

Marker SLAM User Manual

For Windows, Mac, and Linux

Purpose

Determine the position and orientation of a webcam in real time using fiducial marker SLAM (simultaneous localization and mapping).

See the location information on any device using OSC (open sound control).

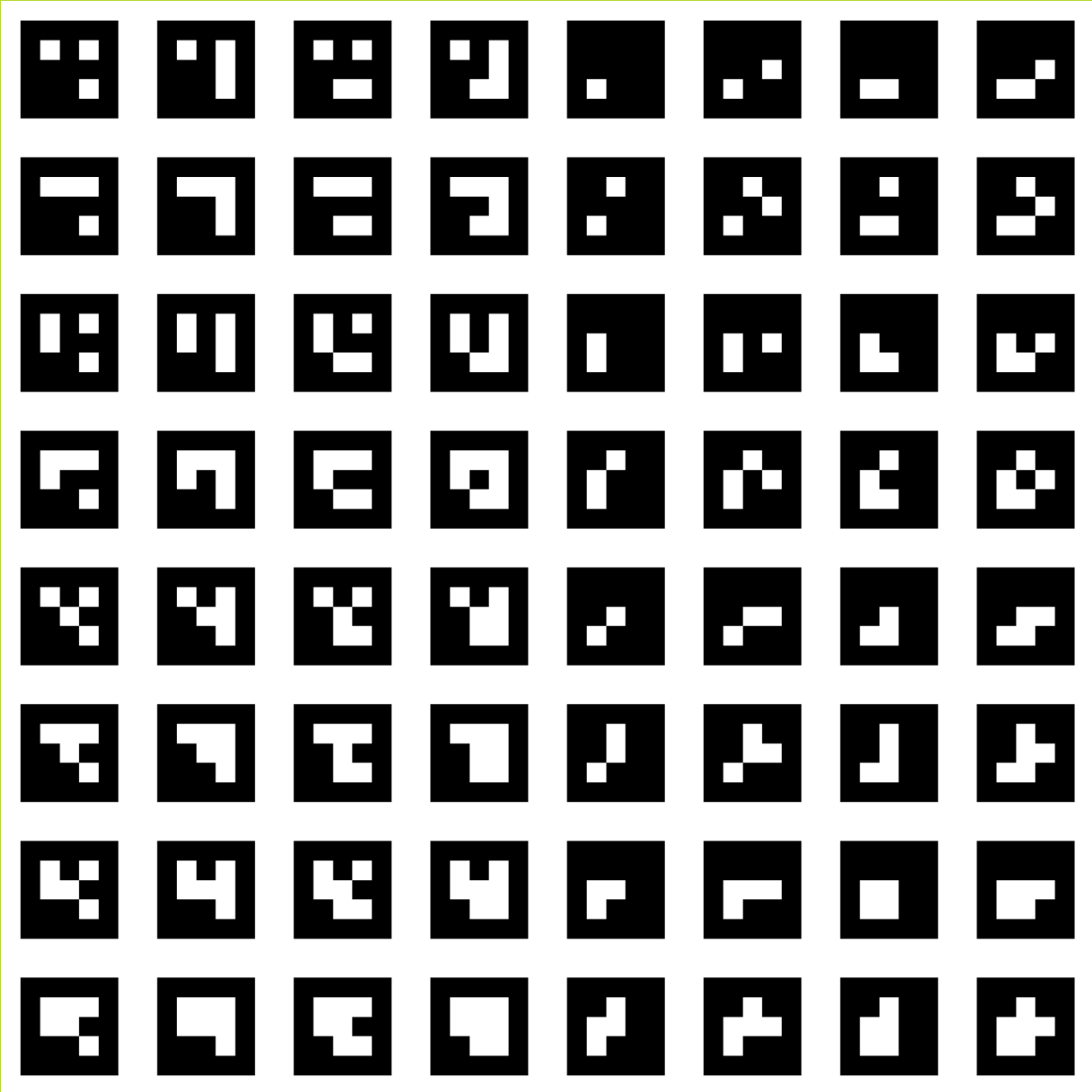
Getting Started

You will need:

- A room
- Printed fiducial marker/s (see list of acceptable markers on next page)
- Webcam/s

Step 1: Markers

Acceptable markers are shown below.
Print out up to 64 different markers on individual pages.
Print out each marker in any size.



