MEAM 620 Project 2 Phase 1

Due: Friday, March 27, 2015

For this phase you will implement a vision based 3-D pose estimator which estimates position and orientation of the quadrotor based on AprilTags[1].

1 Vision Based Pose Estimation

1.1 Environment

The provided data for this phase was collected while a Nano+ quadrotor flew a prescribed trajectory or held by hand over a mat of AprilTags each of which has a unique ID that can be found in `parameters.txt`. Figure 1 shows the layout of the AprilTag mat. The tags are arranged in a 12 x 9 grid. The top left corner of the top left tag should be used as coordinate (0, 0) with the X coordinate going down the mat and the Y coordinate going to the right. Each tag is a 0.152 m square with 0.152 m between tags with the exception of the space between columns 3 and 4, and 6 and 7, which is 0.178 m. Using this information you can compute the location of every corner of every tag in the world frame.

1.2 Calibration

The intrinsic camera calibration matrix and the transformation between the camera and the robot center are given in `parameters.txt`. Two photos (`top_view.jpg` and `side_view.jpg`) are included to visualize the camera-robot transform. You will need to transform your camera-based pose estimation to the robot center, such that you can compare it against the Vicon ground truth.

1.3 Pose Estimation

Each data is provided in one `mat` file. The file contains the image data as a struct array called `data`, which contains all of the data necessary to do pose estimation. It also contains Vicon data taken at 100 Hz which will serve as ground truth. The format of image data struct contains the following data:

1. Time stamp
2. ID of every AprilTag that is observed in the image.
3. The center (p0) and four corners of every AprilTag in the image. The corners are given in the order bottom left (p1), bottom right (p2), top right (p3), top left (p4) expressed in image coordinates, see figure 2. The i^{th} points in each of these fields corresponds to the i^{th} tag in the ID field.
4. Rectified image
5. IMU data (Euler angles, angular velocity, and linear acceleration)

The Vicon data is present as two matrix variables, `time` and `vicon`. The `time` variable contains the timestamp while the `vicon` variable contains the Vicon data in the following format:

$$\begin{bmatrix} x & y & z & \text{roll} & \text{pitch} & \text{yaw} & v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$$

