Unmanned Systems, Vol. 0, No. 0 (2013) 1–11

Monocular Vision-based Localization and Pose Estimation with a Nudged Particle Filter and Ellipsoidal Confidence Tubes

Tony X. Lin[a], Samuel Coogan[a], Daniel M. Lofaro[c], Donald A. Sofge[c], Fumin Zhang[a]

[a] *School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA 30332, USA*
E-mail: {tlin339, sam.coogan, fumin}@gatech.edu

[c] *Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington D.C. 20375, USA*
E-mail: {daniel.lofaro, donald.sofge}@nrl.navy.mil

This paper proposes a nudged particle filter for estimating the pose of a camera mounted on flying robots collecting a video sequence. The nudged particle filter leverages two image-to-pose and pose-to-image neural networks trained in an auto-encoder fashion with a dataset of pose-labelled images. Given an image, the retrieved camera pose using the image-to-pose network serves as a special particle to nudge the set of particles generated from the particle filter while the pose-to-image network serves to compute the likelihoods of each particle. We demonstrate that such a nudging scheme effectively mitigates low likelihood samplings during the particle propagation step. Ellipsoidal confidence tubes are constructed from the set of particles to provide a computationally efficient bound on localization error. When an ellipsoidal tube self-intersects, the probability volume of the intersection can be significantly shrunken using a novel Dempster-Shafer probability mass assignment algorithm. Starting from the intersection, a loop closure procedure is developed to move backward in time to shrink the volumes of the entire ellipsoidal tube. Experimental results using the Georgia Tech Miniature Autonomous Blimp platform are provided to demonstrate the feasibility and effectiveness of the proposed algorithms in providing localization and pose estimation based on monocular vision.

Keywords: Monocular Vision-based Localization; Dempster-Shafer; Particle Filters.

1. Introduction

The use of monocular cameras as 3-D motion sensors has received extensive attention as a cheap and ubiquitous option for information-dense sensing in a variety of applications.[1–3] These sensors provide contextual information about the nearby environment that can directly relate to the completion of a given objective. In particular, localization objectives in consistent environments can highly benefit from monocular cameras as feature or image matching can be used to infer information about the global pose of the robot without externally mounted sensors (such as motion capture cameras, AprilTags, etc.). In addition, uncertainty quantification of the localization estimate is useful for assessing the risk associated with taking certain actions when safety-critical obstacles, like humans or expensive payloads, also exist in the environment.

In this paper, we also consider the ellipsoidal confidence tube estimation and refinement problem when provided a dataset of pose-labeled camera images before runtime and a previously observed landmark (also known as a loop closure). Ellipsoidal confidence tubes are sequences of ellipsoids such that the probability of an unknown parameter being inside the tube is, at minimum, a given parameter p. Ellipsoidal tubes have a variety of memory and computational benefits and are well-studied in other control and estimation tasks.[4–6] To help mitigate the descriptive deficiencies of ellipsoidal sets, we propose a computationally efficient Dempster-Shafer (D-S) based smoothing strategy using loop closure measurements to refine and tighten these sets. The refinement of these sets, when combined with a loop closure measurement, can allow for significant improvement in the estimation of the trajectory between sightings of a landmark. While other smoothing techniques are more accurate,[7–9] our approach is computationally more efficient as it involves only re-fitting a sequence of ellipsoids to sets of existing particles rather than re-tracing the trajectories of all previous particles.

The D-S theory of evidence is a framework for reasoning with probabilities over sets of events instead of individual events alone.[10] As such, the theory has been used for a wide variety of set-based estimation problems including human-activity recognition,[11] sensor fusion,[12,13] and fault detection.[14,15] In this work, we leverage D-S theory to define an optimization program for fusing multiple ellipsoids together. Our approach allows for the approximation of ellipsoid intersections using the particles and the D-S rule of combination to find a smoothed ellipsoidal confidence tube.

