Region Editor and Calibration User Guide

Contents

Overview 1
Real-world representations 2
Grid 2
Markers 2
Background image 2
Creating the map 3
Creating regions 3
Editing regions 3
Boundary 3
Creating the physical map from pre-drawn map 4
Calibration 5
Defining real-world representation 5
Correlating the map to the real-world 5
Creating a custom marker stream 6

Overview

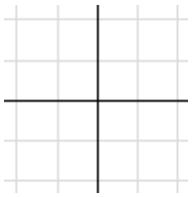
Region Editor is used to build a two-dimensional map which defines the possible positional state space for a given experiment. This new version of Region Editor (developed in Spring 2012) provides much more functionality than the old version, particularly allowing the creation of maps that will not require calibration. Maps are built using “real-world units” (generally meters), so Region Editor is used most efficiently when this is kept in mind.

The Calibration GUI is designed as a “backup,” primarily to be used if the map is built without first correlating it to the real-world coordinates, or if a similar map is desired to be used at different overall sizes. Calibration is simply done by linking points on the map to points in the real world, after which the calibration will transform all map points to correspond correctly.

Real-world representations

The primary advantage of the new Region Editor is its creation of a map using real-world coordinates. These are usually interpreted as meters (for the purpose of consistent marker mapping), but they can be any unit of length. There are three primary ways of ensuring that the map points are placed in their correct real-world locations.

Grid



The simplest method for ensuring correct placement of region points is by putting them in approximately correct positions based on the shown grid. The thick black lines are the x and y axes, and the thin gray lines are placed at single-unit intervals.

Markers

Region Editor and the Calibration GUI have the ability to receive data via UDP port, and plot the corresponding markers on the map. The Autonomous Systems Lab that developed LTLMoP has a setup of Vicon cameras that can determine the position of infrared-reflective balls within its field. That position data is captured in a separate program (ViconMarkerBroadcaster), and then sent to LTLMoP over UDP. Other labs or alternative applications may wish to capture and send different data. See the "Creating a custom marker stream" section for more details on the required interface.

To use the Vicon markers, the ViconMarkerBroadcaster must first build an executable. See readme.txt in that directory for details. This executable must be running prior to clicking the Markers button or menu item that will receive and plot the markers on the map. It is suggested that the marker listener be toggled off after starting unless it is expected and important that the markers be able to move. This will decrease necessary processing and avoid small amounts of flashing going on due to continuous redraws.

Background image

Importing a background image is most useful when trying to build a map for an area outside of the field from which markers can stream. The background image is frequently a floor plan of some sort, or a screenshot of Google Maps or similar for outdoor maps. The image should be scaled to the correct real-world size with the use of the "scale image" functionality. Simply click the scale image button and then two points on the image, of which the real-world distance between is known. This is commonly a scale with a floor plan or outdoor map, but can be any two points on the image. The origin can be "set" by selecting the background image and dragging it until the axes line up at the desired point.

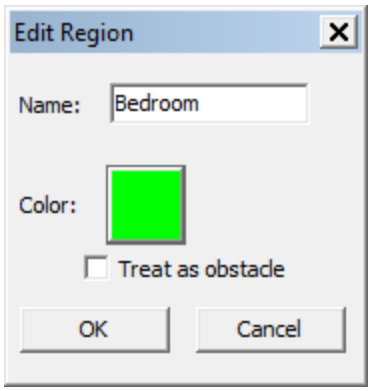
Creating the map

Creating regions

Regions are created in either rectangular or other polygonal shapes, although the representation is the same. Clicking the appropriate button or menu item, or using the keyboard shortcut (“R” or “P”), will cause entry into region creation mode. Rectangles need only to have two of their corners defined by mouse-clicks, while polygons require all points to define.

Right-clicking or hitting “Esc” during region creation will delete the partially created region and exit region creation mode. Hitting “Backspace” or “Delete” will remove the previously defined point. Left button double-clicking will add a point and then complete the region with an edge between the new and the first point, so long as there will be at least three points.

Editing regions



Selecting a region allows for the region vertices’ positions to be changed. A single point, the entire region, or even multiple regions can have their positions changed. Points can be added or removed from regions using the menu Drawing -> Create Point and Drawing -> Delete Point, or shortcut keys “C” and “D” respectively. Points must be created near the boundary of a region, and regions must contain at least three vertices (so no points from a triangle can be deleted).

Double-clicking on a region brings up an Edit Regions dialog, that allows for the changing of region name, color, and/or obstacle indicator. If a region is marked as an obstacle it will show up as darker, and LTLMoP should avoid allowing the robot to move through that region.

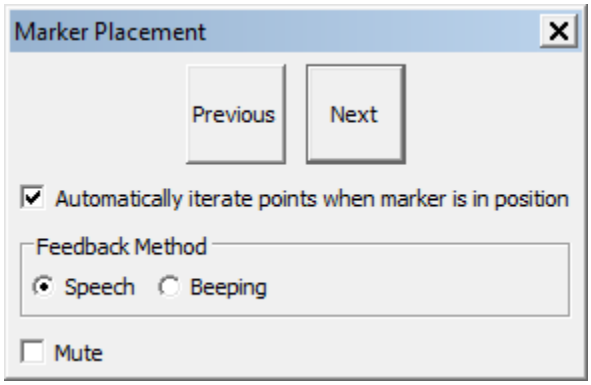
Region points can also be moved by defining the distance between two points. Clicking the “Set length” button (same button as for scaling an image), then clicking two region points will pop up a dialog box. This box will display the current distance between the points and allow a user to change that distance. The first point clicked will remain where it is and the second point will move. Note that this is not true “dimensioning” since the distances will not be saved after the point is moved. This is not a CAD program, so no constraints on the regions are recorded.

