A Sensor Fusion Approach For

Pengju Jin, Pyry Matikainen, Siddhartha Srinivasa

***Abstract*—Although there is an abundance of planar fiducial marker systems proposed for augmented reality and computer vision purposes, none of them are robust enough to estimate the pose accurately in robotic applications where tags are small and the collected data are noisy. This is inherently a difficult problem because these fiducial marker systems work solely within the RGB image space and the resolution of cameras on robots is constrained. As a result, small noise in the image would cause the tag's detection process to produce large pose estimation errors.**

This paper describes an algorithm that improves the pose estimation accuracy of square fiducial markers in difficult scenes by fusing information from RGB and depth sensors. The algorithm retains the high detection rate and low false positive rate characteristics of fiducial systems while making them much more robust to size, lighting and sensory noise for pose estimation. The improvements make the fiducial tags suitable for robotic tasks requiring high pose accuracy in the real world environment.

I. Introduction

Detection and identification using artificial landmarks, known as fiducial markers, has long been used in augmented reality and computer vision applications. Over the last decade, there have been numerous marker systems, such as ARTags [1] Apriltags [2], and Rune Tags [3], designed to improve detection encoding precision. However, these systems suffer from the low pose accuracy problem when they are used in many robotic applications where the data is noisy.

Compared to markerless detection algorithms, fiducial marker methods are simple and easy to detect. They yield great results in augmented reality tasks that require high detection speed. Furthermore, the fiducial tags are popular in the robotic community due to their high detection rates and numerous encoding schemes. For example, Apriltags are commonly used to test SLAM systems, or finding ground truth for objects in manipulation and motion planning tasks as shown in Figure 1.

However, obtaining highly accurate pose estimations using fiducial tags from noisy data remains a challenge. This is especially important for robotic applications because small errors can cause large system failures as the errors propagate and amplify through the system. Currently, the fiducial tag systems yield promising results under well conditioned or rendered environments, but this does not translate to ill-conditioned settings. For instance, when Apriltags, a state of the art fiducial marker, are used with low resolution cameras or harsh lighting conditions, the system often produces poses with tremendous rotational errors. We observe that the Apriltag's localization accuracy performs significantly worse when there is noise in the captured image. This is a difficult problem because RGB sensors are often sensitive to lighting, and current fiducial systems are not designed to take advantage of other sensors commonly available on robots.

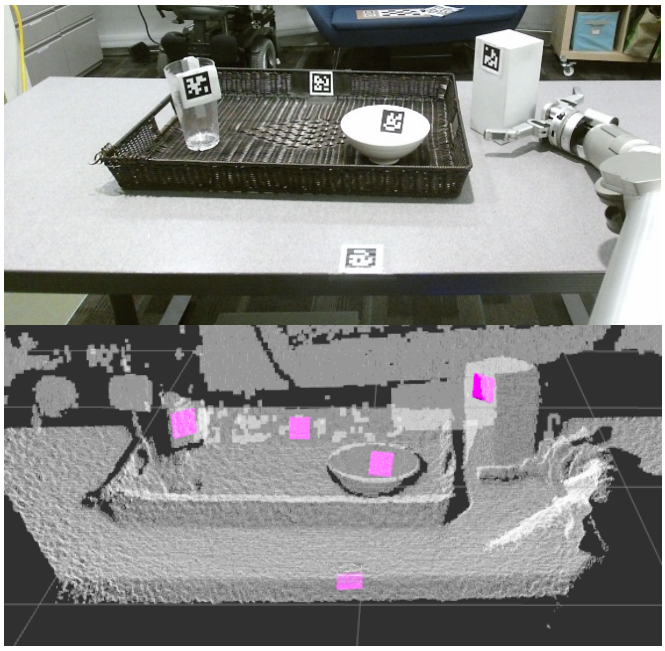


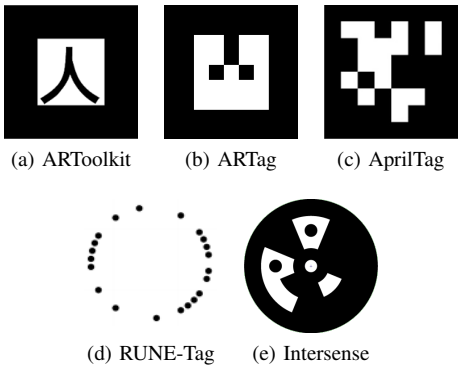
Fig. 1: Robot about to execute a manipulation task and rearrange the objects on the table. Apriltags are used to find the poses of targeted objects in the scene but ultimately fails to gasp the prism because the orientation of its pose is wrong.