Provided a pose-labeled image dataset, other vision-based methods in the camera re-localization literature have also demonstrated successful camera pose inference[16,17] and have also explored using image-to-pose and pose-to-image networks for the purpose of localization.[18] However, the methods described in[16,17] do not provide robust uncertainty estimates. Alternatively, the strategy proposed by Rosenbaum[18] leverages a trained network to provide a localization estimate and an uncertainty density. The proposed strategy though requires training a network that has only been successfully trained in simulation environments. Other approaches consider finding the index of the most similar image in an image dataset from which a relative transformation using extracted features can be computed[19–21] or leverage a descriptive set of known features in the environment which are extracted beforehand.[22,23] Both approaches can be computationally expensive and inaccurate and provide no uncertainty estimates on the quality of the retrieved estimate. Our approach leverages networks trained using supervised learning to design a particle filter which provides uncertainty estimates both through the estimated distribution and the fitting of confidence tubes.

Particle filters are powerful filtering methods capable of handling nonlinear system dynamics and nonlinear observations. As such, these filters have been extensively used in the literature to solve a wide variety of estimation problems including SLAM-based state estimation, biological hidden state tracking, and global optimization.[24–26] These filters estimate a posterior distribution by propagating particles representing a prior distribution forward with a transition model that represents the possibly nonlinear dynamics of the system. These particles are then re-sampled using a likelihood function that describes how probable a given observation was generated by a particle.

When applied to high-dimensional state vectors, nudging strategies are used to improve the filter's sampling performance by optimizing particles with respect to the likelihood before re-sampling. This approach has been widely used in the geophysics literature, in which particle filters are used to model weather patterns which depend on high-dimensional unknown internal states.[27,28] However, computing the likelihood gradient may be difficult when the unknown state vector also incorporates a rotation. As such, we propose a new nudging strategy in which a new sample which represents a maximum likelihood estimate is included before re-sampling. The inclusion of the nudging particle can therefore be considered a form of robust particle propagation, allowing for the sampling of higher likelihood poses even when the particle propagation model is incorrect. This also allows the particle filter to recover from scenarios in which the initial distribution is incorrect. In the literature this problem is solved by inducing a high variance initial distribution and using a large number of particles.[29] If the initial distribution has low variance though, then tracking the target may fail. In this work we show through experiments that the nudging particle allows us to continue tracking the target even when the initial distribution is incorrect and with low variance.

Our contributions are as follows: **i)** we propose a vision-based likelihood function that allows for the application of the particle filter for vision-based localization, **ii)** we design a novel Dempster-Shafer based volume approximation for the intersection of ellipsoids, and **iii)** we utilize the intersection volume approximation to support the computation of fused ellipsoids that can be used to refine an existing ellipsoidal confidence tube given a loop closure measurement.

The structure of this paper is as follows. Section 2 discusses the posterior approximation and loop closure problems that are studied in this work. Sections 3 and 4 provide details on the proposed solutions to these problems using the particle filter and the D-S based volume approximation for ellipsoids. We conclude with experimental results validating our proposed solution in Section 5 and some final comments in Section 6.

2. Problem Formulation

Let the pose of a camera-mounted robot at time τ be described by

$$\boldsymbol{x}_\tau = (\boldsymbol{t}_\tau, \boldsymbol{R}_\tau) \tag{1}$$

in which $\boldsymbol{t}_\tau \in \mathbb{R}^3$ is the real-valued position and $\boldsymbol{R}_\tau \in \mathbb{R}^{3\times 3}$ is a rotation matrix. Poses are related by relative transformations $(\boldsymbol{u}_\tau, \boldsymbol{V}_\tau)$ such that

$$\begin{aligned} \boldsymbol{t}_\tau &= \boldsymbol{t}_{\tau-1} + \boldsymbol{R}_{\tau-1}\boldsymbol{u}_\tau \\ \boldsymbol{R}_\tau &= \boldsymbol{R}_{\tau-1}\boldsymbol{V}_\tau. \end{aligned} \tag{2}$$