Step 2: Webcams

Connect all webcams to your Windows, Mac, or Linux device that will be running the application.

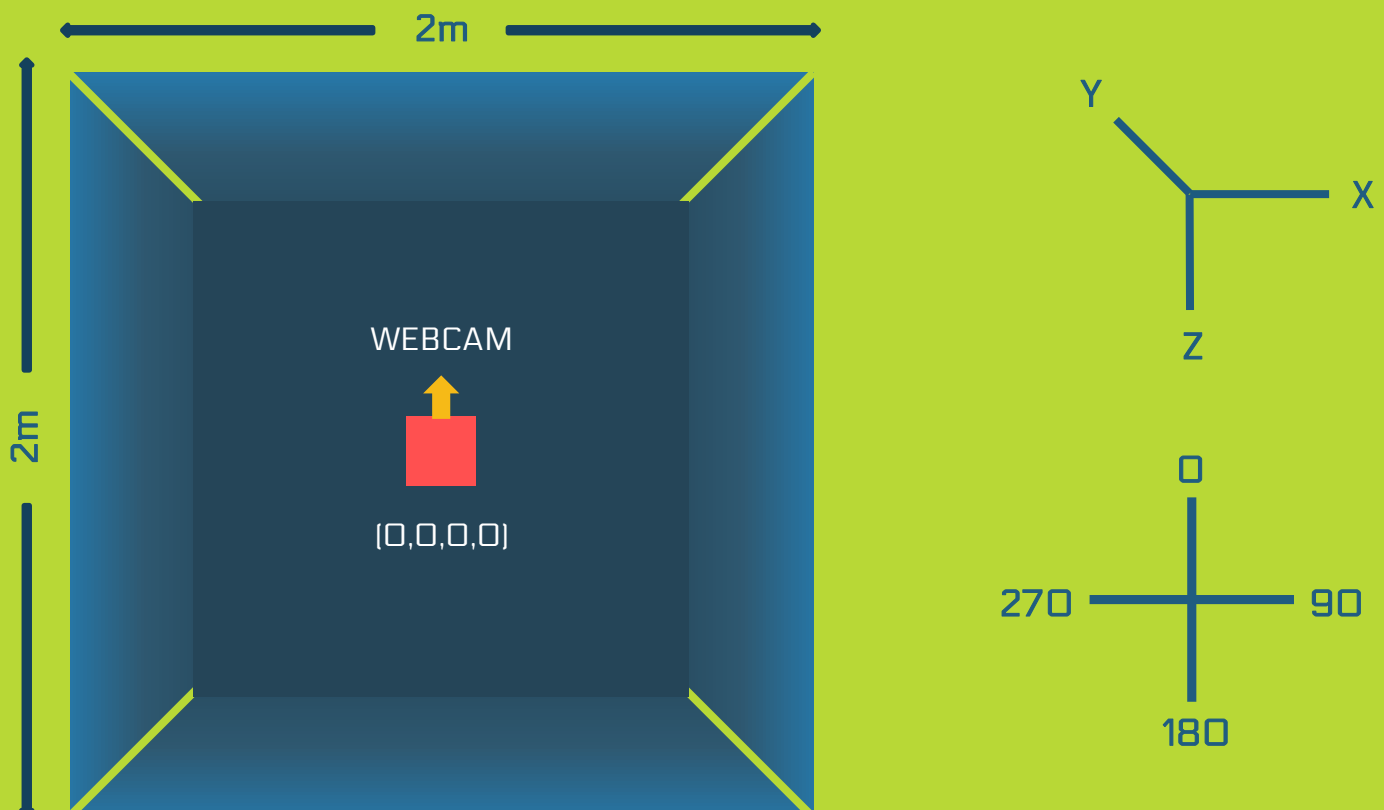
Tip: Try adjusting your webcam's contrast/sharpness so that the image appears sharper. This will allow the program to work better.

