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, ARToolkil, 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:

- 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 subpixel 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 or 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 pay load.

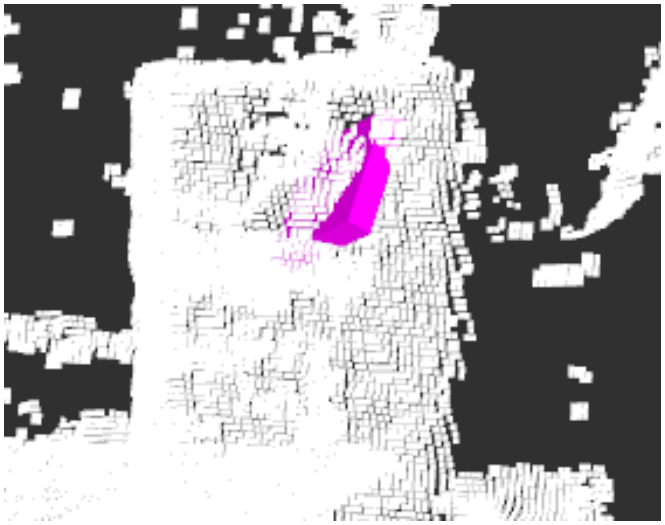


Fig. 1: The orientation of Apriltag placed on the object is greatly misaligned with the actual object

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 [][]. 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 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 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 return either one of the two solution.

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 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. Second, 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.

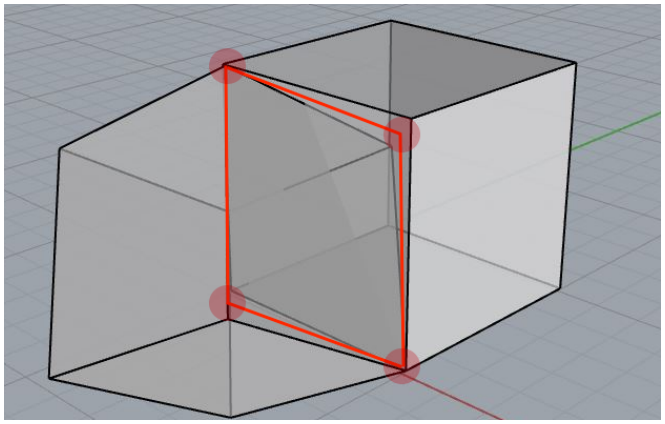


Fig. 2: Perspective Ambiguity illustrated with overlapping cubes

The key insight to our method is that we should weight the observations in the pose estimation process by their uncertainty. For instance, if the initial pose estimation is precise, then

A. Depth Plane Fitting

Since the fiducial tags are planar, we can obtain the pose the tag using a 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 region of the tag offset by at most a few pixels. 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 where depth sensor are noisy, we want to lower the weight of our initial pose estimate later in the pipeline.

In implementation, we used 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

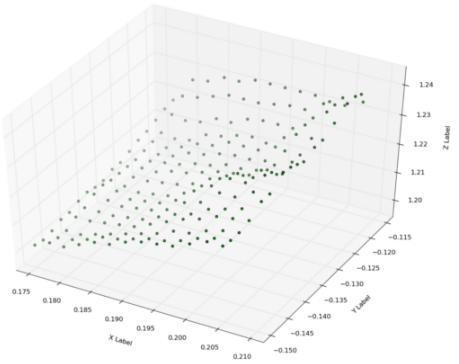


Fig. 3: The depth points of the Apriltag plane. The points are noisy but good for an approximate estimation.

transformation with a set of two 3D point correspondences: the projected points of the tag in the camera frame and the points in the tag frame. The points in the camera frame would be the projected corners which form a $4x3$ matrix of homogeneous points denoted as p_i. The points, q_i 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. We can formulate this to be a 3D rigid body transformation estimation problem [Reference to SVD 3D rigid body]. Solving for the optimal transformation $[R, \boldsymbol{t}]$ requires minimizing a least squares error objective function given by:

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

There are numerous approaches to solve Eq. 1. Since we have very few correspondences and they are assumed to be correct, it can be computed quickly using SVD:

$$\bar{p} = \frac{1}{N} \sum_{i=1}^{N} p_i \qquad p_{ci} = p_i - \bar{p} \tag{2}$$

$$\bar{q} = \frac{1}{N} \sum_{i=1}^{N} q_i \qquad q_{ci} = q_i - \bar{q} \tag{3}$$

$$p_c^\top q_c = U \Sigma V^\top \tag{4}$$

$$R = V U^\top \tag{5}$$

$$\boldsymbol{t} = \bar{q} - R\bar{p} \tag{6}$$

R and t are the rotation and translation components of the the homogenous transform. When we project 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 consistent 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

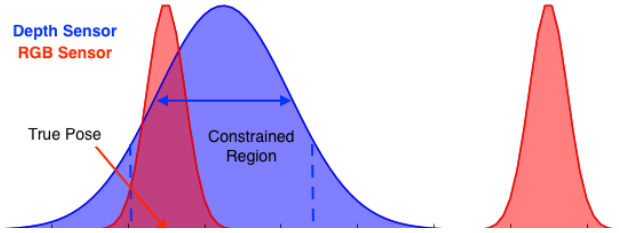


Fig. 4: An abstract visualization of the optimization constraints. The blue curve is the initial pose estimation obtained from the depth plane. The red curves are the ambiguous poses from the RGB image. We constrained the region of optimization based on how well we fit the depth plane.

error. Given camera model K, initial rotation and translation estimates R_{init} and t_{init}

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

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 intheon Apriltag detection algorithmson 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. 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 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

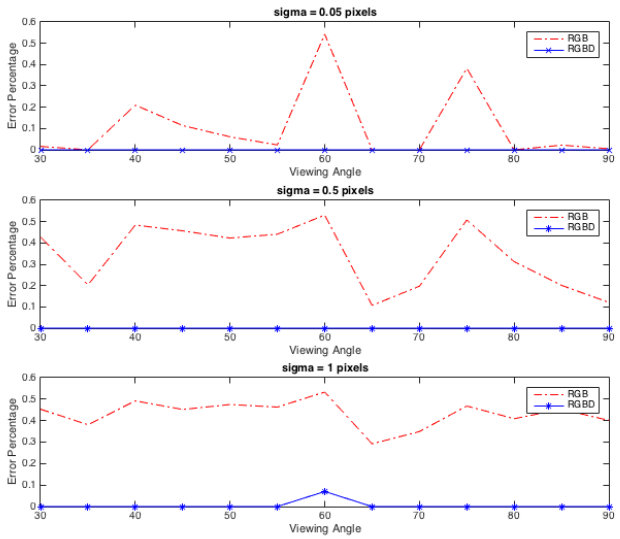


Fig. 5: Viewing Angle vs Error Percentage under different simulated noise level. The new RGBD based algorithm can resist noise in the RGB image and it vastly outperforms the original algorithm.

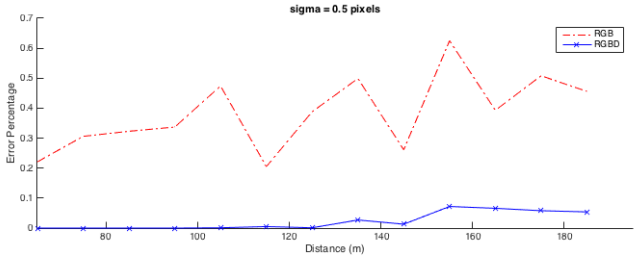
three different indepdent variables: viewing angles, distances, and lighting conditions. In all three experiments, we introduced 3 different levels of simulated detection noise into our images. We placed a standard camera calibration chessboard and an Apriltag of known size on a solid planar board. The apriltag 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 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 [7] shows some of the results.

As the result shown in figure 7 indicates, 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.

