Title

Pengju Jin, Siddhartha Srinivasa

***Abstract*—Abstract Goes here**

I. Introduction

Talk about Fiducial tags in general.

In most robotic applications, it is important to be able to accurately and reliably calculate the pose of the fiducial tags. There has been large amounts of effort for improving the detection algorithm by making them faster and more accurate using the RGB images. These algorithm yield great results under ideal and simulated conditions. (Provide reference to specific studies). However, in real robotic application, these algorithms are often tested under various lighting and sensory noises. In these conditions, the fidicual tags suffers greatly from the perceptual ambiguity problem and makes the pose estimation difficult without additional information. In fact, we observe that the localization accuracy of the state of the art Apriltags is significantly worse at difficult view angles or when there are noise in the scene.

In this paper, we present an algorithm that take advantage of the RGBD sensor to accurately estimate the pose from a single tag under noisy conditions. There are few key features to this algorithm:

- The algorithm is generalizable to all square based fiducial tags.
- The algorithm performs at worse as good as only using RGB images.

II. Related Work

Related work here

III. Perceptual Ambiguity

In most square fiducial tag detection, the pose of the tag is calculated from the quad fitted around the tag. The corners are extracted from the tag and the pose of the tag is estimated using some 3D to 2D point correspondence optimization. This is a specific case of the perspective-n-point problem and it has been well studied in [][]. In particular, there is a determinastic solution to the persepctive-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 persepctive 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.

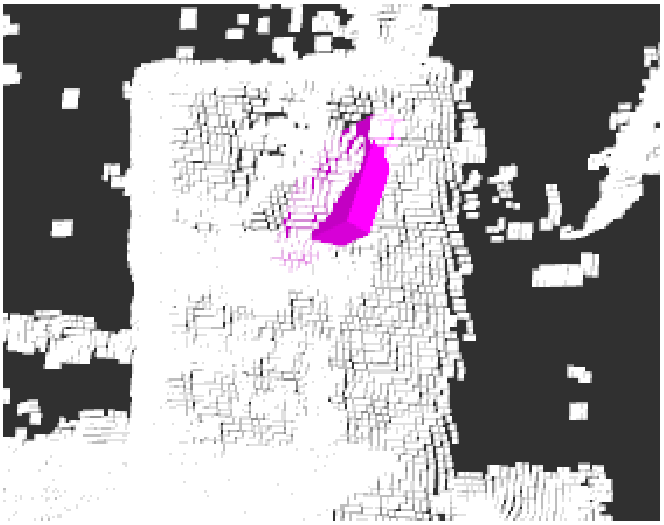


Fig. 1: The orientation of Apriltag placed on the object is greatly misaligned with the actual object

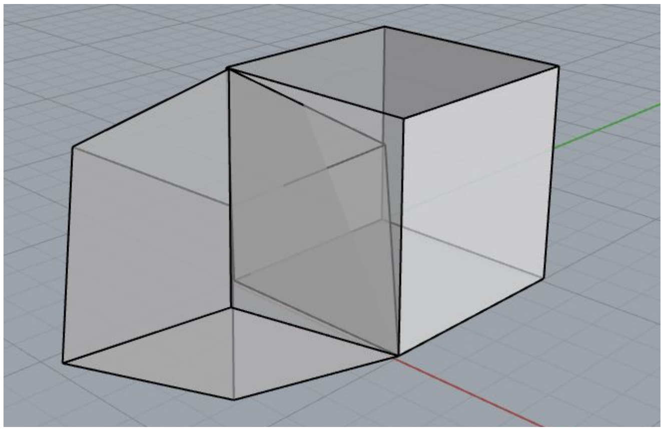


Fig. 2: Perspective Ambiguity illustrated with overlapping cubes

We will illustrate this effect by using two over lapping 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 over lapping squares becoming 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 distrubtion depending on the viewing angle. The result of the 3D to 2D correspondence optimization might return either one of the two solution.