Step 3: Find Your Starting Point

Pick any spot in the room to be point $(0, 0, 0)$.
Pick any direction to be the 0° orientation.

Example:

I will choose $(0, 0, 0)$ to be in the center of the room, and 0° will be in the North direction. (See picture below)



Step 4: Place Your Markers

Choose where you would like to place the markers relative to the origin point.

Determine the orientation of the marker first. Then determine the x, y, z position when you are in that orientation. Make sure to measure to the center of the marker.

Example:

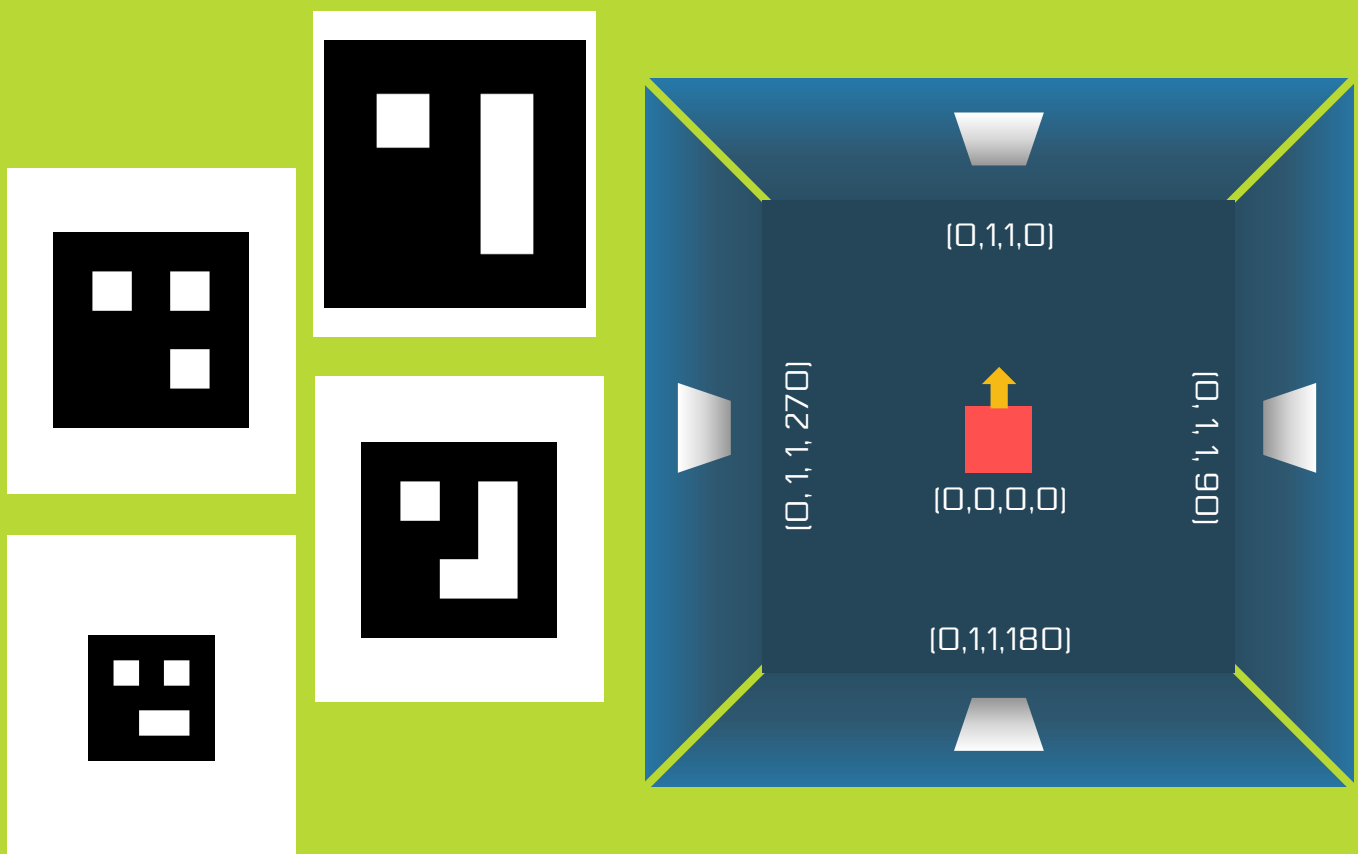
I printed out 4 markers. (See picture below).