We present two main contributions in this paper. First, we conducted an in-depth analysis on the effect of various noises on the 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 method 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 real time. Our key insight is that RGB and depth sensors work optimally under different conditions. We can leverage this to improve the pose estimation process and increase the localization accuracy under difficult conditions. The key features to this algorithm are:

- This method is highly robust to noise in the scene. It can obtain accurate poses suitable for a wide range of robotic applications.
- It is easily generalizable to most fiducial tag designs.
- It has very small computation overhead, and can be ran in real time.

This paper also presents empirical results demonstrating the successful performance of the algorithm on captured data from a humanoid robot. Our implementation of the algorithm



(a) ARToolkit (b) ARTag (c) AprilTag

(d) RUNE-Tag (e) Intersense

Fig. 2: Different types of popular fiducial tags. ARToolkit, ARTags, and AprilTags are square tags with black borders. RUNE-tags and Intersense use different circle features as landmarks

is based off of the Apriltag detection pipeline and it is integrated with ROS.

II. Related Work

Obtaining highly accurate pose estimation has been an important research area in robotics. Numerous algorithms rely only on RGB or gray scale images. Solving the projection geometry of some detected features and then minimize the reprojection error of the features in the image space [4]. Similarly, methods such as Iterative Closet Point [5] were developed to solve the pose estimation problem using range data by minimizing the Euclidean distance between the model and the depth data. Recently, some approaches propose to enhance the accuracy of traditional algorithms by fusing RGB and depth data in various problems using extended Kalman filters [6, 7]. Compared to the single sensor approaches, algorithms utilizing RGBD data are more accurate and perform well in noisy situations where other approaches fail.

Fiducial markers solve pose estimation by exploiting easily detectable features in the RGB space. Although there is an abundance of unique tag designs, most of them carry easily recognizable yet precise binary patterns in the inner region to encode information. There are two types of common tags: circular tags and square tags seen in Figure 2.

Circular tags are created to encode the payload using small circular patterns arranged in various shapes. The example of circular tags include Intersense [8] and Rune tags [3]. The perspective transformation of a circle is an ellipse, which can be used to directly compute the pose using back projection methods. Localization of circular features is generally more accurate, and thus generates better pose estimation at the cost of higher computation time [9]. However, small circular features become hard to detect when they are far away from the camera or prospectively rotated, and thus their effective range is much smaller than that of square tags. This characteristic makes them less useful in applications with size constraints.

ARTags [1], ARToolkit [10], AprilTag [2] and AprilTag 2 [11] 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. However, since the tags are detected using rectangles and lines, the accuracy of their corner point subpixel locations is limited. Among the square tags, ARToolkit is one of the earliest detection systems, and it is mainly used for Augmented reality applications. Instead of using a binary payload, it used various symbols to encode the tag, and it was computationally expensive to decode the tag. Built on top of ARToolkit, ARTags and Apriltag reduced the computation time by using a 2D binary pattern as the payload. Both systems use the image gradient to compute the tag border making it robust to lighting changes and partial occlusions. Relative to ARTags, Apriltags have a lower false positive rate, as they use a lexicode-based system that is invariant to rotation. In addition, Apriltags have higher detection rates at further distances and at more difficult viewing angles. Recently AprilTag 2 improved upon the original Apriltag. It implements a new boundary segmentation algorithm which further reduces the computing time for detection and increases the detection rate. Compared to circular tags, the advantages of square tags are that they can be located very efficiently and they have reliable decoding schemes. Therefore, even though the square tags have slightly lower localization accuracy, they are more suitable for robotic applications that require a robust system.