IV. Approach

In order to obtain reasonable and reliable pose of the fiducial tag under noisy condition, we fused the RGBD data of the Kinect One sensor. Our system first uses the depth sensor data to calculate an initial pose estimation of the detecting tag. Afterwards, we further refine the pose estimation by minimizing the reprojection error in RGB image using a constrained optimization method. In this section, we describe how we obtain the initial pose estimation of the tag

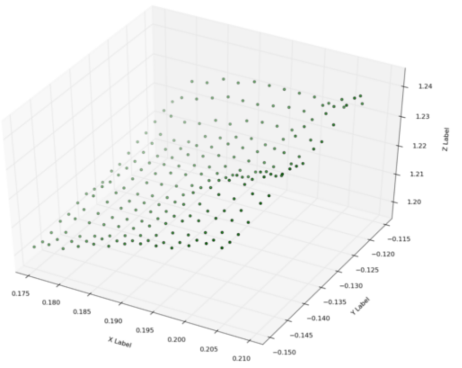


Fig. 3: The fiducial tag plane captured by the depth sensor

using depth sensor data and fuse it with rgb information to refine the pose. Our key insight is that we can constrain the pose refinement process by weighting the sensor information with their uncertainty. In order words, we will weight the initial pose estimation more when the depth sensor is good otherwise we will allow the region of refinement to be bigger.

A. Depth Plane Fitting

Since the fiducial tags are planar, we can obtain the pose the tag using depth sensor by fitting a plane over the depth points. Although the corners of the fiducial tags found in the RGB images might be noisy, they are accurate enough to locate the rough region of the tag with at most a few pixels of offset. With a calibrated RGBD sensor, we can extract the patch of depth points containing the planar tag.

The raw range data retrieved from the Kinect One sensor are usually far from perfect for fitting an exact plane. 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. Furthermore, the remaining points could have a large variance depending on the background lighting and the magnitude of the plane rotation. In the cases of depth sensor are noisy, we want to lower the weight of our initial pose estimate later in the pipeline.

Therefore, we implemented a Bayesian plane fitting algorithm (describe the algorithm below) which assumes some noise model of the data and computes the mean and covariance of the plane. The noise model we used is a Kinect specific noise model [reference], which increases with the axis rotation of the depth plane. Note that a higher covariance implies that we are more uncertain about our fitting plane.

B. Initial Pose Estimation

Once the plane is computed, we can estimate the pose of the tag. We project the noisy corners of the apriltag back onto the estimated plane to get their 3D coordinates in the camera frame. (Write out the formula) From here, the pose of the tag can be described as the homogenious transformation from the tag frame to the camera frame. We can solve for the transformation with a set of two 3D points correspondence: The 3D projected points of the tag in the camera frame and the 3D points in the tag frame. The points in the camera frame would be the projected corners which forms a $4X3$ matrix of homogeneous points denoted as Pt_{obj}. The points, Pt_{tag} in the tag frame are the coordinate of the four corners measured from the center of the tag with depth of 0 because they are planar.

$$Pt_{obj} = HPt_{tag}$$

This becomes a 3-D rigid body transformation estimation problem. In our implementation:

$$\begin{aligned}
p &= mean(Pt_{obj}) \\
q &= mean(Pt_{tag}) \\
x &= Pt_{obj} - p \\
y &= Pt_{tag} - q \\
U, S, V^T &= \mathbf{SVD}(x^T y) \\
R &= VU^T \\
t &= q - Rp
\end{aligned}$$

R and t are the rotation and translation components of the the homogenous transform. When we projected the points onto the depth plane, the squares will likely deform. Therefore, we use the above algorithm to find the optimal transformation H that minimizes the error in the least square sense. The pose obtained from the range data, although not always accurate, is rconsistent under noise and does not suffer from the perceptual ambiguity problem.

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}

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

The challenge here is to determine the region Γ_R and Γ_t which the initial pose estimate can be adjusted. If we unbound ΔR and Δt 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. Thererfore, this becomes a constrained optimization problem where the bound on the independent variables of ΔR and Δt 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 emperically tested to yield the best results for our robot.

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 demostrate in this paper: First, we want to characterize the effect of perceptual ambiguity and noise on Apriltags detection algorithm. Second, we want to test the resislience 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.

To do this, we measured the rotational and translation accuracy of the Apriltag under three different conditions: viewing angles, distances, and lighting conditions. In all three experiments, we introduced 3 different levels of simulated detection noise into our images. We placed a standard camera calibration chessboard and an Apriltag of known size on a solid planar board. The apriltag is has a fixed distance 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 noise.

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 gazing 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 data and computed for the resulting tag pose. This is repeated for 1000 trails at each noise level and the errors are computed for each trial. Figure [7] shows some of the results.

As the result shown in figure 7, the viewing angle has large effect on the rotation error. As we expected, the emperical results show a very clear bimodel 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 thresholded all the poses based on their rotational errors and ploted 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 locationlization accuracy. At most viewing angles, relatively small noises casuses a signficant accuracy decrease.

B. Distance

In additional to the viewing angle, we caputred the images at different distances away from the camera. We moved the testing board perpendiualr to the sensor.