Marker 1: Orientation: 0, Position: (0, 1, 1)

Marker 2: Orientation: 90, Position: (0, 1, 1)

Marker 3: Orientation: 180, Position: (0, 1, 1)

Marker 4: Orientation: 270, Position: (0, 1, 1)



Step 5: Basic App Setup

Select a webcam and whether it is front or back-facing.
If desired, view the help screen.

The screenshot shows the app's main interface. At the top, there are three callouts: 'See the Help Screen' pointing to the 'Help' button, 'Select a Webcam' pointing to the 'TOSHIBA Web Camera - HD' dropdown, and 'Show/Hide List of Markers' pointing to the 'Registered Markers' dropdown. The main view is a camera feed showing a red square marker on a white paper, labeled 'Marker #1 [0,1,1,0]'. A blue callout 'List of Markers' points to a table on the right side of the screen. At the bottom left, a 'Flip Camera' callout points to a camera icon. At the bottom, a status bar contains several elements: 'Webcam Position x, y, z direction (in meters)' pointing to 'Position: 0.14, 0.97, 0.31'; 'Webcam Orientation y direction (in degrees)' pointing to 'Orientation: 1.42'; '# of Markers /Add More Markers' pointing to 'Count: 4' and a '+' button; 'Save Marker Data' pointing to a save icon; and 'Load' pointing to a folder icon.

ID	Icon	0	1	2	3	4	5
1.		0	1	1	0	0.15	-
2.		0	1	1	90	0.2	-
3.		0	1	1	180	0.05	-
4.		0	1	1	270	0.1	-

Step 6: Register Your Markers

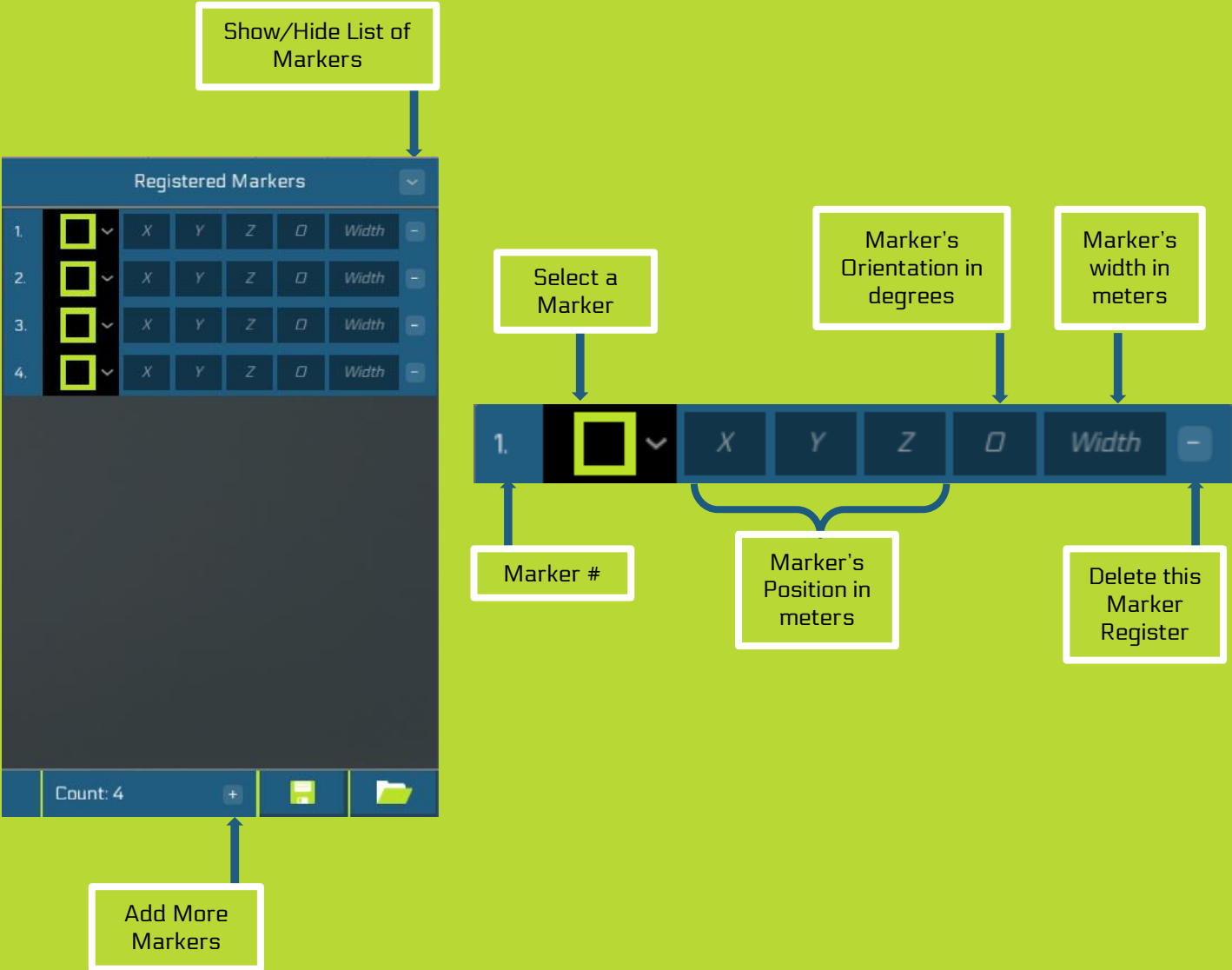
Click on the + sign by *Count* to add a marker to the list of registered markers.

