AprilTrack: A Particle-Filter-based AprilTag Tracker

Daniel Pfrommer

Abstract— **The AprilTrack algorithm proposed in this paper bridges two existing fields of research in computer vision: the detection of fiducial markers (bar code type tags), and monocular object tracking. Visual tags are highly popular for establishing ground truth location in visual odometry experiments and on mobile robots, but detection can be difficult under rapid camera motion and in poor lighting conditions due to motion blur. In contrast with other works proposing redesigned, more blur resistant tags, we use the popular AprilTag fiducials and improve detection quality by means of a particle filter based monocular object tracker. Our algorithm is able to reliably track a fiducial's position and orientation in six degrees of freedom, allowing for tag-based localization in situations where current detectors fail. This novel patch-based particle filter tracking algorithm, which is extended to simultaneously track multiple tags in ego-motion scenarios, is qualitatively and quantitatively evaluated on several AprilTag sequences, demonstrating that it is capable of reliably localizing blurred tags.**

I. Introduction

Visual tracking of fiducial markers (or simply fiducials) plays an important role in many robotics and computer vision related fields and applications, including augmented and virtual reality [2], visual odometry/SLAM [3], and object tracking. However, as many of these applications involve the use of fiducials in structured indoors environments where illumination is often limited, long exposure times to compensate for the reduced lighting produce blurred images, especially when taken from a mobile platform such as a quadrotor or a handheld device.

The focus of this work is primarily on providing better pose estimation for visual odometry datasets which use AprilTags placed to known locations to provide the ground truth pose data. As the pose estimation in this scenario can be done in a post-processing step after the data has been collected, the current implementation of AprilTrack is not designed to run in realtime. We intend to explore the use of AprilTrack in realtime scenarios in future research.

A. Previous Work

The majority of previous literature in fiducial detection has been focused on single-frame fiducial detection [1, 7, 5]. A common assumption in these algorithms is that the image being processed shows little motion blur, an expectation which breaks down in practice. For instance, the widely-used AprilTag marker [5] and its successor, the AprilTag 2 [8], rely on strong gradients between the tag's background and the surrounding white border to detect the tag. As shown in figure 1, oftentimes such algorithms fail to detect tags that have been blurred, either missing them entirely or mistakenly discarding them when eliminating false positives through some encoded fiducial ID (for AprilTags this consists of a 6x6 binary grid in the center of the tag which is used to verify the validity of the detection).

Previously, there has been interest in designing blur resilient fiducials. Meghshyam et al. [7] have developed ringed markers to ensure that a region of the tag will always be perpendicular to the blur direction. However, this approach, while resistent to linear blur, does not take into account more complex blurs (e.g resulting from the tag pivoting around the vertical axis). In addition, the increased resilience to blur comes at the cost of localization accuracy as only the center point of the ringed tag can be reliably determined [7], whereas with other fiducials, such as the AprilTag [5], full pose estimation in 6 degrees of freedom (DOF) can be done based on just a single tag. Another significant disadvantage of the ringed marker of Meghshyam et al. is the limited number of tag ID's, as their proposed encoding method allows for a maximum of 4 distinct tags [7].

Unlike previous attempts to engineer more resilient tags, the algorithm proposed here exploits the temporal coherence between successive frames in a video sequence to track a tag's location between successful detections. As many high-blur scenarios involve a continuous stream of images taken at equidistant time intervals (e.g a quadroter-mounted moving camera viewing tags or a stationary camera tracking an AprilTag-labeled moving object), we propose to use knowledge of a fiducial's prior location to augment the detection algorithm. As such, this paper presents a novel particle-filter-based tracking algorithm (AprilTrack) capable of tracking AprilTags in 6 DOF with both joint tag motion and camera ego-motion.