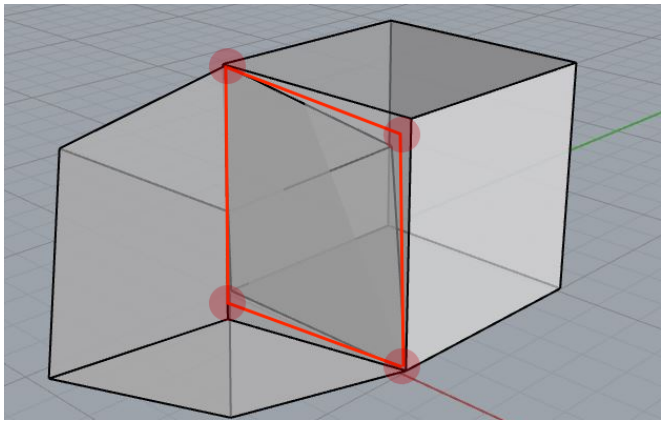
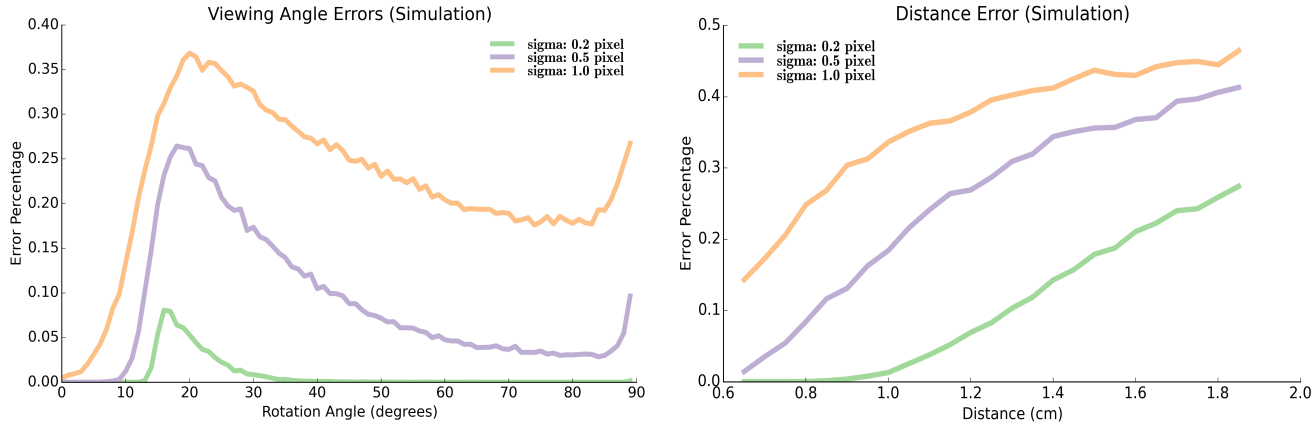


Fig. 3: The ambiguity effect is illustrated with two rendered cubes in the perspective view. The two cubes are rotated such that two faces are interlaced. The red square is a simulated projection of a square tag. The red circular regions denote the region of potential corner detections in a noisy scene. The pose of the red square can converge to either one of the two faces.

III. Challenges

In square fiducial marker detection, the pose is calculated by using the four corners of the tag. Since the tags are planar, it is easy to compute perspective point correspondences from the corners. This can be formalized as a specific case of the Perspective-N-Point problem and it has been well studied in geometry based Computer Vision literatures [12, 13]. There are numerous optimization methods such as ones proposed in [14] and [15] to solve this problem. In particular, in [16], the author shows that there is a deterministic solution to the Perspective-4-Point (P4P) problem when the points are coplanar. In other words, given the projection of a tag's



(a) Incorrect Pose Percentage w.r.t Viewing Angle

(b) Incorrect Pose Percentage w.r.t Distance

Fig. 4: Simulation results of Apriltag pose estimation under different noise level. In both simulations, four corners are projected onto a rendered image with some noise then computed their poses. In 4a, corners are projected to the center of the camera 0.8 meters away and rotated from 0° to 90°. In 4b, the corners are projected at 40° and moved from 0.6 meters to 1.8 meters. The resulting poses are threshold to calculate the percentage of unacceptable poses.

four corners, the pose of the tag is unique. In practice, however, these methods are very sensitive to noise in the scene. When ARTags, Apriltags and ARToolkit systems are used in scenarios shown in Figure 1, the poses of the tags are unstable even when the scene is static. Since the minimal number of perspective points are used to estimate the pose, a small variance in the corner detection process will yield estimations far from the true pose.

