# AprilTrack: A Particle-Filter based AprilTag Tracker

Daniel Pfrommer

*Abstract*— **The AprilTrack algorithm proposed in this paper seeks to bridge two existing fields of research in computer vision literature: fiducial detection and monocular object tracking. With a rise in the popularity of fiducials, or artificial markers, as a source of ground-truth position data for mobile robots, there has been a considerable interest in designing robust and blur-resistant fiducials. Rather than devising a new fiducial, This paper attempts to improve on existing fiducials such as the popular AprilTag marker by augmenting fiducial detection algorithms with a particle-filter based monocular object tracker which reliably tracks a fiducial's position and orientation in six degrees of freedom, allowing for tag-based localization in situations where still frame based methods would be unable to do so. This novel particle-filter based algorithm is evaluated on several datasets involving both ego-motion and tag-based-motion and is compared to standalone tag detection algorithms and ground-truth tag position data.**

## I. Introduction

Visual tracking of fiducial markers (or fiducials) plays an important role in many robotics and computer vision related fields, including augmented and virtual reality, visual odometry, and object tracking. Due to these scenarios taking place in tructured 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 quadrotor.

The majority of previous literature in fiducial detection and monocular object tracking has been focused single-frame fiducial detection. A common assumption in these algorithms is that the image being processed is blur free, expectation which breaks down in empirical situations. For instance, the widely-used AprilTag [cite] marker and its successor, the AprilTag 2 [cite], 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 discarding them as false positives as the identifying characteristics are completely blurred (for AprilTags this consists of a 6x6 binary grid in the center of the tag).

There has been considerable interest in designing blur resilient fiducials. Meghshyam et al. [cite] 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 itself pivoting around an axis). In addition, the increased resilience to blur comes at the cost of localization accuracy as only the center point of the tag can be reliably determined, whereas with other tags, such as the AprilTag, full pose estimation in 6 degrees of freedom (DOF) can be done based on just a single tag.

Unlike previous attempts to engineer more resilient tags, we propose to exploit the temporal coherence between successive frames in a video sequence. As many high-blur scenarios, such as a quadroter-mounted moving camera or tracking an AprilTag-labeled moving object, involve a continuous stream of images taken at equidistant time intervals, we propose using prior knowledge on a fiducial's 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.

Previous work involving the tracking of blurred objects includes the Blur-driven Tracker (BLUT) framework proposed by Wu et al. [cite] for tracking motion-blurred targets. BLUT, a particle-filter based tracker, works on the observation that blur strength can be used estimate an object's motion and provide valuable information for successfully tracking a blurred object. While capable of successfully tracking blurred objects, BLUT suffers from the disadvantage that it can do so only in two dimensions and so cannot track a sequence of motion involving arbitrary rotations and translations.

The novel method proposed in this paper is able to augment the AprilTag detector from Olson et al. [cite] with a particle-filter based tracker to successfully track an AprilTag through an arbitrary movement, providing valuable position information when existing methods would be unable to do so. In addition, the proposed algorithm is capable of incorporating the joint motion of an arbitrary number of tags to increase the accuracy of the tracked orientation.

**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

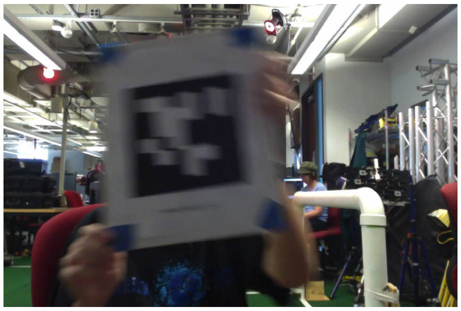

Fig. 1. A blurred AprilTag. Such tags are difficult for still frame based algorithms to detect as the payload (the ID encoded in the center of the tag) will cause the tag to be discarded as a false positive.

describes the implementation of the algorithm and evaluates its performance on a hand-labeled dataset of blurred moving tags with both individual tag motion and multi-tag ego motion. In sections 4 and 5 we discuss the limitations of this approach and suggest possible improvements for further research.

## II. Method

The AprilTrack algorithm is similar in concept to the BLUT tracker by Wu et al. [cite] 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 3 parts: one describing the particle filter, another on the method for evaluating candidate particles, and another describing the modifications made for tracking in multi-tag scenarios.

### A. The Particle Filter

The particle filter used for the AprilTrack algorithm is a Bayesian sequential importance resampling technique which approximates a posterior distribution of state variables. 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, $\dot{\boldsymbol{q}}$ 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 $\omega_x$, $\omega_y$, and $\omega_z$. We use $\boldsymbol{z}_k$ to denote the measurement (appearance) of the tag a particular time $t$.