Previous work involving the tracking of blurred objects includes the Blur-driven Tracker (BLUT) framework proposed by Wu et al. [9] for tracking motion-blurred targets. BLUT,

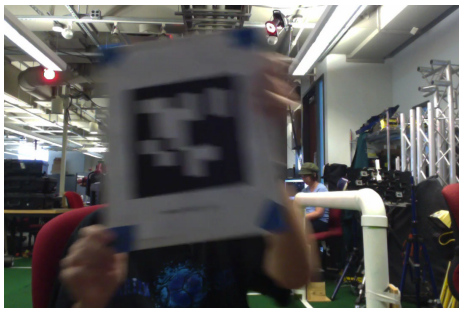

Fig. 1. A blurred AprilTag. Such tags are difficult for still frame based algorithms to detect as the payload (the ID encoded in the center of the tag) will cause the tag to be discarded as a false positive.

a particle-filter based tracker, works on the observation that blur strength and direction can be used to estimate an object's velocity and provide valuable information for successfully tracking a blurred object. While capable of reliably tracking blurred objects, BLUT suffers from the disadvantage that it can do so only in two dimensions (i.e it is assumed that the object always faces the camera) and so cannot track a sequence involving arbitrary tag rotations and translations.

The novel method proposed in this paper augments the AprilTag detection algorithm originally designed by Olson [5] and improved by Wang et al [8] with a particle-filter based algorithm to reliably track an AprilTag through an arbitrary sequence of images, providing fiducial-based position information when existing methods fail. In addition, the proposed algorithm is extended to incorporate the joint motion of an arbitrary number of tags to increase the accuracy of the tracked orientation and position.

Overview. The remainder of the paper is organized as follows. Section 2 gives an overview of the method proposed in this paper and describes the algorithm in detail. Section 3 describes the implementation of the algorithm and evaluates its performance on a manually-labeled dataset of blurred moving tags both quantitatively and qualitatively. In section 4 we discuss the limitations of this approach and suggest possible avenues of future research.

II. Method

The AprilTrack algorithm is similar in concept to the BLUT tracker by Wu et al. [9] in that it uses a particle filter to integrate all available information about a marker's position. Overall, the description of the algorithm can be broken down into three parts: one describing the particle filter, another on the method for evaluating candidate particles, and one describing the modifications made to simulatenously track multiple tags.

A. The Particle Filter

The particle filter used for the AprilTrack algorithm is a Bayesian Sequential Importance Resampling (SIR) technique which approximates a posterior distribution of state variables at discrete time intervals. There are three major steps in the algorithm: prediction, update, and resampling. We use the vector $\boldsymbol{x}_t$ to denote the 13 dimensional state vector describing the position, orientation, velocity, and rotational velocity of the tracked target at time t

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{r} \\ \dot{\boldsymbol{r}} \\ \boldsymbol{q} \\ \boldsymbol{\omega} \end{bmatrix} \tag{1}$$

where $\boldsymbol{r}$ is the three dimensional position of the tracked object, $\dot{\boldsymbol{r}}$ is the three dimensional velocity, $\boldsymbol{q}$ is a four dimensional unit quaternion containing the pose of the tracked object, and $\boldsymbol{\omega}$ is an angular velocity with components ω_x, ω_y, and ω_z.

Optimal filtering. The prediction step can be formulated in a Bayesian setting as the computation of the prior distribution $p(\boldsymbol{x}_t|\boldsymbol{z}_{t-1})$ from the filtering distribution $p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})$ where $\boldsymbol{z}_t$ represents the measurement (appearance of a fiducial) taken at time t. As in [6]:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1} \tag{2}$$

Likewise, the update step can be formulated in a similar manner, where the prior distribution calculated in the prediction step is updated with the latest measurement to obtain the posterior over $\boldsymbol{x}_t$. Mathematically this can be derived using Bayes' law to express $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ in terms of $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ and $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$. Using Bayes' law yields

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{z}_{1:t-1})$$

$$p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{z}_{1:t-1}) = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{z}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})p(\boldsymbol{z}_{1:t-1})}{p(\boldsymbol{z}_{1:t})}$$

As the model $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is Markovian, the current measurement is conditionally independent of previous measurements given $\boldsymbol{x}_t$, meaning that $p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{z}_{1:t-1})$ can be written as $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$. $p(\boldsymbol{z}_{1:t})$ can also be expressed as $p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})p(\boldsymbol{z}_{1:t-1})$ resulting in the final update equation [6]