For each marker register, select a marker. Then type in its position and orientation.

Measure the width of the marker. Type it in.

Click on the – sign by a marker register to delete it.

A marker will not be detected unless registered by the user.

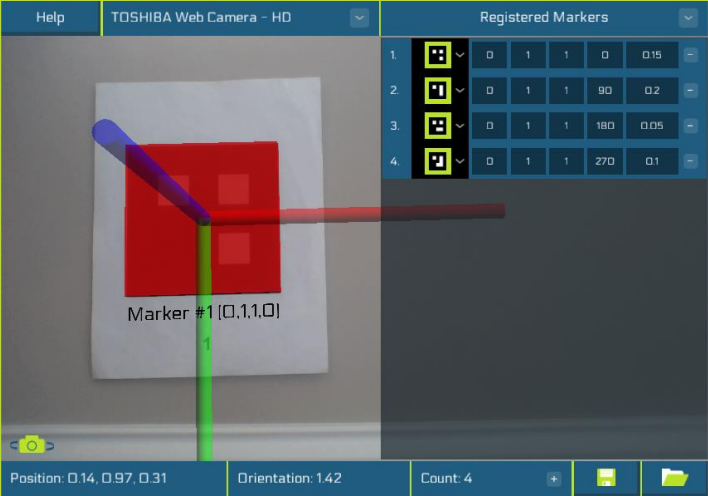


Step 7: Determine the Webcam Position/Orientation

Once a marker has been registered, it can be detected by the program.

A set of axes and a red cube will be overlaid on the marker if it has been detected. The marker's registered number, position, and orientation will also be displayed on the marker.

In the bottom left hand corner of the screen, the webcam's position and orientation will be displayed, once the program has detected at least one marker.



Current Location of Webcam



Step 8: See the Webcam Position/Orientation on Any Device

The program sends out a packet of information containing the webcam's position and orientation every few seconds via OSC.