We will illustrate an ambiguity effect caused by noise by using two over lapping cubes, shown in Figure 3. The overlapping face of the two cubes are interlaced but rotated by 120 degrees. However, due to perspective projection, the squares appear to be on the same plane. With low camera resolution, the overlapping squares become virtually indistinguishable. The red circular regions are the detected corners under some sensory noise. Even though the reprojection error is minimized in the 2D space using P4P optimization methods, the 3D pose can still be far off. The result of the optimization can be characterized as a bimodal distribution and a function of the the viewing angle. Depending on the noise level in the scene, the optimization might converge to either one of the local minima causing the pose estimation to be unreliable. In Figure 4, we ran the Apriltag pipeline on simulated tags with small Gaussian noises introduced to the image. The results show the percentage of poses that have more than 30° of rotational error.

IV. Approach

This section describes a method for accurately estimating poses for square fiducial tags in noisy settings by fusing RGBD data. The process of detecting and decoding the tag is identical to previous fiducial tag systems. After the tag corners are detected, they are treated as approximated locations of the true corners. Using the corners, the method implicitly evaluate the depth data and RGB data as two separate observations and fuse them to minimize the error in 2D and 3D space.

There are three distinct components to this method. First, we find the plane in $SO(3)$ containing the tag using depth data and detected corners. Secondly, an approximate initial pose is computed using the depth plane. Finally, the method refines the initial pose using the RGB data by minimizing the reprojection error within a constrained space. Each component is described in detail in the following subsections.

A. Depth Plane Fitting

The first step is to extract the plane which the tag is laying on. We assume that the RGBD sensor is calibrated such that depth and RGB streams are registered to the same frame. The rectangular patch of points in the depth image bounded by the approximated corner pixels $\boldsymbol{y} = [y_1, y_2, y_3, y_4]$ contains the range information of all the points on 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 depth sensors are generally noisy. The borders and dark regions of the tag produce unreliable range data and artifacts due to a weakness of our depth sensors (time of flight sensor from Kinect V2). Therefore, we first filter the data by removing points too far from the median before fitting the plane. Nevertheless, the remaining points could have a large variance depending on the lighting condition and the magnitude of the in-plane rotation. The accuracy of the plane fit and initial pose estimation is directly affected by the noise level of data. We will characterize the uncertainty of the plane fit and adjust the weight of the initial estimation accordingly during the fusing stage.

In implementation, we used a Bayesian plane fitting algorithm described in [17] which computes the Hessian Normal parameters $[\hat{\boldsymbol{n}}, d]$ of a plane for noisy range data through

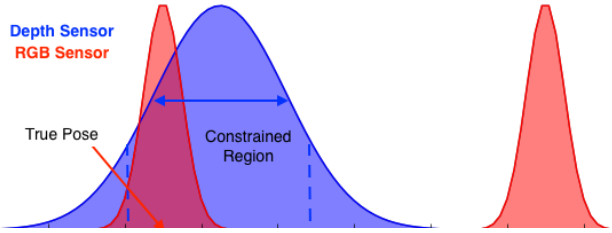


Fig. 5: An abstract visualization of the optimization constraints. The blue curve is the initial pose estimation obtained from the depth plane. The red curves are the ambiguous poses from the RGB image. We constrained the region of optimization based on how well we fit the depth plane.

optimizing

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

where $\hat{\boldsymbol{n}}$ is the local normal to the planar surface of the depth point and $\hat{\boldsymbol{m}}_{\boldsymbol{j}}$ is the measurement direction for the sensor for point p_j. 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

$$\sigma\{\bar{p}_j\} = \frac{kd^2}{\|\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{m}}_{\boldsymbol{j}}\|} \tag{2}$$

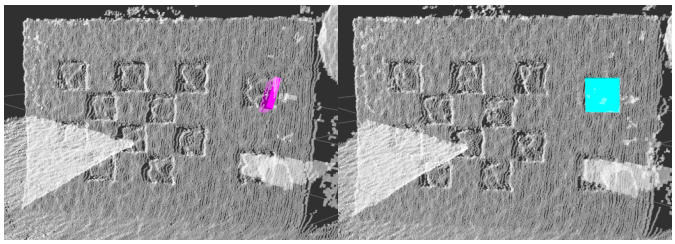
The coefficient $k > 0$ is an estimated value obtained from sensor calibration. In our implementation, we obtained k by using the Kinect V2 model obtained from [Kinect Noise Model paper].