From two sequential images $I_{\tau-1}, I_\tau$, we assume some image processing pipeline is able to provide an initial estimate of the relative transformation, denoted $(\hat{\boldsymbol{u}}_\tau, \hat{\boldsymbol{V}}_\tau)$ that is perturbed by noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma_t})$ and $\boldsymbol{\psi} \sim \boldsymbol{U}[-a, a]^3$, $a \in [0, \pi]$. Note that $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ and $\boldsymbol{U}[x, y]$ denotes the Uniform distribution with lower bound x and upper bound y. The relative transformation is therefore given by

$$\begin{aligned} \hat{\boldsymbol{u}}_\tau &= \boldsymbol{u}_\tau + \boldsymbol{\epsilon} \\ \hat{\boldsymbol{V}}_\tau &= f_{\boldsymbol{R}}(\boldsymbol{\psi})\boldsymbol{V}_\tau \end{aligned} \tag{3}$$

where $f_{\boldsymbol{R}}$ converts the sampled Euler angle to a rotation matrix. As stated previously, we assume our agent also has access to prior information about what images in the environment will look like. Let $\mathcal{D} = \{(I_i, \boldsymbol{x}_i)\}_{i=1}^M$ be a dataset of pose-labelled images that are provided beforehand. Using this dataset, we assume that a view prediction CNN denoted f_{image} can be trained that predicts the camera view given an input pose. Given a true image I and a pose $\boldsymbol{x}$, we choose the likelihood of a pose for the particle filter as

$$L(\boldsymbol{x}; I) = \Big[\sum_{x=1}^{W} \sum_{y=1}^{H} \| I_{x,y} - f_{image}(\boldsymbol{x})_{x,y} \| \Big]^{-\gamma} \tag{4}$$

where $\gamma \geq 1$ is a fixed scalar and subscript x, y indicates the 3-channel pixel at location x, y. This likelihood function captures the similarity between two images by comparing the pixel-wise intensity. While small changes in camera poses may lead to pixel intensity coordinate changes, images with areas of consistent intensity are likely to be easily described as similar by this function. The choice of γ here depends on the pixel intensities inherent in the environment as brighter pixels will yield larger likelihood differences (which may warrant a lower γ) while and dimmer pixels will yield lower likelihood differences (which may warrant a higher γ). Leveraging (4), we may now define the particle filtering problem[7] for the specific case when our measurements arrive as images.

Problem 2.1. *Assume a sequence of RGB images $I_1, I_2, \ldots, I_T$ and the starting pose of the camera $\boldsymbol{x}_1$ is provided from which a sequence of transformation estimates $(\hat{\boldsymbol{u}}_\tau, \hat{\boldsymbol{V}}_\tau)$ for $\tau = 1, 2, \ldots, T$ can be estimated. Approximate the posterior distribution $f(\boldsymbol{x}_\tau \mid I_1, I_2, \ldots, I_\tau)$ for $\tau = 1, 2, \ldots, T$ given the likelihood function (4).*

Problem 2.1 is a widely studied problem in the robotics literature for unknown parameter estimation in hidden Markov models (HMMs) and is particularly useful for describing localization problems. Often, the solution to Problem 2.1 has no analytical form or is difficult to compute, motivating the use of an approximation strategy.[7] The particle filter serves to solve this problem by approximating the posterior as a set of samples. For our case of camera pose localization, we leverage the particle filter to approximate the posterior distribution when the hidden state is the camera pose associated with a given image.

Let $\Xi_\tau = \{\xi_{\tau,i}\}_{i=1}^N$ be the set of particles $\xi_{\tau,i} = (w_{\tau,i}, \hat{\boldsymbol{x}}_{\tau,i})$ each with weight $w_{\tau,i}$ and poses $\hat{\boldsymbol{x}}_{\tau,i} = (\hat{\boldsymbol{t}}_{\tau,i}, \hat{\boldsymbol{R}}_{\tau,i})$. At time τ, Ξ_τ defines the probability density function approximation

$$f(\boldsymbol{z}; \Xi_\tau) = \sum_{i=1}^{N} w_{\tau,i} \delta(\boldsymbol{z} - \hat{\boldsymbol{t}}_{\tau,i}). \tag{5}$$

in which $\sum_{i=1}^N w_{\tau,i} = 1$ and δ is the dirac delta function. Given a sequence of particles, a secondary problem of interest in the literature is identifying and computing confidence tubes in which membership is ensured with a desired probability p. These tubes are heavily studied for stochastic control and estimation problems.[4,5]

