* List cited sources on frontier centroid management (-All)
  + Ideas on optimization: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6705430&tag=1>
  + <https://my.wpi.edu/bbcswebdav/pid-628611-dt-content-rid-3338687_1/courses/RBE3002-D16-MASTER/Fast%20Frontier%20Detection%20for%20Robot%20Exploration.pdf>
  + Looks like the algorithm is pretty simple: wavefront out from the robot to find any explored cells with unexplored neighbors, and add them to the frontier. The wavefront is needed because we don’t want to count cells enclosed within obstacles as frontier cells. Once the frontier exists, build sub-frontiers using blob detection (simple distance algorithms for example) and compute their centroids. There are two ways a centroid could be unreachable: it could be inside a known obstacle or enclosed in a ring of obstacles, or it could be inside an obstacle or ring of obstacles that hasn’t been searched. The first is easy to detect. In the second case, the mapper will keep trying to reach the centroid until it is proven inaccessible. Hopefully, the nav stack can indicate this condition! If not, we’ll have to create a means for detecting a captured centroid.
  + <http://wiki.ros.org/frontier_exploration> Sends movement commands to move\_base (Optional) Global /map provided by [map\_server](http://wiki.ros.org/map_server), [gmapping](http://wiki.ros.org/gmapping), or the global costmap from [move\_base](http://wiki.ros.org/move_base)
  + <https://www.ikg.uni-hannover.de/fileadmin/ikg/staff/thesis/finished/documents/ma_gangl.pdf> (Important around Chapter 2, page 32 of the book (pg 47 of the .pdf))
* What it means to be finished mapping (-Bill)
  + Update: Joe says the room will be closed off. Thus, completion marks an enclosed perimeter containing the robot. It would be ideal to check for completion every time the map updates, but this will probably be too computationally intensive. Really, we only need to check if there is a possibility that completion has occurred. If the robot moves, then the time during the movement and for some period after the movement are the windows for completion to occur. So, a reasonable approach would be to periodically check for completion at all times, and check more frequently when the robot is moving.
  + So keep going until you’ve found you’re trapped in a box? Go another time around to check the walls and make extra sure it’s not just a dead end? (-Annie)
* Requirements for the GUI (-Trevor)
  + Obstacles
  + Mapped area
  + Unmapped area
  + Robot path
  + Area in viewing range, but unseeable due to obstacles
  + The robot
  + <http://www.aaai.org/Papers/AAAI/2000/AAAI00-131.pdf>
* What nodes we need and what do they subscribe to, basic architecture (-Annie)
  + Package gmapping, calls node slam\_gmapping which creates occupancy grid
  + Slam\_gmapping subscribes to tf/tfMessage and sensor\_msgs/LaserScan
  + Publishes to nav\_msgs/MapMetaData, nav\_msgs.OccupancyGrid, and std\_msgs/Float64
  + Uses service nav\_msgs/GetMap to get map data
  + <http://wiki.ros.org/gmapping>