An important result we used from [17] is the covariance matrix for the plane-parameters. The covariance is obtained by taking the *Moore-Penrose generalized inverse* of Hessian matrix computed from the Lagrangian. It characterizes the uncertainty of the plane fit and implicitly measures the relative accuracy of the depth data.

B. Initial Pose Estimation

The 6 DOF pose of the tag can be described as the transformation $[R, \boldsymbol{t}]$ aligning the tag frame's coordinate system and the sensory frame's coordinate system of the robot. The depth plane $D[\hat{\boldsymbol{n}}, d]$ alone is insufficient to determine the transformation as it only defines 3 DOF. Since the depth plane was computed base on the approximate center of the tag, we can use the center of the tag and center of the plane as a pair point correspondence. However, there are still infinite number of valid poses rotating about the normal $\hat{\boldsymbol{n}}$. One solution is to constrain the pose by using a corner as an extra point correspondence to solve for the optimal rotation. In practice, 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 $D[\hat{\boldsymbol{n}}, d]$ to get the coordinates $\boldsymbol{p} = [p_1, p_2, p_3, p_4]$ in the robot sensory frame. The corner coordinates $\boldsymbol{q} = [q_1, q_2, q_3, q_4]$ 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, and the coordinates are simply the location of the corners on a Cartesian plane. Given these two sets of 3D point correspondences, the pose can be computed



(a) RGB (b) RGBD

Fig. 6: The pose of the Apriltag visualized in RViz computed using the original library VS our RGBD fused method.

as a rigid body transformation estimation. Solving for the optimal transformation $[R, \boldsymbol{t}]$ requires minimizing a least squares error objective function given by:

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

There are numerous approaches to solve Eq. 3 described in [18]. Since we have very few correspondences and they are assumed to be correct, it can be computed quickly using SVD:

$$\bar{p} = \frac{1}{N} \sum_{i=1}^{N} p_i \qquad p_{ci} = p_i - \bar{p} \tag{4}$$

$$\bar{q} = \frac{1}{N} \sum_{i=1}^{N} q_i \qquad q_{ci} = q_i - \bar{q} \tag{5}$$

$$p_c^\top q_c = U \Sigma V^\top \tag{6}$$

$$R = VU^\top \tag{7}$$

$$\boldsymbol{t} = \bar{q} - R\bar{p} \tag{8}$$

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, provide a good approximation for the true pose.

C. Pose Refinement

Lastly, the pose is refined by minimizing the reprojection error using the initial pose estimated from the previous step. The camera is assumed to be calibrated and the camera projection model K is known. We denote R^* and $\boldsymbol{t}^*$ to be the optimal pose in the constrained optimization function

$$[R^*, \boldsymbol{t}^*] = \underset{R^*, \boldsymbol{t}^*}{\operatorname{argmin}} \sum_{i}^{n} \|(K[R^*|\boldsymbol{t}^*])\boldsymbol{p_i} - \boldsymbol{y_i}\|^2 \tag{9}$$

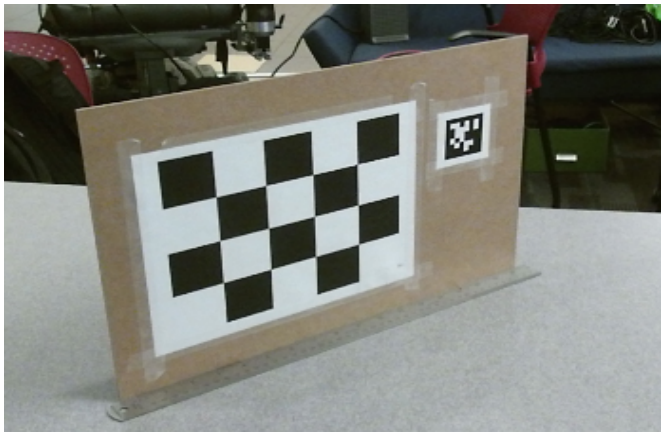
$$R^* = R(\Delta R) \tag{10}$$

$$\boldsymbol{t}^* = \boldsymbol{t} + R(\Delta \boldsymbol{t}) \tag{11}$$

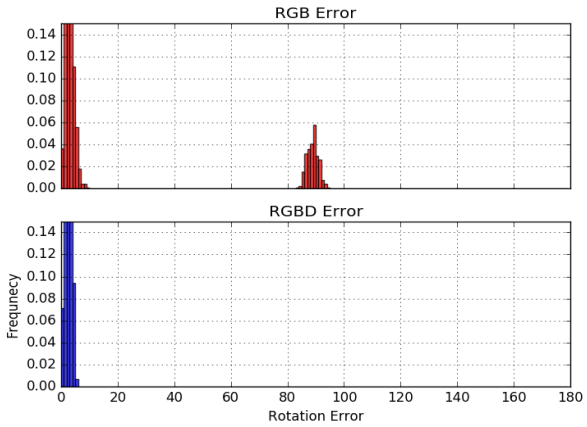
subject to:

$$\Delta R < \Gamma_R \tag{12}$$

$$\Delta \boldsymbol{t} < \Gamma_t \tag{13}$$



(a) RGB image at 60°



(b) Rotation errors across 1000 trials

Fig. 7: An example of the experimental setup in 7a. Groundtruth is computed from a large chessboard where the relative transformation to the tag is known. Each data collection, shown in 7b, is ran through 1000 trails and pose errors are measured.

Intuitively, the most optimal pose is the one with minimal reprojection error in the RGB space and align with the plane in the depth space. Therefore, the goal of the optimization is to find the local minimum closest to the initial estimation within allowable region Γ as illustrated with Figure 5. The key challenge is to determine the constrained region Γ_R and Γ_t such that it include a locally optimal pose and exclude the ambiguous pose. In most cases where the depth plane yields a good fit, this region should be small because the optimal pose is close to the initial estimate. When the depth sensor is noisy, the Γ increases since the initial estimate might be far off. Thus, the constrained region Γ is defined by the uncertainty in the initial estimate and it is characterized by the covariance of the plane parameters. In implementation, we used a trust-region optimization algorithm to bound the constraints. The scaling parameters for the covariance is empirically test to obtain the best results for our robot.

The strength of this method is that it harness the benefits of RGB and depth information without explicitly assuming their relative accuracy. One advantage of RGBD sensors is that the camera and the depth sensor often work optimally with different constraints. In the example of Kinect, the RGB camera is sensitive to lighting and works poorly in scenes with low illumination. However, the time of flight depth sensor are unaffected by such a problem. On the hand, the time of flight sensor yield poor range results on surface edges, but the RGB camera works exceptionally well with edges where there is a high color contrast.

V. Experimental Results

The key problem we are trying to resolve is the localization accuracy of Apriltags in noisy situations. Therefore, we want to test the resilience of our algorithm and show that it can obtain reasonable pose estimations under high level of noise. Figure 6 demonstrates an example visualization of the result. We also compare our method against *ar_track_alvar*, a popular ARTag detection package that incorporated depth information. Finally, we briefly tested the runtime of the algorithm to show that it remains capable of real time detection.

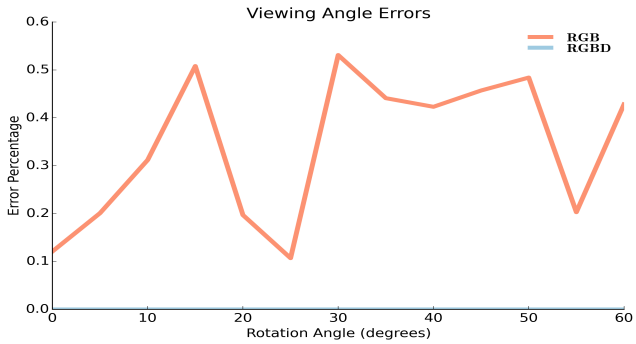


Fig. 8: Viewing Angle vs Error Percentage under different simulated noise level. The new RGBD based algorithm can resist noise in the RGB image and it vastly outperforms the original algorithm.

In our experiments, we measured the rotational and translation accuracy of the detection 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 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.

