

# Title

Pengju Jin, Siddhartha Srinivasa

**Abstract**—Abstract Goes here

## I. INTRODUCTION

Detection and identification using artificial landmarks has long been used in augmented reality and computer vision applications. Over the last decade, there has been numerous marker systems for obtaining more accurate detections and information encodings. Specifically, planar markers such as ARTag, ARToolkit, are among the most popular forms of fiducial markers.

Compared to markerless detection algorithms, these methods are simpler and produce much more reliable detections. There has been significant effort in improving the detection speed and encoding accuracy. They have yield great results for computer vision tasks that require high detection accuracies like camera calibrations, 3D reconstruction. Furthermore, they have gained popularity in the robotic community for having unique characteristics of high detection rates and numerous encoding schemes. For example, ARTags are commonly used to test SLAM systems or finding ground truth for objects in manipulation and motion planning tasks.

Despite all the improvements in this field, obtaining accurate pose estimation from the tags remain a challenge. This is especially important for generalizing the use of fiducial tags in robotic applications. While current detection algorithm yield great results under well conditioned or simulated environments, it is very difficult to obtain reliable poses in noisy settings. For instances, when ARTags are used with low resolution camera or in poor lighting conditions, the system often produce poses with tremendous rotational errors as shown in Figure 2. In fact, we observe that the localization accuracy of the state of the art system perform significantly worse at difficult viewing angles or when there are noise in the scene.

Based on the observation, we present two contributions in this paper. First, we conducted an in-depth analysis on the effect of various noises on pose estimation process. In particular, the noise in RGB images creates a perspective ambiguity problem that makes the pose estimation challenging without additional information. Second, we describe a novel algorithm that takes advantage of the RGBD sensors that are commonly available on most robotic systems to accurately estimate the pose from a single tag under noisy conditions. In the core of the algorithm, we recognize that depth data can produce good approximate initial pose in noisy scene. From the initial estimation, accurate poses can be calculated using constrained optimizations based on known uncertainty models of the sensor. There are few key features to this algorithm:

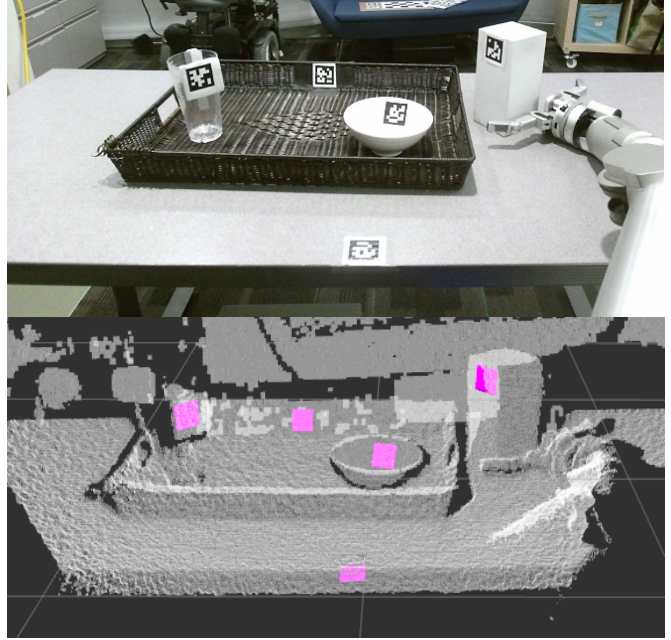


Fig. 1: Apriltag used to localize objects in mainpulatio tasks

- This method is highly robust to noise in the scene. It can obtain accurate poses suitable for robotic applications.
- It is easily generalizable to most fiducial tags designs.
- It performs at worse as good as using only RGB images.

This paper also presents empirical results demonstrating the the successful performance of the algorithm on captured data from the robot. Our implementation of the algorithm is based off of the Apriltag detection pipeline and it is integrated with ROS.

## II. RELATED WORK

Although there are an abundance of fiducial markers in the computer vision and augmented reality field, most of them focus on detection and decoding using RGB images. They are typically designed with a specific binary pattern to encode information in the inner region. There are two large type of common tags, square tags and circle tags.