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})} \tag{3}$$

Particle Filters.[1] Particle filters attempt to approximate the posterior distribution at time $t-1$, $p(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{1:t-1})$, using Monte-Carlo methods. A particle filter consists of a set of N samples $\{(\boldsymbol{x}^i_{t-1}, w^i_{t-1})\}_{i=1}^N$ drawn from the proposal distribution $q(\boldsymbol{x}^i)$ and weighted so as to take into account any discrepancy between $q(\boldsymbol{x})$ and $p(\boldsymbol{x})$ in that $w^i_t \propto p(\boldsymbol{x}^i_{0:t}|\boldsymbol{z}_{1:t})/q(\boldsymbol{x}^i_{0:t}|\boldsymbol{z}_{1:t})$.

As with the optimal filtering scenario, the prior distribution (the distribution of particles at the current timestep before taking into account the measurement) is calculated in the propagation step. In a particle filter, this simply amounts to drawing a new set of particles based using

$$\boldsymbol{x}^i_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}^i_{t-1}, \boldsymbol{z}_t) \tag{4}$$

where $\sim$ denotes sampling from the state transition distribution $q(\boldsymbol{x}_t|\boldsymbol{x}^i_{t-1}, \boldsymbol{z}_t)$.

Given the prior distribution, the particle weights must then be updated to reflect the target distribution $p(x)$. As previously noted:

$$w^i_t \propto \frac{p(\boldsymbol{x}^i_{0:t}|\boldsymbol{z}_{1:t})}{q(\boldsymbol{x}^i_{0:t}|\boldsymbol{z}_{1:t})} \tag{5}$$

We assume the proposal distribution $q(\boldsymbol{x})$ can be factored as

$$q(\boldsymbol{x}^i_{0:t}|\boldsymbol{z}_{1:t}) = q(\boldsymbol{x}^i_t|\boldsymbol{x}_{0:t-1})q(\boldsymbol{x}^i_{0:t-1}|\boldsymbol{z}_{1:t-1})$$

Since $p(\boldsymbol{x}^i_{0:t}|\boldsymbol{z}_{1:t})$ can be rewritten (in a similar manner to the derivation of (3) with Bayes' law) as

$$p(\boldsymbol{x}^i_{0:t}|\boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}^i_t)p(\boldsymbol{x}_t|\boldsymbol{x}^i_{0:t-1}, \boldsymbol{z}_{1:t-1})p(\boldsymbol{x}^i_{0:t-1}|\boldsymbol{z}_{1:t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})}$$

[1] A more in-depth discussion of particle filters can found in [6]. Much of the notation used in [6] is similar to that used here.

the equation from (5) can then be rewritten recursively in terms of w_{t-1}^i with

$$w_t^i \propto w_{t-1}^i \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i, \boldsymbol{z}_t)} \tag{6}$$

This allows us update the weights from the previous iteration and model the posterior distribution $p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t})$ with the set of weighted particles. The highest weighted particle is seleted as the most likely candidate.

To prevent the set particles from degenerating over time into a handful of highly weighted particles, resampling is applied. Resampling consists of drawing a new set of N particles from the discrete approximation of the filtering distribution $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ with the set of weighted particles:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta_{\boldsymbol{x}_t^i} \tag{7}$$

where the weights of the new resampled particles are then set to $1/N$.

The algorithm described and derived above is the Sequential Importance Sampling (SIS) filter. The AprilTrack tracker uses a Sequential Importance Resampling (SIR) filter, which deviates from the SIS algorithm in that resampling is applied after every iteration and the proposal distribution $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$ is simply the state transition distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$. This means that the propagate and update equations from (4) and (6) simply become:

$$\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i) \tag{8}$$

and (with the weights normalized to a total of 1):

$$w \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) \tag{9}$$

In AprilTrack, we model $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$ as a multivariate gaussian distribution with standard deviations $\boldsymbol{\sigma}$ around a mean $\boldsymbol{\mu}$ where