A. Viewing Angle

Due to the perspective ambiguity effect, the localization accuracy of the Apriltags is heavily affected by the viewing angle of the tag. To characterize the effect, we placed the testing board on a table straight in front of the robot as shown in 7a. Since the sensor is taller than the plane of the table, the robot has to slightly gaze down at it. We placed the testing board 0.65 meters away from the sensor and rotated it at a increment of 5 degrees from 0 degrees to 60. The angles

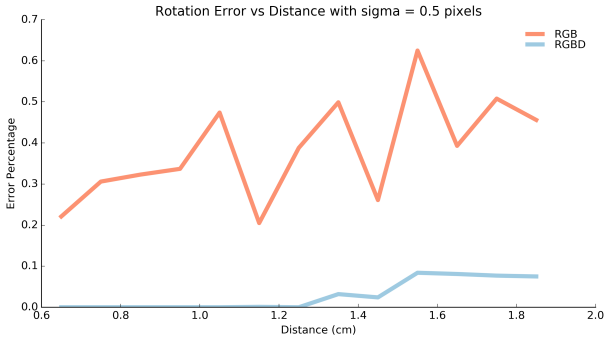


Fig. 9: Distance vs Error Percentage. Data are captured at a 10 cm increment from 65 cm to 185 cm.

are measured from the axis parallel to the sensor. This is about the range which the tag can be detected reliably given the camera resolution and the distance. At each angle, we captured the RGB image, depth image, and detection outputs from the Apriltag library.

For each captured data bundle, 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 for each data bundle per noise level and the errors are computed for each trial.

The empirical result in Figure 7b show a very clear bi-modal distribution, as we expected, for the detected poses for a given data bundle over 1000 trials. In Figure 8, we threshold all the poses based on their rotational errors and plotted the percentage of unacceptable poses at each viewing angle. The proposed RGBD fused algorithm vastly outperforms the original algorithm as it has better localization accuracy at all viewing angles and noise levels.

B. Distance

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. We show the results of both RGB and RGBD methods in Figure 9. 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 as they are in the simulation results.

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 increase in noise with the Kinect sensor at a further distance. In these cases, the variance of the depth plane estimation becomes very wide and the algorithm is unable to converge to the correct pose. Nevertheless, our method shows a significant gain in accuracy at every distance.

C. Lighting

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

(a) Dark (b) Dim (c) Normal (d) Bright

Fig. 10: Apriltags captured by Kinect V2 under different levels of illumination. The RGB sensor dynamically adjust the exposure time to compensate for low lighting. In (a), the image is captured outside of Kinect's adjustable range and the pixels are underexposed. In (b), the long exposure time introduced noticeable noise to the image.

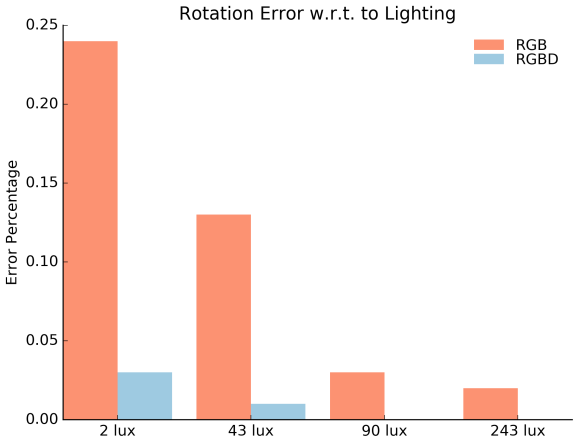


Fig. 11: Illumination vs Error Percentage. Data are captured at 65 cm away from the camera at a 40 degree angle.

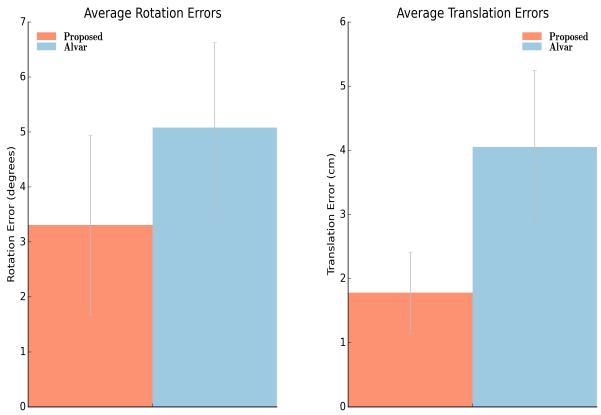
low lighting conditions. When pictures are taken below or near the adjustable range of the sensor, they contain very noticeable noise as shown in Figure 10.

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). We recorded a static scene over 5 seconds and randomly sampled 100 frames to run the test. As results shown in Figure 11, the localization accuracy significantly improves with better illumination. At the lowest illumination, nearly 25% of the poses were unacceptable (more than 20 degrees off) at the particular viewing angle. By using depth sensor which is unaffected by poor source radiance, there are only 3% of unacceptable poses.