ARTags[], Aruco[], ARToolkit[], AprilTag[] and AprilTag 2[] are examples of squared based fiducial tags. The perspective projection of a square becomes a general quadrilateral which can be computed easily using any contour tracing algorithm. Given the scale of a single marker, the full 6-DOF pose can then be estimated using the corners of the quadrilateral. AprilTag and AprilTag2 both use a binary payload that guarantees a minimum Hamming distance between tags. Recently AprilTag 2, built upon the origin Apriltag

and ARTag system, also implemented a new boundary segmentation algorithm which gained significant computation speed. Therefore, the advantages of the square tags are that they can be located very efficiently and they have reliable decoding schemes. However, since they are detected using rectangles and lines, the accuracy of their corner point sub-pixel locations are limited. This downfall cause them to have poor pose estimation results under occlusion and image distortion.

On the other hand, circular tags are created to encode the payload using small circles. The example of circular tags include Concentric Circle, Intersense and Rune tags. The perspective transformation of a circle is an ellipse, which can be used to directly compute the pose using back projection methods. The detection of circular features are more accurate to localize and thus generates better pose estimations. However, small circular features become hard to detect when they are far away from the camera and thus their effective range is smaller than that of square tags. Furthermore, Rune tags and Pi tags have significantly higher computation cost to decode their payload.

### III. CHALLENGES

In most square fiducial tag detection, the pose is calculated by fitting a quad around the planar tag. The corners are extracted from the tag and the pose is estimated using 3D to 2D point correspondence optimization methods such as Levenberg–Marquardt [OpenCV reference] or DLT. This is a specific case of the perspective-n-point problem and it has been well studied in geometry based Computer Vision literatures [1]. In particular, there is a deterministic solution to the perspective-4-point problem. In other words, given the projection of a tag corners, the pose of the tag is unique. In reality, however, when a small tag is captured in a low resolution camera, the perspective projection becomes almost orthographic. In these cases, a small variance in the corner detection process will yield estimations far from the true pose due to a perceptual ambiguity under projection.

We will illustrate this effect by using two overlapping cubes in figure 3. The overlapping face of the two cubes are interlaced but rotated by 120 degrees. However, due to perspective projection the squares appears to be on the same plane. Under low camera resolution, the overlapping squares become virtually indistinguishable. The red circular regions are the detected corners under some sensory noise. In these cases, the optimal PnP solution is no longer singular but a bimodal distribution depending on the viewing angle. The result of the 3D to 2D correspondence optimization might converge to either one of the local minimum.

### IV. APPROACH

This section describes a method for estimating poses of noisy square fiducial tags using RGBD data. Unlike previous methods, the detected corners of the tag are treated as approximated locations of the true corners. Using the corners, the method implicitly evaluate the depth data and RGB

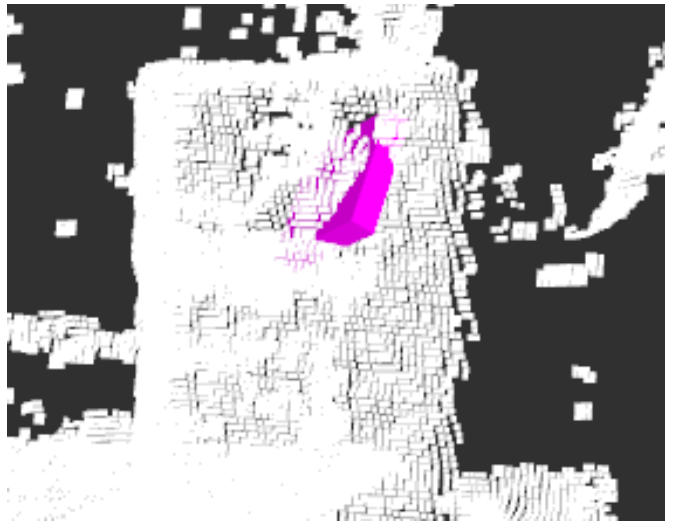


Fig. 2: The orientation of Apriltag placed on the object is greatly misaligned with the actual object

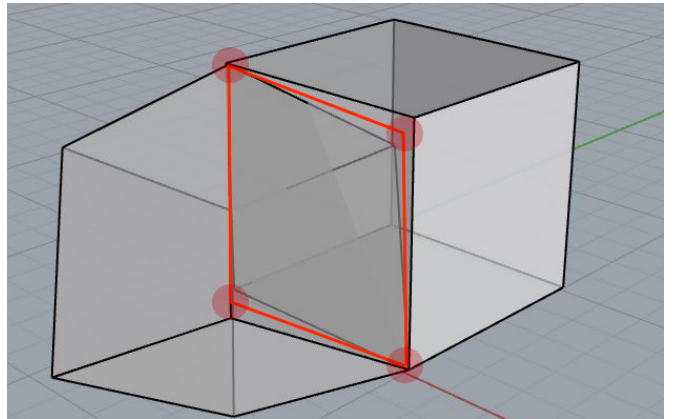


Fig. 3: Perspective Ambiguity illustrated with overlapping cubes

data as two independent observations and fuse them in a meaningful way.