To receive this information on another device, make sure you have Python installed. Then make sure you have the file `reader.py`

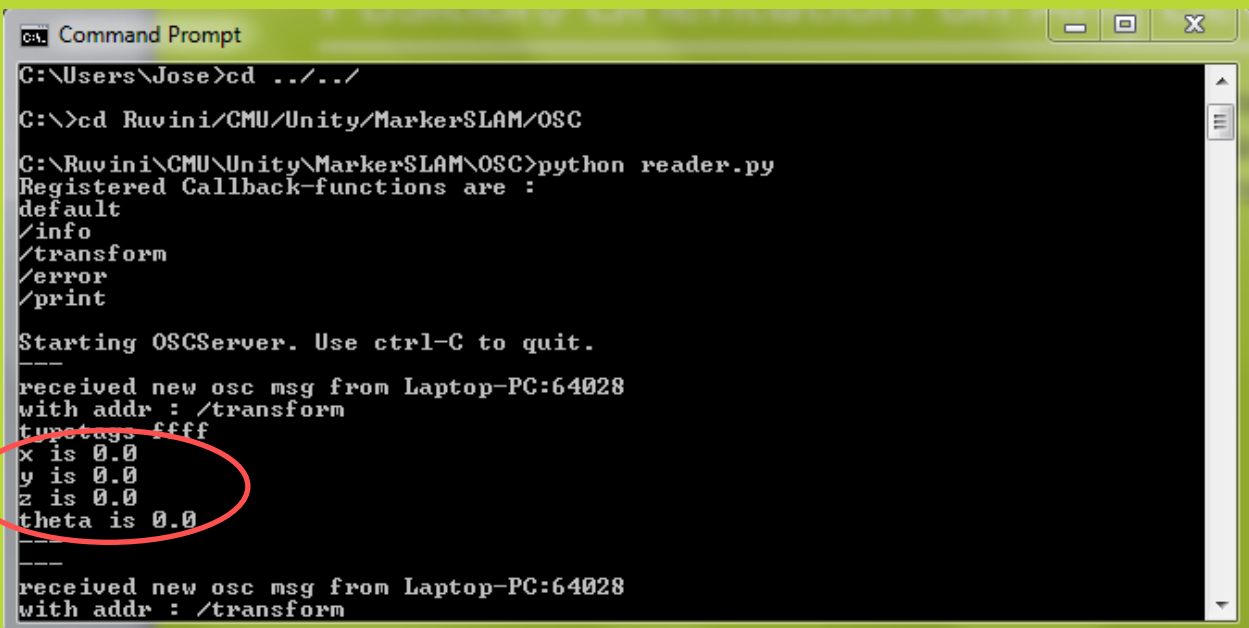
Open Command Prompt, or Terminal.

Go into the folder containing `reader.py`

Type in "`python reader.py`" and click Enter.

If working correctly, Command Prompt should display, "Starting OSCServer..."

Once the Marker SLAM app is running, it will send out the position/orientation information. Command Prompt will then start displaying "received new osc msg..." with the information every few seconds.



```
ca: Command Prompt
C:\Users\Jose>cd ../../
C:\>cd Ruvini\CMU\Unity\MarkerSLAM\OSC
C:\Ruvini\CMU\Unity\MarkerSLAM\OSC>python reader.py
Registered Callback-functions are :
default
/info
/transform
/error
/print

Starting OSCServer. Use ctrl-C to quit.
---
received new osc msg from Laptop-PC:64028
with addr : /transform
type tags: ffff
x is 0.0
y is 0.0
z is 0.0
theta is 0.0
---
received new osc msg from Laptop-PC:64028
with addr : /transform
```

Step 9: Save

Save your all of your registered marker data to use again later.

Click on the floppy disk icon in the bottom right-hand corner.

The program will take a screenshot that will be displayed in the save browser.

Choose a destination to save, type in a filename, then click Save.



Step 10: Load

Load a previously saved set of registered markers.

Click on the open folder icon in the bottom right-hand corner.