Boundary

LTLMoP requires that one of the regions in the map be a boundary region that surrounds all other regions. This defines the limit of positions that the robot is allowed to move through. A boundary region can be created by changing the name of an existing region to “boundary.” Alternatively, the Autoboundary button or menu can be used to automatically draw the smallest rectangle that will contain all existing regions. Autoboundary can only be used when at least one region is already defined. If the map is saved prior to creating a boundary, a dialog box will pop up asking if the Autoboundary functionality should be employed.

Creating the physical map from pre-drawn map

Though it is much easier and recommended to first create the physical map, and then draw the LTLMoP representation to correspond to it, there is a tool to help map creation go the other way. This is also useful when adding regions to an existing map, if uncertain of where they are in the physical world. The Audio Feedback functionality, started via button or menu item, will begin streaming markers and providing audio commands for building the map. Obviously the program that captures marker data and sends it to Region Editor must be started first.



The Audio Feedback functionality picks a point in the regions and checks if there is a marker in that position. If not, it looks for the closest moving marker within a certain range of effect. If it finds a moving marker in that range, it will issue audio commands to assist in moving the marker to the correct region point. Once the marker is settled in the correct position, the audio will indicate this. The region point and the moving marker will be indicated on the map by large red circles.

The “Previous” and “Next” buttons allow the user to cycle through the region points to choose the point on which to give feedback. Checking the “iterate points” checkbox will have the feedback functionality cycle the points when a marker is settled in the correct position. This is useful when setting up the map while far away from the computer. The “Mute” checkbox will allow for stopping all sounds and just displaying the red dots, which is useful when near the computer, but not right next to it.

The speech commands given are in the form of “X negative one point three, Y zero point two.” This seemed to be a good compromise between speed and clarity. It indicates how the moving marker should be moved in each of the Cartesian directions to bring it to the region point. Because of the delay in speaking, it seemed to work the best to immediately move the marker in accordance with the commands as they are relayed, then pause and wait for the feedback to read the new marker position and give the next command. If the marker passes over the goal point without settling, the speech will say “Stop,” and if the marker is settled in the correct position, the speech will say “Success. Next.”

If the speech commands are undesirable or too slow, there is the option of using beeping of different pitch to indicate a kind of “hot or cold” closeness to the target position. The pitch increases the closer the moving marker is to the goal point. Two quick beeps of the same pitch indicate passing over the goal position without stopping, while three quick beeps of increasing pitch indicate successful placement of the marker and moving on to the next position (if checked).

Be warned that the audio feedback can be annoying to others in the area, and the beeping audio is far more annoying than the speech in addition to being less efficient.

Calibration

Calibration can be accessed either through the Region Editor button or menu item, or through the Specification Editor Configuration (specEditor -> Run -> Configure Simulation -> Configure Robot -> Pose Handler Configure -> Run Calibration Tool). When accessed through Region Editor, it is used to change the map to match the real-world. When accessed through the Specification Editor, it is used to define a transformation matrix that will convert between map-coordinates and real-world coordinates.

Defining real-world representation

The Calibration GUI contains two windows, the top of which is the map as created in Region Editor, and the bottom of which is the "real world." The same three real-world representations that were available in Region Editor are available here: the grid, markers, and a background image. As before, the background image will need to be scaled and positioned in the same way. Note that the scaling tool cannot be used on the map representation, and indeed the regions cannot be edited in any way.

This should be done prior to defining the calibration points, as the calibration points will not change with a scaling of the image

Correlating the map to the real-world

The map transformation is defined by creating calibration points that show equivalence between points on the map and real-world coordinates. A lowest-cost (similar to least-squares) transformation matrix is created from these points and is used to transform the map coordinates. There must be at least three points defined to avoid under-defining the transformation matrix.

The points are created by clicking the "Add calibration point" button, then clicking a point in the map panel and its corresponding point in the real-world panel. Undesired points can be removed by using the "Remove calibration point" button.

Hitting the "Save" button or menu item will calculate the transformation matrix and close the Calibration GUI. If Calibration is called from Region Editor, the region points will be remapped. If Calibration is called from the experiment configuration, the transformation matrix will be saved. If the "Aspect ratio" checkbox is marked, then the calculated transformation matrix will not allow shearing. If the aspect ratio is maintained, then the map will only scale as a whole and rotate as appropriate. If unmarked, the dimensions are free to scale independently. However, this can lead to odder looking maps at times.

Creating a custom marker stream

The marker listener implementation receives data through UDP communication on a specific port and plots the marker positions accordingly. Currently it listens to the local computer (IP addresss 0.0.0.0) on port 7500. It listens at an approximate frequency of 20 Hz. These values can be changed in src\lib\vicon.py.

Currently, the ViconMarkerBroadcaster C# program is set up to send that data to the listener. However, a different program can easily be substituted, in any programming language with the capability to send UDP. This program must send its data to the listener on a consistent IP address and port (usually easiest if the program is just run on the same computer). The serial data must be an array of bytes representing all the markers at a given moment in time, to be sent all at once. There should be 16 bytes per marker, the first 8 for the marker x-coordinate, the last 8 for the marker y-coordinate (typical representation of doubles). There should be no header or checksum bytes, as the incorrectness or inconsistency of a single message has little effect on the operation of Region Editor or Calibration.

The ViconMarkerListener.py that comes in the ViconMarkerBroadcaster project folder is similar to, but not the same as vicon.py. This serves no functionality in the current implementation of LTLMoP, but is very useful for receiving the current marker positions from ViconMarkerBroadcaster for miscellaneous purposes, though it does no movement tracking on the markers.