Such tubes can be computed from particle points using the ellipsoidal peeling algorithm,[30] in which a minimum volume ellipsoid is fitted to a subset of data points. These tubes can also be refined when provided with a loop closure measurement or a detection that the camera or robot has returned to a previously visited location. Loop closures, in the context of SLAM, are used to correct odometry drift during the forward dynamics integration and refine trajectory estimates.

For a sequence of particles Ξ_τ, an ellipsoidal confidence tube is defined as a sequence of ellipsoids $\mathcal{E}_\tau = \mathcal{E}(\boldsymbol{A}_\tau, \boldsymbol{b}_\tau)$ in which $(\boldsymbol{A}_\tau, \boldsymbol{b}_\tau)$ are the shape parameters of an ellipsoid such that an ellipsoid defines a set

$$\mathcal{E}_\tau = \{\boldsymbol{z} \in \mathbb{R}^3 : \|\boldsymbol{A}_\tau \boldsymbol{z} + \boldsymbol{b}_\tau\| \leq 1\} \tag{6}$$

with total associated weight greater than a user-given value p. Letting $\mathcal{N}_\tau = \{i : \|\boldsymbol{A}_\tau \hat{\boldsymbol{t}}_{\tau,i} + \boldsymbol{b}_\tau\| \leq 1\}$ be the set of particle indices that lie inside $\mathcal{E}_\tau$, this condition expresses

$$\mathbb{P}(\boldsymbol{t}_\tau \in \mathcal{E}_\tau) = \sum_{i \in \mathcal{N}_\tau} w_{\tau,i} \geq p. \tag{7}$$

A loop closure measurement is defined as

$$\boldsymbol{y}_T = \boldsymbol{t}_T - \boldsymbol{t}_1 \tag{8}$$

which may be inferred through a variety of sensing strategies such as vision-based landmarks or RF signals. We may also detect a loop closure when the ellipsoidal set associated with the current time-step intersects an ellipsoidal set from a previous time-step. These loop closures allow for a refinement of the posterior distribution at the current time-step by treating the loop closure as a relative pose measurement between the current pose and a pose at a previous time-step. As such, the fusion of two such posterior distributions linked by a loop closure allows for a refinement of the information at the current time-step. In addition, this refinement can be used to refine all previous posterior estimates up to the previously identified pose used in the loop closure. This problem is known as smoothing and can greatly improve the trajectory estimation error over the given time interval. Using (6) and (8), the smoothing problem is defined as

Problem 2.2. *Given a confidence tube described by a sequence of ellipsoids $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_T$, their associated particles $\Xi_1, \Xi_2, \ldots, \Xi_T$, and a loop closure measurement $\boldsymbol{y}_T$, find the sequence of smoothed ellipsoids $\mathcal{E}'_1, \mathcal{E}'_2, \ldots, \mathcal{E}'_T$ such that $Vol(\mathcal{E}'_\tau) = -\log(\det((\boldsymbol{A}'_\tau)^{-1}))$ is minimized for all $\tau = 1, 2, \ldots, T$.*

3. View Prediction Particle Filter

In this section we provide details on the vision-based particle filter to estimate the solution to Problem 2.1.