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{r}_{t-1}^i + \Delta t \dot{\boldsymbol{r}}_{t-1}^i \\ \dot{\boldsymbol{r}}_{t-1}^i \\ \boldsymbol{q}_{t-1}^i \dot{\boldsymbol{q}}_{t-1}^i \\ \boldsymbol{\omega}_{t-1}^i \end{bmatrix} \tag{10}$$

and where Δt is the elapsed time and $\dot{\boldsymbol{q}}_{t-1}^i$ is the quaternion representation of $\Delta t \boldsymbol{\omega}_{t-1}^i$, as described in [4]. We also use the *noise quaternion* formulated in [4] to apply the gaussian noise to the quaternion $\boldsymbol{q}_{t-1}^i \dot{\boldsymbol{q}}_{t-1}^i$. For simplicity, we make the assumption that the input images have been taken at equidistant time intervals and incorporate Δt into the $\boldsymbol{\omega}$ and $\dot{\boldsymbol{r}}$ components of $\boldsymbol{\sigma}$.

B. Evaluation of Candidate Particles

The tracking of AprilTags in 6 DOF requires a method for evaluating the likelihood of candidate particles with arbitrary three dimensional positions and rotations in order to calculate $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$. To this effect, the proposed method uses homographies to generate image *patches*, which are unprojected representations of an AprilTag for a particular pose and image.

Given the function $I(\begin{bmatrix}\lambda x & \lambda y & \lambda\end{bmatrix}^\intercal)$ representing the intensity of pixel at a particular (x, y) of an image, a patch in the image can be defined by the function

$$P(x_p, y_p|\boldsymbol{H}, I) = I(\boldsymbol{H} \begin{bmatrix} x_p & y_p & 1 \end{bmatrix}^\intercal) \tag{11}$$

where x_p and y_p are between -1 and 1 and $\boldsymbol{H}$ is a 3x3 matrix (a homography) mapping from $(x_p, y_p, 1)$ to $(\lambda x, \lambda y, \lambda)$ which incorporates the pose of an AprilTag relative to the camera.

This allows for the evaluation of candidate particle $\boldsymbol{x}^i$ by the measuring the similarity between the particle's patch $P^i(x_p, y_p|\boldsymbol{H}^i, I)$ in the current image and the patch from the AprilTag's last successful detection, $P_{ref}(x_p, y_p|\boldsymbol{H}_{ref}, I_{ref})$. Here, H^i can be calculated with

$$\boldsymbol{H}^i = \begin{bmatrix} \widehat{\boldsymbol{q}^i} & \boldsymbol{r}^i \end{bmatrix} \begin{bmatrix} \tau/2 & 0 & 0 \\ 0 & \tau/2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

where $\widehat{\boldsymbol{q}^i}$ is the rotation matrix representation of particle $\boldsymbol{x}^i$'s rotation quaternion $\boldsymbol{q}$, $\boldsymbol{r}$ is the particle's translation, and τ is the tag's edge length. The patch for the detected AprilTag uses the homography $\boldsymbol{H}_{ref}$ supplied by the AprilTag library, which computes the homography using the Direct Linear Transform algorithm [5].

To compare the patches efficiently, P^i and P_{ref} are evaluated at a regular grid from $(-1, -1)$ to $(1, 1)$ at intervals of $2/\rho$. This results in two $\rho \times \rho$ matrices $\boldsymbol{M}^i$ and $\boldsymbol{M}_{ref}$. To compare the patches, the error between the two matrices is calculated using the correlation coefficient between the individual elements of the matrices.

$$\epsilon(\boldsymbol{A}, \boldsymbol{B}) = \frac{1}{2} - \frac{\sum_{i=1}^{\rho}\sum_{j=1}^{\rho}(\boldsymbol{A}_{ij} - \mu_{\boldsymbol{A}})(\boldsymbol{B}_{ij} - \mu_{\boldsymbol{B}})}{2\sigma_A\sigma_B} \tag{13}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are the matrix approximations of two patches, $\sigma_{\boldsymbol{A}}$ and $\sigma_{\boldsymbol{B}}$ are the standard deviations of the respective matrices' elements and $\mu_{\boldsymbol{A}}$ and $\mu_{\boldsymbol{B}}$ are their means. By normalizing by mean and standard deviation, using correlation as a similarity metric (as opposed to the sum of the absolute difference or squared difference) ensures lighting invariance for comparing patches containing the same marker under different illuminations.