There are three distinct components to this method. First, we find the plane in  $SO(3)$  containing the tag using depth data. Secondly, an approximate pose is computed using the depth plane and the corners from tag detection. Finally, the method refines the initial pose using the RGB data. Each component is described in detail in the following subsections and illustrated in Fig.

#### A. Depth Plane Fitting

With a calibrated RGBD sensor, depth and RGB streams are registered to the same frame. The square patch of points in the depth image defined by the  $g$  also contains the range information of the tag. Here we take advantage of the planar characteristic of the tag. By fitting a plane over the range data, we can constrain the pose of the tag to be on the plane.

The raw range data retrieved from the Kinect One sensor are usually noisy as illustrated by Figure 4. Specifically, borders of the tag and some of the dark regions of the tag produce highly unreliable range data. Therefore, we first filter the data by removing points too far from the median before fitting the plane. Nevertheless, the remaining points could

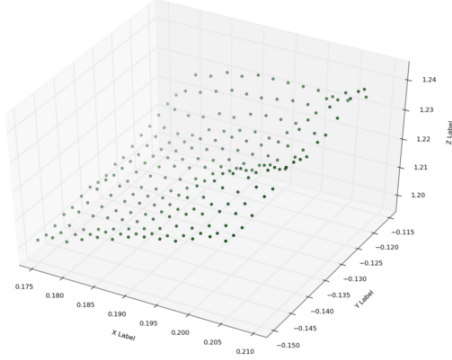


Fig. 4: The depth points of the Apriltag plane. The points are noisy but good for an approximate estimation.

have a large variance depending on the lighting condition and the magnitude of the plane rotation. The accuracy of the plane and tag pose is directly affected by the noise level of data. In fact, we want to characterize the uncertainty and weight the initial estimation accordingly during the fusing stage.

In implementation, we used a Bayesian plane fitting algorithm described in [Uncertainty Analysis] which computes the Hessian Normal parameters of a plane through optimizing

$$\min_{\hat{n}, d} \sum_{j=1}^N \frac{(p_j(\hat{n} \cdot \hat{m}_j) - d)^2}{(\hat{n} \cdot \hat{m}_j)^2 \sigma^2\{\bar{p}_j\}} \quad (1)$$

The algorithm in the paper assumes a radial Gaussian noise in the range data  $p_j$  with the standard deviation modeled by a function in the form

$$\sigma\{\bar{p}_j\} = \frac{kd^2}{\|\hat{n} \cdot \hat{m}_j\|} \quad (2)$$

where  $\hat{n}$  is the local normal to the planar surface of the depth point and  $\hat{m}_j$  is the measurement direction for the sensor for point  $p_j$ . The coefficient  $k > 0$  is an estimated value obtained from sensor calibration. In our implementation, we obtained  $k$  by simplified the results from [Kinect Noise Model paper].

Another important result we used from [] is the covariance matrix for the plane-parameters. The covariance matrix is obtained by taking the *Moore-Penrose generalized inverse* of Hessian matrix computed from the Lagrangian. The covariance characterizes the uncertainty of the plane and defines the refinement regions of the pose later in the pipeline.