3.1. *Particle Filtering*

Given a sequence of images $I_1, I_2, \ldots, I_T$ and transformation estimates $(\hat{\boldsymbol{u}}_\tau, \hat{\boldsymbol{V}}_\tau)$, we estimate the posterior distribution as a set of particles Ξ_τ. The relative transformation estimates may be estimated directly from a sensing strategy such as VO[31] or from a vehicle motion model such as a unicycle model or the miniature autonomous blimp used in this work.[32,33] Each propagated particle $\xi^+_{\tau,j} = (\hat{\boldsymbol{t}}^+_{\tau,j}, \hat{\boldsymbol{R}}^+_{\tau,j})$ is propagated using these transformations by

$$\begin{aligned} \hat{\boldsymbol{t}}^+_{\tau,j} &= \hat{\boldsymbol{t}}_{\tau,j} + \hat{\boldsymbol{R}}_{\tau,j} \hat{\boldsymbol{u}}_\tau + \boldsymbol{\epsilon}_j \\ \hat{\boldsymbol{R}}^+_{\tau,j} &= f_{\boldsymbol{R}}(\psi_j) \hat{\boldsymbol{R}}_{\tau,j} \hat{\boldsymbol{V}}_\tau \end{aligned} \tag{9}$$

where $\boldsymbol{\epsilon}_j$ and $\boldsymbol{\psi}_j$ are sampled from distributions $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma_t})$ and $\boldsymbol{U}[-a, a]^3$. Propagated particles are then re-sampled according to the likelihood via the weight

$$w_{\tau,j} = \frac{L(\hat{\boldsymbol{x}}^+_{\tau,j}; I_\tau)}{\sum_{i=1}^N L(\hat{\boldsymbol{x}}^+_{\tau,i}; I_\tau)} \tag{10}$$

to estimate the posterior $\Xi_{\tau+1}$. This sequence of propagation and re-sampling thereby allows the filter to track camera poses that best matches candidate poses propagated by the relative transformation estimates up to the quality of the estimate generated by a learned neural network f_{image}. From our experience, we found that the best image prediction performance was achieved when the input image was first downsampled and then blurred. While maintaining higher quality images would likely improve performance, higher quality image prediction appears to require a larger network than we can currently support. As such, with our currently designed structure we chose to downsample and blur the input images.

Samples of the network's prediction performance are shown in Fig. 1 in which the top row shows the original $360 \times 360 \times 3$ RGB image, the middle row shows the down-sampled $32 \times 32 \times 3$ blurred RGB image, and the bottom row shows the predicted results from the trained f_{image}.

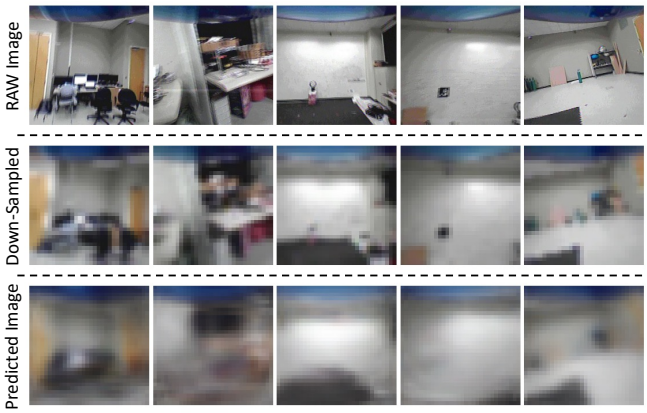

Fig. 1. Training examples and results generated from the trained view prediction network. **Top**: Raw images collected using the on-board FPV camera, **Middle**: Down-sampled and blurred images used for training, and **Bottom**: Predicted images from the neural network.

The network is able to predict general shapes of the environment in the predicted images, for example as in the large black mat on the bottom of the room and the yellow doorways. However, fine details such as a pink box in image 3 and a fiduciary tag in image 4 (from the left) is ignored entirely by the view prediction network. This indicates that the proposed View Prediction Particle Filter (VPPF) will likely be constrained to large obvious shapes in the environment. The network structure used in f_{image} is shown in Table 1. This network structure was experimentally found over various training iterations. To both speed up run-time evaluation of (4) and training time of f_{image}, received images are downsampled to $32\times32\times3$. ReLU activation layers are used between each deconvolutional layer and the final output is passed through a sigmoid layer to generate a valid image. Using dataset $\mathcal{D}$, the network is trained in PyTorch with the Adam optimizer and a step-size of 3e−5.