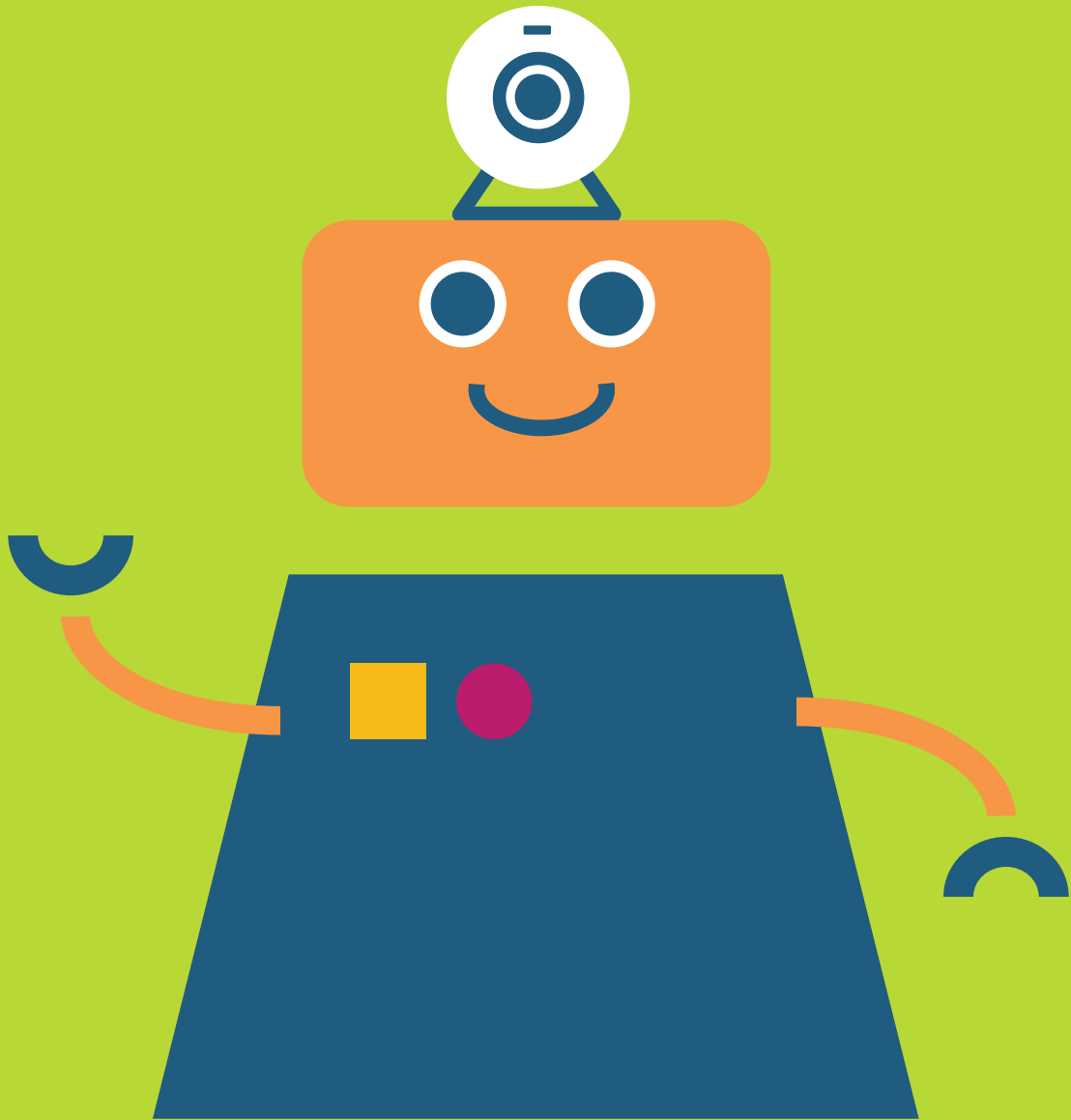
Find the destination where you last saved the marker data.

Click on the file that you want to load.

Confirm this is the correct file by looking at the screenshot that appears.



Thanks for using the app!



License

IMPORTANT: READ BEFORE DOWNLOADING, COPYING, INSTALLING OR USING.

By downloading, copying, installing or using the software you agree to this license.

If you do not agree to this license, do not download, install, copy or use the software.

License Agreement For Open Source Computer Vision Library

Copyright (C) 2000-2008, Intel Corporation, all rights reserved.

Copyright (C) 2008-2011, Willow Garage Inc., all rights reserved.

Third party copyrights are property of their respective owners.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* The name of the copyright holders may not be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed.

In no event shall the Intel Corporation or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.