As in BLUT [9], the patch error $\epsilon(\boldsymbol{M}^i, \boldsymbol{M}_{ref})$ is used to compute the observation likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ to update the weights and select the best particle with

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = exp(-\gamma\epsilon(\boldsymbol{M}^i, \boldsymbol{M}_{ref})) \tag{14}$$

where the parameter γ allows for more or less selective resampling.

C. Multi-Tag Scenarios

To adapt this method to multi-tag scenarios, several extra parameters have to be introduced. In order to jointly track multiple tags, it is assumed that the relative poses of the tags are known. For each tag T_k with ID k, the location $\boldsymbol{x}_k$ and orientation quaternion $\boldsymbol{g}_k$ are expressed such that

$$\boldsymbol{X}_c = \widehat{\boldsymbol{q}^i}(\widehat{\boldsymbol{g}_k}\boldsymbol{X}_t + \boldsymbol{x}_k) + \boldsymbol{r}^i \tag{15}$$

where $\boldsymbol{X}_t$ is a coordinate relative to the tag's reference frame, $\boldsymbol{X}_c$ is relative to the camera reference frame, and $\boldsymbol{q}^i$ and $\boldsymbol{r}^i$ are the position and orientation of particle $\boldsymbol{x}_t^i$. In the simple case of jointly moving tags, the $\boldsymbol{H}^i$ from equation (12) can then be rewritten as

$$\boldsymbol{H}_k^i = \begin{bmatrix} \widehat{\boldsymbol{q}^i\boldsymbol{g}_k} & \widehat{\boldsymbol{q}}\boldsymbol{x}_k + \boldsymbol{r}^i \end{bmatrix} \begin{bmatrix} \tau_k/2 & 0 & 0 \\ 0 & \tau_k/2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

where $\boldsymbol{H}_k^i$ represents the homography for particle i and tag k and τ_k is the edge length of tag k.

In the ego-motion scenario, each particle's rotation and translation can be taken to be the camera's rotation and translation such that

$$\widehat{\boldsymbol{q}^i}\boldsymbol{X}_c + \boldsymbol{r}^i = \widehat{\boldsymbol{g}_k}\boldsymbol{X}_t + \boldsymbol{x}_k \tag{17}$$

resulting in the homography $\boldsymbol{H}_k^i$ and the corresponding patch approximation $\boldsymbol{M}_k^i$ where

$$\boldsymbol{H}_k^i = \begin{bmatrix} \widehat{\boldsymbol{q}^i}^{-1}\widehat{\boldsymbol{g}_k} & \widehat{\boldsymbol{q}^i}^{-1}(\boldsymbol{x}_k - \boldsymbol{r}^i) \end{bmatrix} \begin{bmatrix} \tau_k/2 & 0 & 0 \\ 0 & \tau_k/2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

In both ego-motion and joint tag motion, the observation likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ then becomes

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = exp(\sum_{k=1}^{K} -\gamma\epsilon(\boldsymbol{M}_k^i, \boldsymbol{M}_{k,ref})) \tag{19}$$

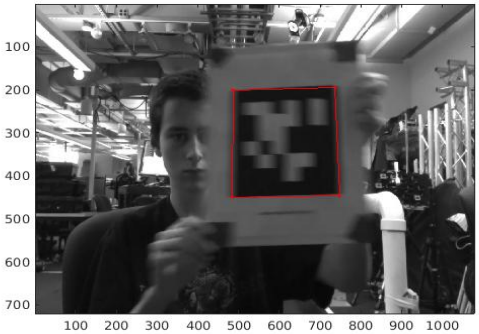

Fig. 2. A blurred tag being tracked. The red border indicates that the tag was not detected with the AprilTag detector.

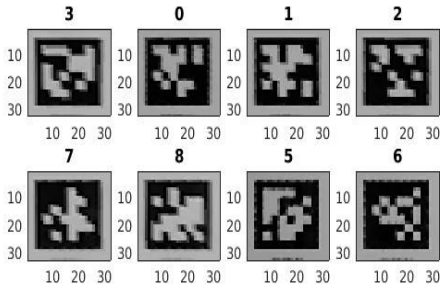

Fig. 3. The reference patches of various tags (by ID) in a multi-Tag tracking scenario.

where K is the number of tags being tracked and $\boldsymbol{M}_{k,ref}$ is the reference patch approximation for tag k from the last successful detection.