### B. Initial Pose Estimation

The pose of the tag can be described as the transformation from tag's frame to the sensory frame of the robot given by  $[R, t]$ . However, the depth plane  $D[\hat{n}, d]$  alone is insufficient to determine the transformation because it only defines 3 DOF in a 6 DOF space. Furthermore, the center of the tag must align with the center of the plane and thus further constrain 2 DOF. However, there are still infinite number of valid poses rotating about the normal  $\hat{n}$ . One solution is to constrain the pose by using a corner as an extra point

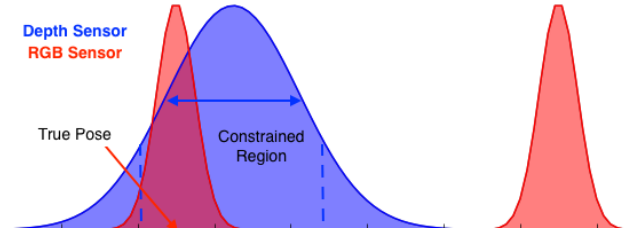


Fig. 5: An abstract visualization of the optimization constraints. The blue curve is the initial pose estimation obtained from the depth plane. The red curves are the ambiguous poses from the RGB image. We constrained the region of optimization based on how well we fit the depth plane.

correspondence to solve for the optimal rotation. However, the accuracy of this method largely depends on the depth accuracy of the chosen corner point.

An alternative is to use all 4 detected corners as 4 pairs of point correspondences for the optimization. We projected the detected corners onto  $D[\hat{n}, d]$  to get the coordinates  $p = [p_1, p_2, p_3, p_4]$  in the robot sensory frame. The corner coordinates  $q = [q_1, q_2, q_3, q_4]$  in the tag frame can be easily calculated since the tag is a square plane. We define the center of the tag as the origin, the corner coordinates are simply the location of the corners on a Cartesian plane. Given two sets of 3D point correspondences, this can be computed as a rigid body transformation estimation [Reference to SVD 3D rigid body]. Solving for the optimal transformation  $[R, t]$  requires minimizing a least squares error objective function given by:

$$[R, t] = \underset{R \in SO(3), t \in \mathbb{R}^3}{\operatorname{argmin}} \sum_{i=1}^n w_i \|R p_i + t - q_i\|^2 \quad (3)$$

There are numerous approaches to solve Eq. 3. Since we have very few correspondences and they are assumed to be correct, it can be computed quickly using SVD:

$$\bar{p} = \frac{1}{N} \sum_{i=1}^N p_i \quad p_{ci} = p_i - \bar{p} \quad (4)$$

$$\bar{q} = \frac{1}{N} \sum_{i=1}^N q_i \quad q_{ci} = q_i - \bar{q} \quad (5)$$

$$p_c^\top q_c = U \Sigma V^\top \quad (6)$$

$$R = V U^\top \quad (7)$$

$$t = \bar{q} - R \bar{p} \quad (8)$$

$R$  and  $t$  are the corresponding rotation and translation components of the the transformation. The above approach minimizes the least square error of the transformation and it is robust to small errors in the correspondences. The resulting pose obtained from the range data, although not accurate, is robust to noise and provide a good approximation for the true pose.

### C. Pose Refinement

Given the initial pose estimate of the tag and the RGB image, we can refine the pose that minimizes the reprojection error. Given camera model  $K$ , initial rotation and translation

estimates  $R_{init}$  and  $t_{init}$

$$\begin{aligned} \hat{y} &= K[(R_{init} + \Delta R)x + (t_{init} + \Delta t)] \\ \min_{\Delta R, \Delta t} & (y - \hat{y}) \\ \text{subject to:} & \\ |\Delta R| &\leq \Gamma_R, \quad |\Delta t| \leq \Gamma_t \end{aligned}$$

The challenge here is to determine the region  $\Gamma_R$  and  $\Gamma_t$  which the initial pose estimate can be adjusted. If we unbound  $\Delta R$  and  $\Delta t$  in the Apriltag detection algorithm on Apriltag detection algorithm the optimization, the solution might converge to a pose optimal for the reprojection error but far away from the true pose due to perceptual ambiguity. On the other hand, if we constrain the optimization too much, the final pose might not be far away from the true pose because of the inaccurate initial estimation. We recognize that the bound of on this optimization is related to the variance of the initial pose estimation. In one extreme, if there is no uncertainty in the depth camera and the range data are perfect, we don't need to further refine the pose of the tag. Similarly, if we don't have any depth information (uncertainty of the initial estimate is infinity), then the best we can do is find the pose solely based on the reprojection error which is the same as solving the unbounded optimization problem. Therefore, this becomes a constrained optimization problem where the bound on the independent variables of  $\Delta R$  and  $\Delta t$  is proportional to the covariance of our estimated depth plane parameters. In our implementation, we used the trust-region algorithm to bound the optimization. The scaling threshold parameter is empirically tested to yield the best results for our robot.

