# Guidance on Position Estimation Setup

PanFeng, 15210720168@fudan.edu.cn

Indoor positioning is an important part in our project. The autonomous flight of drones heavily relies on accurate local frame coordinate. This article is a guidance on how to set up the markers and the camera to get the best position estimation performance.

NOTE: This position estimation method uses an open-source software which we call pixy\_node. The source code can be found here: https://github.com/gaochangyu/pixy\_node

## 1. Setup the markers.

First of all, we need to stick some markers on the wall and measure the distance between these markers.



Fig.1. Marker setup in our experiment.

The four stickers correspond to the four point , , , and . From the left to the right, The three red stickers are point L, R, S. And the green sticker is point M. Figure 1 in the paper[1] that shows the relationship clearly.

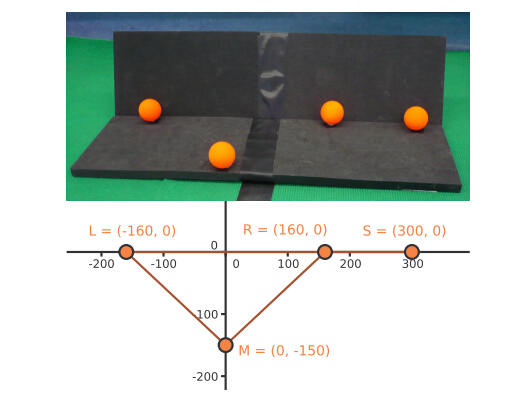


Fig. 2. The marker reference frame mentioned in the paper.

The distance between L and R, M and O (O is the middle point of L and R) needs to be measured. And then change the corresponding place in pixy\_node. The names of the variable are distanceOfLr and distanceOfMo, the unit is cm.

The default value of LR is 50cm,MO is 34cm.You can set the two arguments in the program.

## 2. Set the signature in PixyMon.

Signature is the color of marker. In pixymon, you can set single color or mixed color as the color of the marker. When the program does not output the right position, we can reset the signature. And remember to observe the pixymon program with default mode.

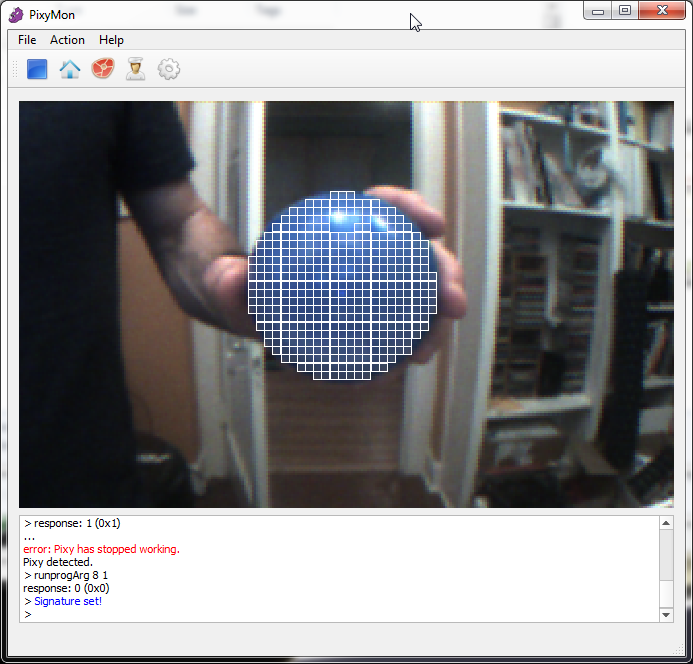


Fig. 3. Set signature in pixymon

Because of the change of external illumination, It’s recommended to manually set the signature before the experiment, which increases the stability of the marker detection.

A small pixel coordinate error leads to great positioning error. So we need to get very accurate pixel coordinate.

## Set the arguments of the position estimation program.

There are two optional arguments for the pixy\_node currently.

|  |  |
| --- | --- |
| -i | Invert the camera image (upside down). |
| -D | Disable the pan tilt function. |
| -h | Show the help. |
| -d | Followed by a integer. Set the distance between lens and the image sensor. It is a relative value,default is 240. |
| -l | Followed by a integer(cm). Set the distance of LR. |
| -m | Followed by a integer(cm). Set the distance of MO. |

## Run the program.

The program relies on ROS. After setting up the ROS environment, you can use rosrun or roslaunch to run the program.

Example:

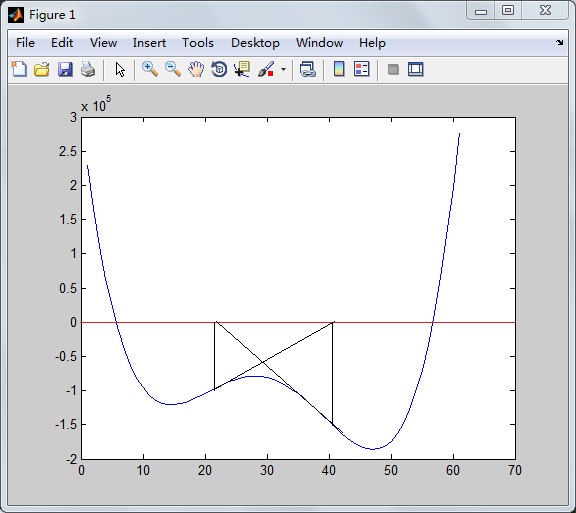
rosrun pixy\_node pixy\_node -i -d 250 -l 50 -m 20

The program has published ros topic Pixy\_Node/pose. You can type rostopic echo /Pixy\_Node/pose to check the output.

## Trouble shooting.

We use Newton’s method to calculate the position of point M. When it comes into such situation like what is showed in the following figure, Newton method will be useless. It will keep running and cannot get closer to the answer we want. That is what we need to avoid. But up to now, we haven’t find a useful method to solve this problem.

p.s. The idea of Newton’s method is as follows: one starts with an initial guess which is reasonably close to the true root, then the function is approximated by its [tangent line](https://en.wikipedia.org/wiki/Tangent_line) (which can be computed using the tools of [calculus](https://en.wikipedia.org/wiki/Calculus)), and one computes the x-intercept of this tangent line (which is easily done with elementary algebra). This x-intercept will typically be a better approximation to the function's root than the original guess, and the method can be [iterated](https://en.wikipedia.org/wiki/Iterative_method).



## Reference:

[1] A. Masselli and A. Zell, “A novel marker based tracking method for position and attitude control of MAVs,” presented at the Proceedings of International Micro …, 2012.