III. Results and Discussion

The AprilTrack algorithm was primarily evaluated on a manually-labeled single-tag sequence with a moving tag. The algorithm was also run on an ego-motion multi-tag sequence which will be qualitatively discussed. [2]

The specific implementation of the AprilTrack algorithm used in this experiment was written in MATLAB. For both scenarios, the parameters $\gamma = 10$, $\rho = 32$ were used. τ was 0.1936 m for all tags, slightly larger than the actual tag size of 0.1635 m to include some of the whitespace around the tag. In the single-tag scenario, the standard deviation $\boldsymbol{\sigma}$ of the gaussian noise used in the state transition distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$ was 0.01 for the $\boldsymbol{r}$ component, 0.05 for $\boldsymbol{q}$, 0.02 for $\dot{\boldsymbol{r}}$, and 0 for ω.

When evaluated on the single-tag sequence, the algorithm ran with $N = 3000$ at around 1 fps on an i7 Haswell Intel CPU. In experiments with fewer particles, it was found that the algorithm could track a tag well with as few as 1000 particles with around 5 fps. With a more optimized C++ implementation, the framerate could perhaps be brought up to 10 fps, making the algorithm (run with less agressive parameters) a possible candidate for real-time scenarios. Currently, the MATLAB implementation is intended for offline post-processing of a sequence and so real-time performance was not a focus of this research.

As shown in figure 4, the pixel corner error is about 10-20 pixels for most of the sequence. The tracker loses the tag momentarily around frame 30 but is able to recover rapidly without any manual intervention. Much of the error observed throughout the sequence is the result of disagreement on the boundary of a blurred tag rather than a systematic issue with the tracker itself.

[2]Videos of the sequence and the tracker's performance are available online at `https://www.youtube.com/watch?v=67LREs8fs9Q` and `https://www.youtube.com/watch?v=Rk87aE_9p-A`. The code is available online at `https://github.com/pfrommerd/tag-tracking-matlab`