The key insight to our method is that it harnesses the different strength of RGB and depth information during the pose optimization process. By considering all the points on the tag, the depth data is more robust to noises in the scene. However, range data is inherently inconsistent and the pose is thus imprecise. In contrast, the advantage of RGB image is that the corners can be detected at a sub-pixel precision. It is useful for refinement optimization.

## V. EXPERIMENTAL RESULTS

The key problem we are trying to resolve is the localization accuracy of Apriltags in noisy situations. There are two major components we want to demonstrate in this paper: first, we want to characterize the effect of perceptual ambiguity and noise on the Apriltag detection algorithms. Second, we want to test the resilience of our algorithm and show that it can obtain reasonable pose estimations under high level of noise. Finally, we briefly tested the runtime of the algorithm to show that it remains capable of real time detection.

In our experiments, we measured the rotational and translational accuracy of the detections algorithms with respect to three different independent variables: viewing angles, distances, and lighting conditions. We placed a standard camera calibration chessboard and an Apriltag of known size on a solid planar board. The Apriltag has a fixed distance

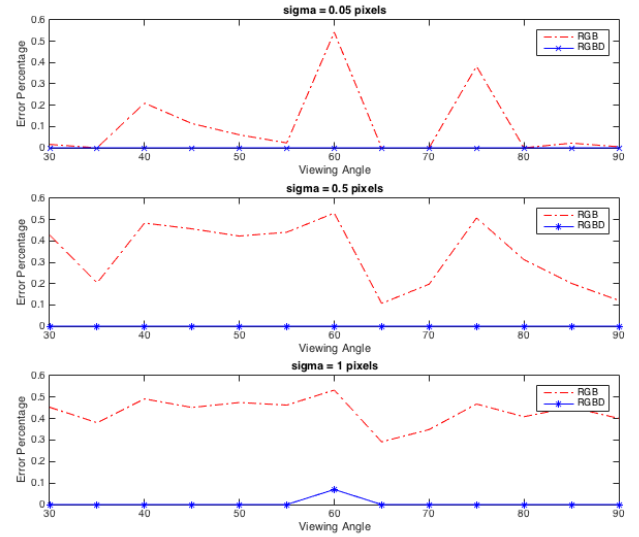


Fig. 6: Viewing Angle vs Error Percentage under different simulated noise level. The new RGBD based algorithm can resist noise in the RGB image and it vastly outperforms the original algorithm.

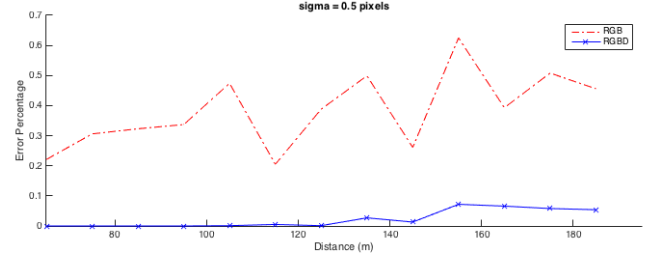


Fig. 7: Distance vs Error Percentage. Data are captured at a 10 cm increment from 65 cm to 185 cm.

from the chessboard. This is used to compute the ground-truth pose for the tag. By using a large chessboard, we can detect the corners to a sub-pixel accuracy and compute accurate ground-truth poses unsusceptible to lighting and sensory noises. Figure 9 demonstrates the experimental set up.

### A. Viewing Angle

The low localization accuracy caused by the perceptual ambiguity of the Apriltags is a non-linear function on the viewing angle of the tag. To characterize the effect, we placed the testing board on a table straight in front of the robot in a well lit room. Since the sensor is taller than the plane of the table, the robot has to slightly gaze down at it. We rotated the testing board at a increment of 5 degrees from 0 degrees to 70 degrees. This is about the range in which the tag can be detected reliably given the camera resolution and distance. At each angle, we captured the RGB image, depth image, and detection outputs from the Apriltag algorithm.

For each captured data collection, we introduced 3 levels of Gaussian noise of  $\sigma = 0.2$ ,  $\sigma = 0.5$ ,  $\sigma = 1$  to the RGB image and computed the resulting tag pose. This is repeated for 1000 trails at each noise level and the errors are computed for each trial. Figure 6 shows some of the results.