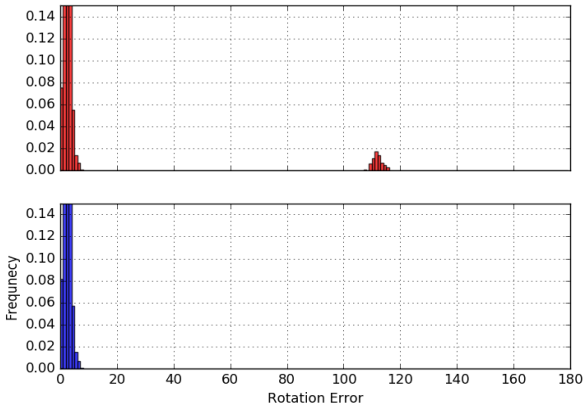
The Kinect sensor automatically adjusts the exposure settings to compensate for the low lighting. The pictures captured in the dark rooms still appears bright but much more grainy and apparent Gaussian noise in the image. Depth sensor and RGB sensor works optimally under different lighting conditions.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

VI. Discussion

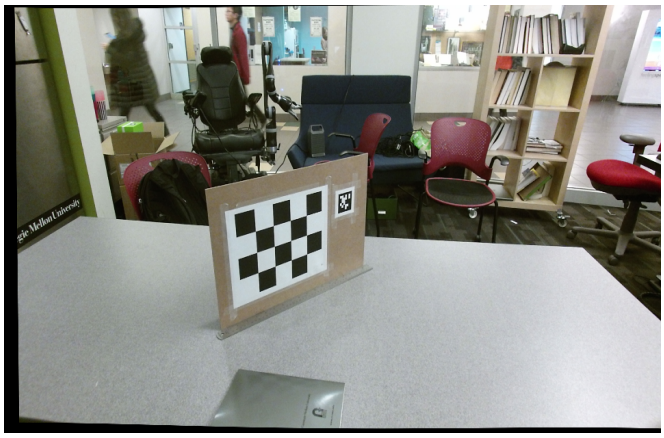
A. Perceptual Uncertainty

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.

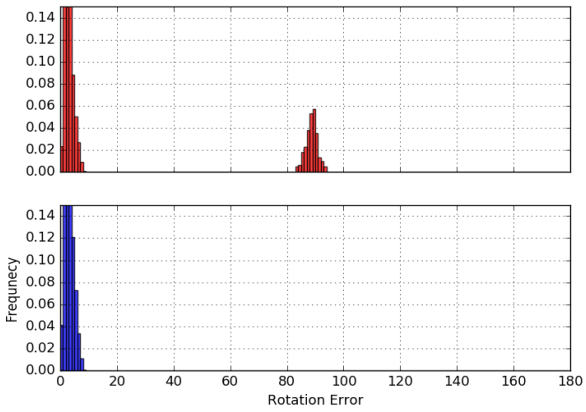
(a) RGB Image at 75°



(b) Distrubtion

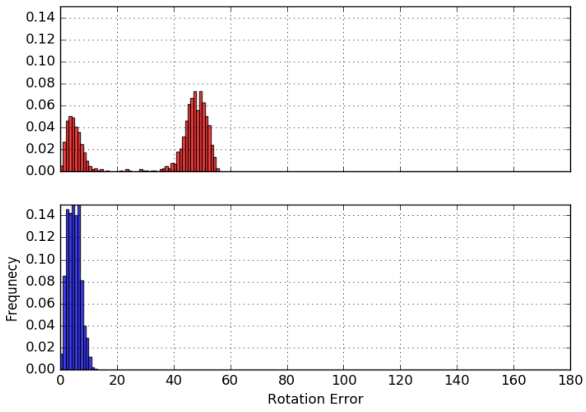


(c) RGB Image at 40°



(d) Close

(e) RGB Image at 5°



(f) Close

Fig. 6: The fiducial tag plane captured by the depth sensor

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 tags. Our implementation is fully open sourced and available at:

http://somegithublink.com