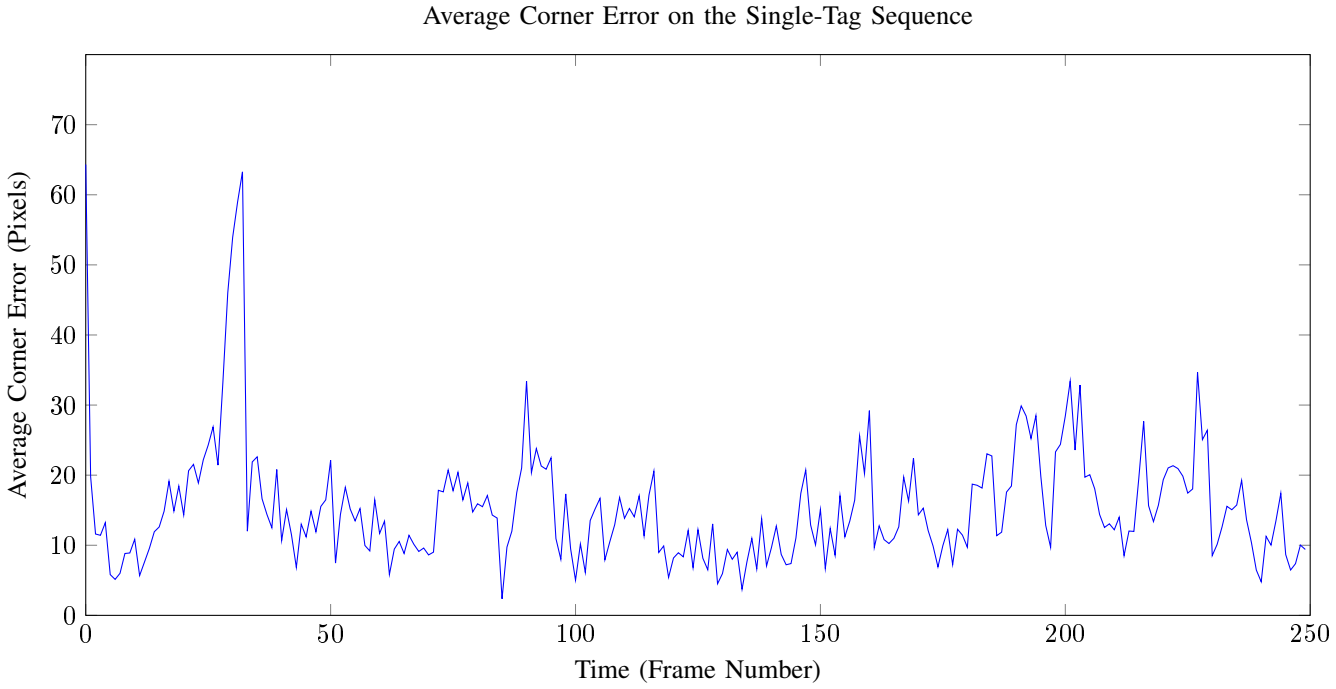

Fig. 4. The average corner error between the hand-marked tag location and the tracked location. Much of the error comes from discrepancy between the tracker and the ground truth on the boundaries of the blurred tags. At around frames 30 and 200, the tracker loses the tag momentarily due to a sudden rotation but is able to recover without needing to be manually reset.

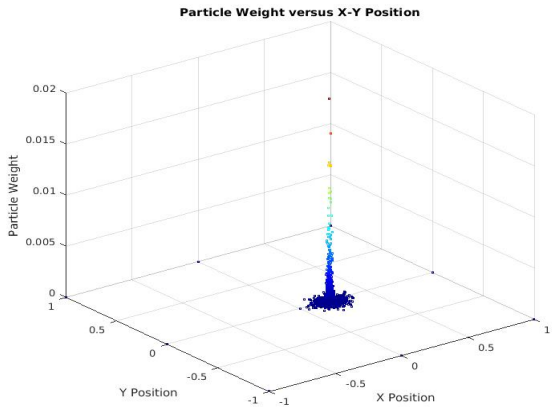

Fig. 5. The distribution of particles and their weights with respect to the X,Y position of the particles.

When running with a multi-tag configuration, the tracker is able to make use of the additional information to more accurately track the camera's location than in a single-tag scenario. The main challenge for running the algorithm with multiple tags was not successfully tracking the tags—the algorithm was able to do that with half the particles of the single-tag scenario. Rather, the majority of the reprojection error arises from error in the relative tag positions. This opens up new potential avenues of research, such as tag-based bundle adjustment, to alleviate these issues.

IV. Conclusion

We have presented the AprilTrack algorithm, which augments standard AprilTag detection algorithms with a tracker capable of tracking fiducials in 6 DOF. Experimental results show that the tracker can track both single tags and consolidating information from multiple tags in ego-motion and joint tag-motion scenarios. In future research we plan to investigate other possible error metrics for patch comparison and look to deploy AprilTrack in realtime scenarios.

References

[1] 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*, pages 113–120. IEEE, 2011.

[2] Y. Cho, J. Lee, and U. Neumann. A multi-ring color fiducial system and an intensity-invariant detection method for scalable fiducial-tracking augmented reality. In *In IWAR*. Citeseer, 1998.

[3] S. Houben, D. Droeschel, and S. Behnke. Joint 3D Laser and Visual Fiducial Marker based SLAM for a Micro Aerial Vehicle. 2016.

[4] E. Kraft. A quaternion-based unscented kalman filter for orientation tracking. In *Proceedings of the Sixth International Conference of Information Fusion*, volume 1, pages 47–54, 2003.

[5] E. Olson. AprilTag: A robust and flexible visual fiducial system. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011.

[6] E. Orhan. Particle Filtering, 2012.

[7] M. G. Prasad, S. Chandran, and M. S. Brown. A motion blur resilient fiducial for quadcopter imaging. *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, pages 254–261, 2015.

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

[9] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred target tracking by blur-driven tracker. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1100–1107, 2011.