D. Benchmark Against ar_track_alvar

ar_track_alvar is a ROS wrapper package for Alvar [19], an open source AR tag tracking library. The package is designed for AR tag detection and pose estimation for robots similar to Apriltags. In particular, it implements a module where depth sensor is integrated to improve the pose estimation. The package use the detected corner points to extract a patch of point clouds containing the tag. Then it proceed to fit a plane of the point cloud data and find its centroid using PCL. The pose of the tag is computed by aligning the centroid with the center of the tag.

We implemented a similar module for the Apriltag and compared the pose accuracy between our proposed method



(a) Rotation Error (b) Translation Error

Fig. 12: Average pose errors compared with ar_track_alvar package.

and the module using all the collected data. The results are shown in Figure 12. The two algorithms performed similarly in rotation error, but the proposed method was on average 2 cm better with the position component. The spread of error is also much smaller for the position component indicating that our purposed method is more consistent.

E. Computation Time

We briefly tested the computation time of the new algorithm. With our current implementation in Python, the additional computation time for sensor fusing process is 11 ms. Therefore the entire detection pipeline can process a 960 x 540 image within 35 ms. 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 and estimate the pose in near real time. There is no significant time increase on a higher resolution image for the fusing process because our algorithm does not need to process the entire image.

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++.

VI. Conclusion

An important extension of the purposed method is to formally define the perceptual uncertainty produced by the pose estimation algorithm. 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. This is especially helpful inputs to any motion planing algorithms that takes uncertainty into consideration.

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

References

[1] M. Fiala, "Artag, an improved marker system based on artoolkit," *National Research Council Canada, Publication Number: NRC*, vol. 47419, p. 2004, 2004.

[2] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3400–3407, IEEE, 2011.

[3] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello, "Rune-tag: A high accuracy fiducial marker with strong occlusion resilience," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 113–120, IEEE, 2011.

[4] D. Grest, T. Petersen, and V. Krüger, "A comparison of iterative 2d-3d pose estimation methods for real-time applications," in *Scandinavian Conference on Image Analysis*, pp. 706–715, Springer, 2009.

[5] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*, pp. 586–606, International Society for Optics and Photonics, 1992.

[6] O. S. Gedik and A. A. Alatan, "Rgbd data based pose estimation: Why sensor fusion?," in *Information Fusion (Fusion), 2015 18th International Conference on*, pp. 2129–2136, IEEE, 2015.

[7] A. Assa and F. Janabi-Sharifi, "A robust vision-based sensor fusion approach for real-time pose estimation," *IEEE transactions on cybernetics*, vol. 44, no. 2, pp. 217–227, 2014.

[8] L. Naimark and E. Foxlin, "Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker," in *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, pp. 27–36, IEEE, 2002.

[9] A. C. Rice, R. K. Harle, and A. R. Beresford, "Analysing fundamental properties of marker-based vision system designs," *Pervasive and Mobile Computing*, vol. 2, no. 4, pp. 453–471, 2006.

[10] H. Kato, "Artoolkit: library for vision-based augmented reality," *IEICE, PRMU*, vol. 6, pp. 79–86, 2002.

[11] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 4193–4198, IEEE, 2016.

[12] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[13] C.-X. Zhang and Z.-Y. Hu, "A general sufficient condition of four positive solutions of the p3p problem," *Journal of Computer Science and Technology*, vol. 20, no. 6, pp. 836–842, 2005.

[14] D. DeMenthon and L. S. Davis, "Exact and approximate solutions of the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1100–1105, 1992.

[15] B. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International journal of computer vision*, vol. 13, no. 3, pp. 331–356, 1994.

[16] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle, "An analytic solution for the perspective 4-point problem," *Computer Vision, Graphics, and Image Processing*, vol. 47, no. 1, pp. 33–44, 1989.

[17] K. Pathak, N. Vaskevicius, and A. Birk, "Uncertainty analysis for optimum plane extraction from noisy 3d range-sensor point-clouds," *Intelligent Service Robotics*, vol. 3, no. 1, pp. 37–48, 2010.

[18] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-d rigid body transformations: a comparison of four major algorithms," *Machine vision and applications*, vol. 9, no. 5-6, pp. 272–290, 1997.

[19] S. Niekum, "ar$_{t}rack_{a}lvarrospackage.$"