The predicting distribution can be formulated in a Bayesian setting as the computation of the distribution $p(\boldsymbol{x}_t|\boldsymbol{z}_{t-1})$ from the prior distribution $p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})$:

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

Likewise, the update step can be formulated in a similar manner, where the prior distribution is updated with the latest measurement to obtain the posterior over $\boldsymbol{x}_t$:

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

In this regard, particle filters attempt to approximate the posterior distribution at time $t-1$ using Monte-Carlo methods. A particle filter involves a set of $N$ samples $\{(\boldsymbol{x}^i_{t-1}, w^i_{t-1})\}^N_{i=1}$ where each sample is called a *particle*. In the simplified case of a sequential importance resampling algorithm the prediction and update equations from 2 and 3 can written as:

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

$$w \propto p(\boldsymbol{z}_t|\boldsymbol{x}^i_t) \tag{5}$$

After selecting the particle with the highest weight as the most likely candidate, the final resampling step simply consists of drawing a new set of $N$ particles with equal weights (normalized to one) from the distribution $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ where $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ is approximated by the set of weighted particles.

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \approx \sum_{i=1}^{N} w^i_t \delta_{\boldsymbol{x}^i_t} \tag{6}$$

In AprilTrack, we model $p(\boldsymbol{x}_t|\boldsymbol{x}^i_{t-1})$ as a multivariate gaussian distribution with noise $\boldsymbol{\sigma}$ around

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{r}^i_{t-1} + \Delta t * \dot{\boldsymbol{r}}^i_{t-1} \\ \dot{\boldsymbol{r}}^i_{t-1} \\ \boldsymbol{q}^i_{t-1}\dot{\boldsymbol{q}}^i_{t-1} \\ \boldsymbol{\omega}^i_{t-1} \end{bmatrix} \tag{7}$$

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

### B. Evaluation of Candidate Particles

The successful tracking of AprilTags in 6 DOF requires a method for evaluating the likelihood of candidate particles at arbitrary three dimensional positions and rotations. 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

$$P(x_p, y_p|\boldsymbol{H}, I) = I(\boldsymbol{H} * \begin{bmatrix}x_p & y_p & 1\end{bmatrix}^\intercal) \tag{8}$$

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 particles 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, $P_{ref}(x_p, y_p|\boldsymbol{H}_{ref}, I)$. Here, $H^i$ can be calculated with

$$\boldsymbol{H}^i = \begin{bmatrix}\widehat{\boldsymbol{q}^i} & \boldsymbol{r}^i\end{bmatrix} \begin{bmatrix} \tau/2 & 0 & 0 \\ 0 & \tau/2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where $\widehat{\boldsymbol{q}}$ is the rotation matrix representation of particle's rotation quaternion, $\boldsymbol{r}$ is the particle's translation, and $\tau$ 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 [cite].

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.

$$\epsilon(\boldsymbol{A},\boldsymbol{B}) = \frac{1}{2} - \frac{\sum_{i=1}^{\rho}\sum_{j=1}^{\rho}(\boldsymbol{A}_{ij}-\mu_{\boldsymbol{A}})(\boldsymbol{B}_{ij}-\mu_{\boldsymbol{B}})}{2\sigma_A\sigma_B} \tag{10}$$

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

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = exp(-\gamma\epsilon(\boldsymbol{M}^i, \boldsymbol{M}_{ref})) \tag{11}$$

where the parameter $\gamma$ allows for more or less selective resampling.

### C. Multi-Tag Scenarios

To adapt this method to multi-tag scenarios, several extra parameters were 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

$$\boldsymbol{X}_c = \widehat{\boldsymbol{q}^i}(\widehat{\boldsymbol{g}_k}\boldsymbol{X}_t + \boldsymbol{x}_k) + \boldsymbol{r}^i \tag{12}$$

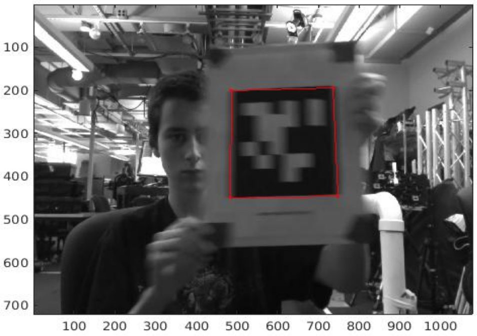

Fig. 2. A blurred tag being tracked. The red border indicates that the tag was not detected with the AprilTag detector.

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 9 can then be rewritten as

$$\boldsymbol{H}_k^i = \begin{bmatrix} \widehat{\boldsymbol{q}^i\boldsymbol{g}_{\boldsymbol{k}}} & \widehat{\boldsymbol{q}}\boldsymbol{x}_{\boldsymbol{k}} + \boldsymbol{r}^i \end{bmatrix} \begin{bmatrix} \tau_k/2 & 0 & 0 \\ 0 & \tau_k/2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

where $\boldsymbol{H}_k^i$ represents the homography for particle $i$ and tag $k$ and $\tau_k$ 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

$$\widehat{\boldsymbol{q}^i}\boldsymbol{X}_c + \boldsymbol{r}^i = \widehat{\boldsymbol{g}_k}\boldsymbol{X}_t + \boldsymbol{x}_k \tag{14}$$

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

$$\boldsymbol{H}_k^i = \begin{bmatrix} \widehat{\boldsymbol{q}^i}^{-1}\widehat{\boldsymbol{g}_k} & \widehat{\boldsymbol{q}^i}^{-1}(\boldsymbol{x}_k - \boldsymbol{r}^i) \end{bmatrix} \begin{bmatrix} \tau_k/2 & 0 & 0 \\ 0 & \tau_k/2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

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

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = exp(\sum_{k=1}^{K} -\gamma\epsilon(\boldsymbol{M}_k^i, \boldsymbol{M}_{k,ref})) \tag{16}$$

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 evluated 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.