As the result shown in figure 7 indicates, the viewing angle



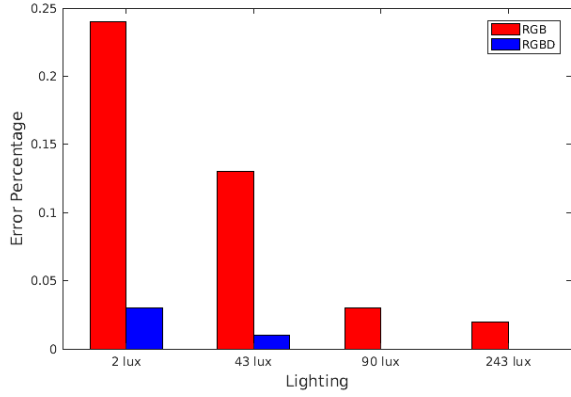


Fig. 8: Illumination vs Error Percentage. Data are captured at 65 cm away from the camera at a 40 degree angle.

has large effect on the rotation error. As we expected, the empirical results show a very clear bimodal distribution for the Apriltags at different viewing angles. The depth-sensor fused algorithm vastly outperforms the previous algorithm as it is not affected by the perceptual ambiguities. The small amount of the noise introduced to the data only cause a small rotational change around the true pose of the tag. In Figure[8], we threshold all the poses based on their rotational errors and plotted the percentage of unacceptable poses at each viewing point. One interesting observation from the data is that, at most viewing angles, the magnitude of noise above a certain threshold has little effect on the localization accuracy. At most viewing angles, relatively small noises causes a significant accuracy decrease.

### B. Distance

In addition to the viewing angle, we captured the images at different distances away from the camera at a fixed angle. Figure 7 shows the experiment results.

The relationship between the distance and localization accuracy is much more apparent. As the tag moves further away from the sensor, the number of pixels on the tag decreases. The perspective ambiguity effect becomes more apparent when there is only a small patch of pixels on the tag. However during the experiment, it is difficult to keep the viewing angle precisely consistent at each trail. Therefore, the pose error percentage using RGB is not increasing smoothly in Figure 7.

We see a clear increase in error percentage in the proposed method when the tag is far away from the camera. This is contributed both by a smaller tag patch size in the depth image and an increasing in noise with the Kinect sensor at a further distance. In these cases, the variance of the depth plane estimation became very wide and the algorithm is unable to converge to the correct pose. The Nevertheless, our method has a significant gain in accuracy at every distance.

### C. Lighting

From our past observations, poor lighting condition is the most significant contributing factor to sensory noise and it results in low localization accuracy. The Kinect V2 sensor used in our experiments dynamically adjust the exposure time

under low lighting conditions. The images produced contain very noticeable noise as shown in Figure ??.

Therefore, we also tested the algorithm under harsh lighting conditions in a real world setting. The data were captured under 4 different lighting conditions: 20 lux (dark), 43 (lux) dim, and (90 lux) normal, 243 lux (bright). As results shown in Figure 8, the localization accuracy significantly improves with better illumination.

The Kinect sensor automatically adjusts the exposure settings to compensate for the low lighting. The pictures captured in the dark rooms still appears bright but much more grainy and apparent Gaussian noise in the image. Depth sensor and RGB sensor works optimally under different lighting conditions. In the dark setting, the depth sensor performs well.

## VI. DISCUSSION

### A. Perceptual Uncertainty

An important problem is to formally define the perceptual uncertainty produced by any detection algorithm similar to those in SLAM. For example, if the detector knows the pose estimated have large margins of error, it is useful to have some notion of uncertainty around the estimated pose. These are especially helpful inputs to any motion planing algorithms that takes uncertainty into consideration. In our implementation, we have a loose definition of detection uncertainty. We implicitly treated the pose estimation from depth sensor and RGB sensor as two independent observations. Uncertainty of the pose can be described by the variance and differences between two observations. This can be achieved using a similar technique to the updating step in Kalman filters.

### B. Computation Time

We briefly tested the computation time of the new algorithm. With our current implmentation in Python, the algorithm can process a *pixelbypixel* image in *xxx* seconds. All tag detectors and the fusing process were running in a single-threaded mode of an Intel core. Since our sensory updates at roughly  $35Hz$ , the entire pipeline can process the tags in real time. There is no signifciant time increase on a higher resolution image because our fusing algorithm does not need to process the entire image. Therefore, the only time increase comes from the initial Apriltag detection process which has been shown to work under 30 ms for large images.