The relationship between the distance and localization accuarcy is much more apparent. As the tag moves further away from the sensor, the number of pixels on the tag decreases. The perspective ambiguity becomes more apparent when there is only a small patch of pixels in the tag.

C. Lighting

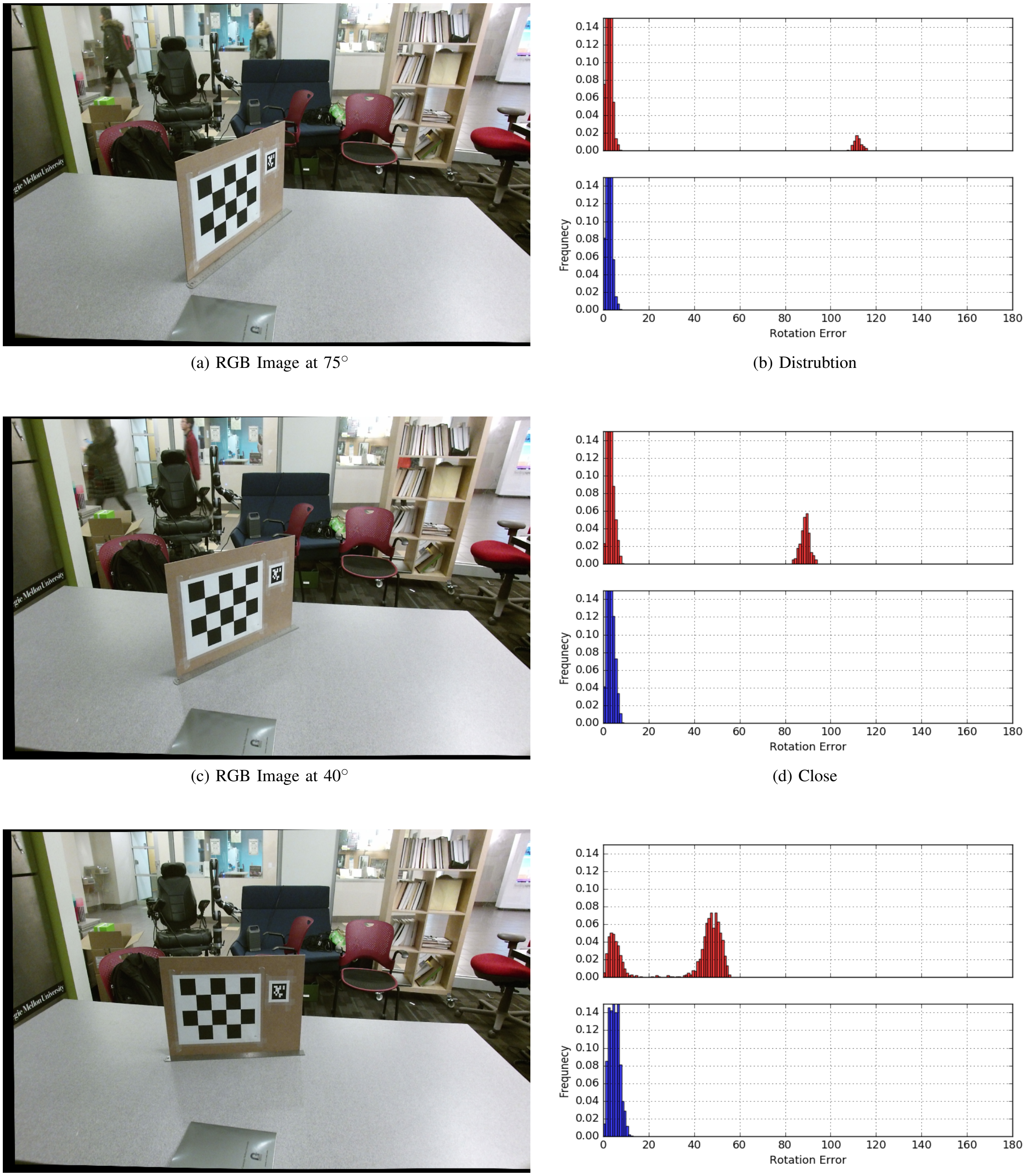
In the unsimulated environemnt, poor lighting is a large contributing factor of sensory noise. We tested the effect of the lights on our detection process by controling the background lighting. We captured the pictures under three different lighting conditions: dark, normal, and highly exposed.

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.

D. 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++.



(a) RGB Image at 75°

(b) Distrubtion

(c) RGB Image at 40°

(d) Close

(e) RGB Image at 5°

(f) Close

Fig. 4: The fiducial tag plane captured by the depth sensor