[1]

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. $\tau$ 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 $\omega$.

When evaluated on the single-tag sequence, the algorithm ran with $N = 6000$ at around $0.5$ fps on an i7 Haswell Intel CPU. In experiments with fewer particles, it was found that

[1] Videos of the sequences and the tracker's performance are available online at https://www.youtube.com/watch?v=BmZLKFBkcxU, https://www.youtube.com/watch?v=ebWaxO7JhKE, and

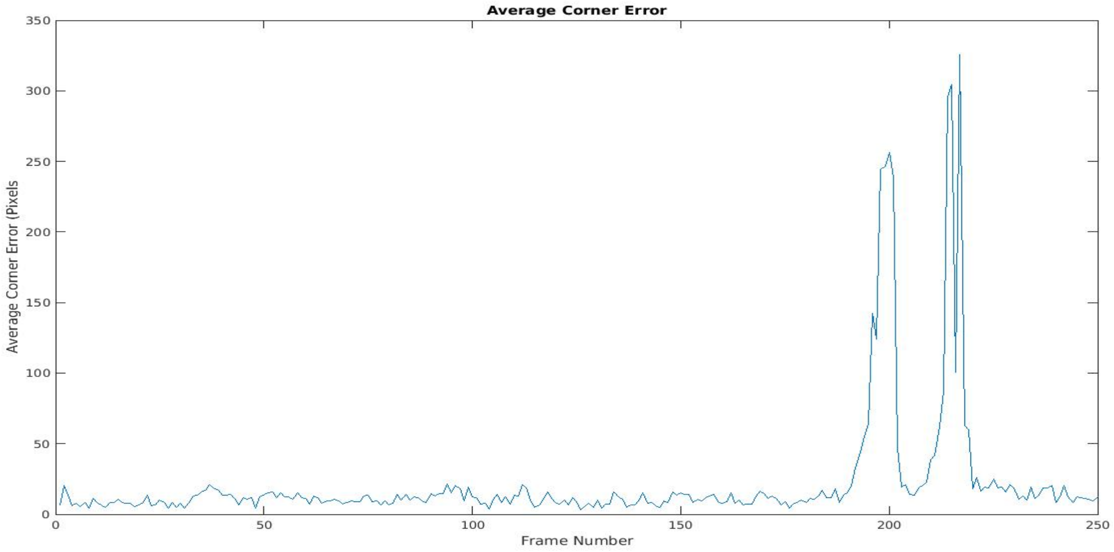

Fig. 3. The average corner error between the hand-marked tag location and the tracked location. Much of the error from frames 0-200 comes from discrepancy between the tracker and the ground truth on the boundaries of blurred tags. At around frame 200, the tracker loses the tag due to a sudden rotation but is able to recover without needing a hard reset.

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 likely be brought up to $10 - 15$ 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.

The pixel corner error hovers at around 10 pixels for most of the sequence. The tracker loses the tag momentarily during a quick twist at around frame 200 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.

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.

## IV. Conclusion

This paper has presented the AprilTrack algorithm augmenting the standard AprilTag detection algorithm with a novel tracker capable of tracking the tags in 6 DOF by using homographies to evaluate candidate particles. Experimental results show that the tracker is capable of accurately tracking 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.

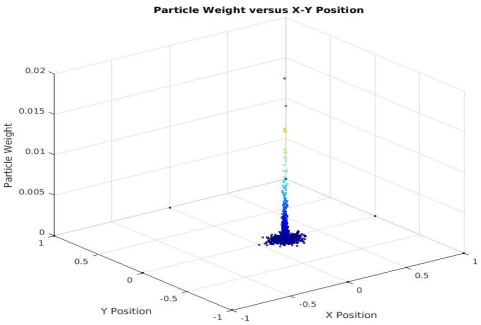

Fig. 4. The distribution of particles and their weights with respect to the X,Y position of the particles.

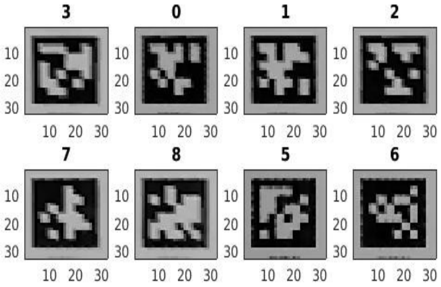

Fig. 5. The reference patches of various tags (by ID) in a multi-Tag tracking scenario.