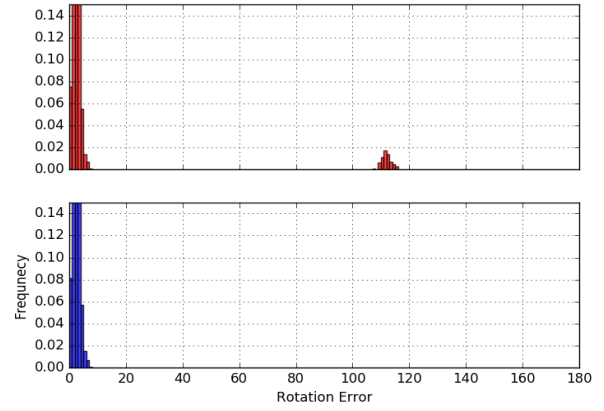
The most time consuming step is running the trust region optimization for refining the pose. This process can be sped up significantly by simply implementing the pipeline in C++.

## VII. CONCLUSION

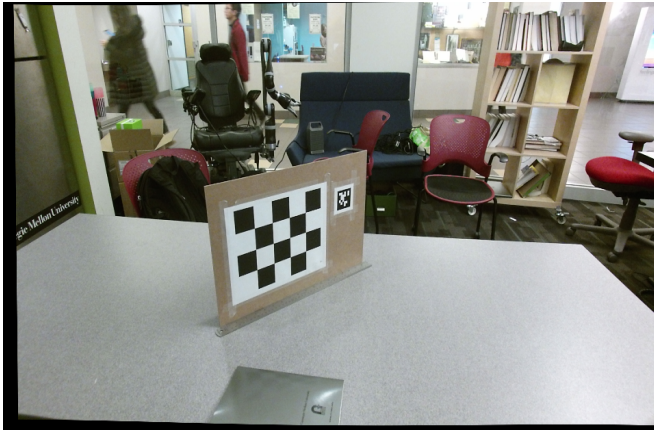
In this paper, we did a in depth analysis of the localization problem with Apriltags. We proposed a novel algorithm of using RGBD sensors to accurately compute the pose of Apriltags robust to noise. It is particularly suitable for robotic applications which requires precise poses such as manipulation, SLAM, and others. Furthermore, this technique can be easily generalized to other types of planar fiducial



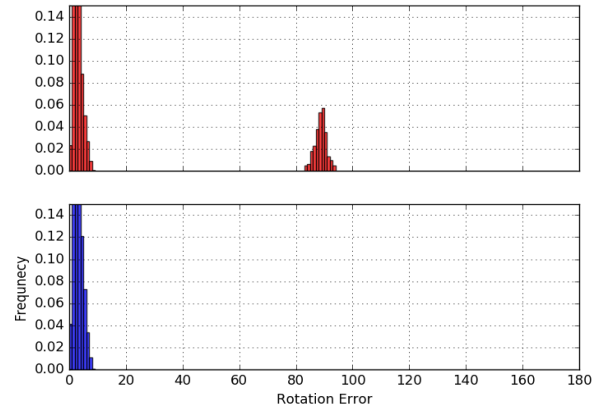
(a) RGB Image at 75°



(b) Distrubtion



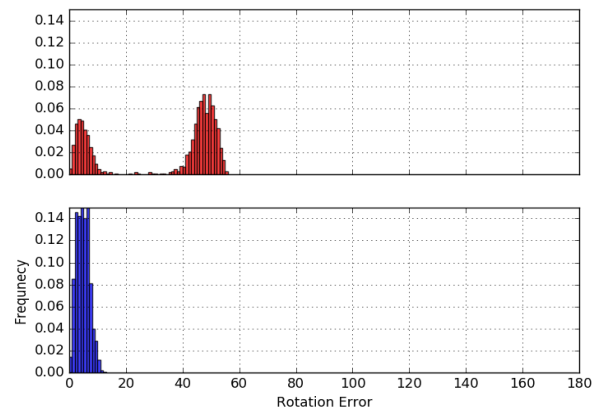
(c) RGB Image at 40°



(d) Close



(e) RGB Image at 5°



(f) Close

Fig. 9: The fiducial tag plane captured by the depth sensor

tags. Our implementation is fully open sourced and available at: <http://somegithublink.com>