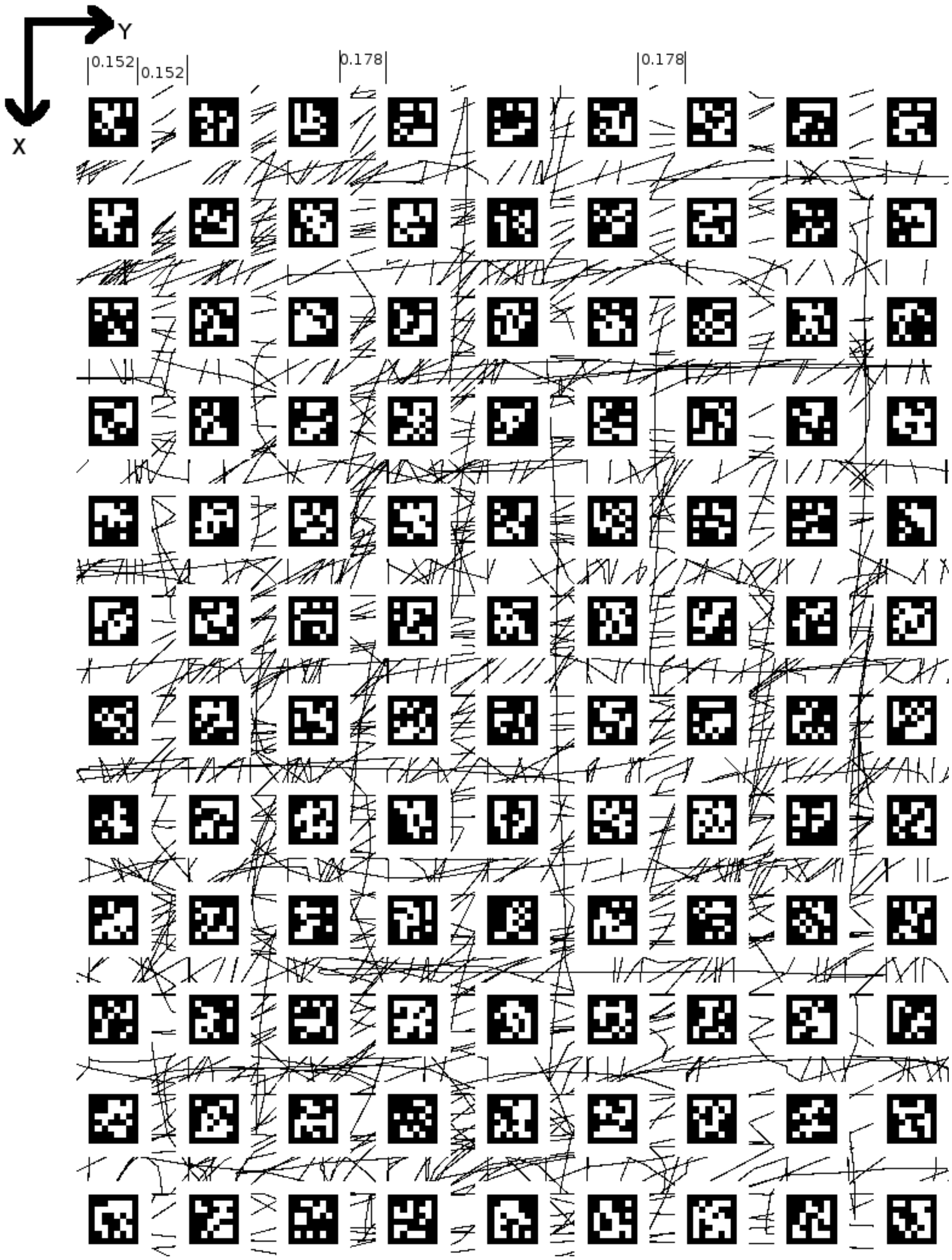

Figure 1: The AprilTag mat

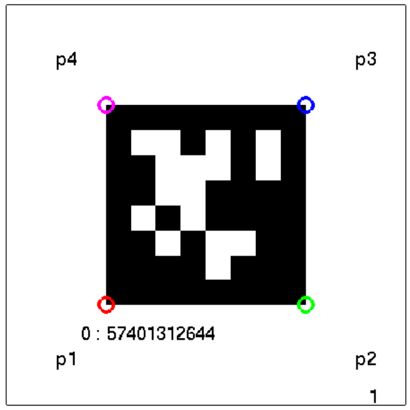

Figure 2: Corners of the AprilTag

Note that for some packets, no tags are observed, you therefore do not need to compute the pose for those packets. The rectified images and IMU data are not necessary for this phase, we keep them for the sake of data format consistency. Using the camera calibration data, the corners of the tags in the frame, and the world frame location of each tag you can compute the pose of the Nano+ for each packet of data.

2 Submission

You will use the `turnin` system for submitting your code for this phase. The project name for this assignment is called `proj2phase1`, so the command for `turnin` would be

```
$ turnin -c meam620 -p proj2phase1 -v *
```

Your submission should include:

- A `README` file containing anything specific we should be aware of.
- Files `init_script.m`, `estimate_pose.m` and any additional files you use for your computations.

Shortly after submitting using `turnin` you should receive an email from `meam620@seas.upenn.edu` stating that your submission has been received. After that, you can check the status of your submission on the monitor webpage at `https://alliance.seas.upenn.edu/~meam620/monitor/`.

Once the automated tests finish for your code, you should receive an email containing your test results. This email will contain plots of your estimates compared to the ground truth and also tell you how long each iteration of your code takes. You should write your own functions for evaluating the error between your estimate using vision and the ground truth. Your grade would depend on the time taken per iteration and the error between your estimate compared to ground truth.

References

[1] AprilTags, `http://april.eecs.umich.edu/wiki/index.php/AprilTags`