one in continuous semantic CSM. Further, the one in continuous semantic CSM depicts a certain gradual gradient boundary but not drastically change from one to another. This looks more familiar with the reality.

Provide a discussion on which mapping algorithm(s) should be implemented on the robot and why.

Different situations should fit different plots and methods as mentioned above. Supposed there are a robot with high capacity of computational calculation and would like to gain strong and high resolution maps for that it could finish some intricate action, it would require the continuous CSM with smaller grid size. It would then clearly depicted the obstacles and boundaries. If the computational capacity of a robot is confined to some extend, then it would better pick a suitable grid size which wouldn't influence the efficacy of the computer. Also, the variance of some continuous CSM of some boundaries are more continuous for that it could give the robot a better understanding of the distance to the walls so that the robot could read through the map and having lower chance to bump into the walls.