Table 1. Network structure of f_{image}

type	depth	output size
fully connected	1	1×256
reshape	2	1×1×256
deconvolutional	3	6×6×216
deconvolutional	4	10×10×192
deconvolutional	5	14×14×184
deconvolutional	1-5	14×14×184
deconvolutional	6	17×17×168
deconvolutional	7	20×20×154
deconvolutional	5-7	20×20×154
deconvolutional	8	23×23×148
deconvolutional	9	25×25×136
deconvolutional	10	27×27×128
deconvolutional	7-10	27×27×128
deconvolutional	11	29×29×108
deconvolutional	12	30×30×96
deconvolutional	10-12	30×30×96
deconvolutional	13	31×31×84
deconvolutional	14	32×32×72
deconvolutional	15	32×32×64
deconvolutional	16	32×32×54
deconvolutional	12-16	32×32×54
deconvolutional	17	32×32×36
deconvolutional	18	32×32×3

Remark 3.1. The construction and training of f_{image} using the dataset $\mathcal{D}$ necessitates a specific environment and camera per $\mathcal{D}$ and must be re-trained if images are generated using a new camera or environment. Compared to other methods in the camera re-localization literature though, our direct image inference allows us to avoid performing an image index search or feature point matching which may be costly or inaccurate. In addition, our approach can be extended to larger domains by considering a mixture-of-experts and using a separate dataset for other areas, e.g., new rooms. Letting f^i_{image} be the i^{th} view prediction network trained from the i^{th} dataset $\mathcal{D}^i$, with $i = 1, 2, \ldots, K$ (4) can be extended to

$$L(\boldsymbol{x}; I) = \Big[\sum_{i=1}^{K}\sum_{x=1}^{W}\sum_{y=1}^{H} \|I_{x,y} - f^i_{image}(\boldsymbol{x})_{x,y}\|\Big]^{-\gamma}. \tag{11}$$

As shown in our previous work and other results in the literature,[16,17] our dataset $\mathcal{D}$ also allows for the training of an image-to-pose CNN we denote f_{pose} that directly infers camera poses from images. We now describe our proposed

strategy to integrating f_{pose} into the VPPF as a nudging particle in order to improve the tracking accuracy.

3.2. *Nudged Particle Filtering*

While existing nudging strategies depend on optimizing the particle points Ξ_τ, we propose to infer the maximum likelihood pose using a CNN trained network denoted f_{pose} to act as our nudging strategy. While we may treat the pose estimate $\tilde{\boldsymbol{x}}_\tau$ provided by f_{pose} as another sensor and re-sample according to some distance between poses, the trained f_{pose} can yield erroneous measurements that might worsen the filter performance. This effect is evidenced in Fig. 2 in which the worst tracking performance is seen in spikes of position tracking error as high as $\sim 1.38m$. In addition, the quaternion tracking error is also prone to inaccuracies. The structure and training of this network is borrowed from our previous work in.[17]

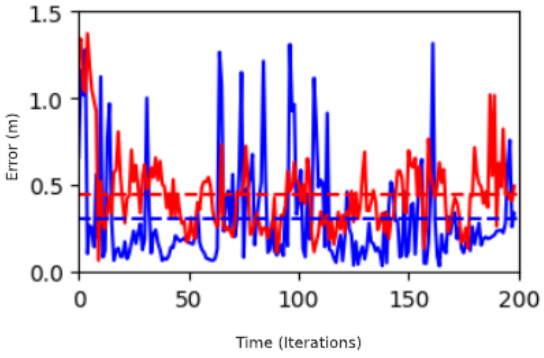

Fig. 2. Trained f_{pose} estimation error over a testing dataset sample. Over 200 timesteps, f_{pose} achieved an average position tracking error of $\sim 0.48m$ (red) and an average quaternion tracking error of $\sim 16.04^\circ$ (blue).

Instead, we leverage $\tilde{\boldsymbol{x}}_\tau$ as an additional propagated particle by adding the pose to the particle set after applying our relative transformation estimates and before re-sampling. If the estimate provided by f_{pose} yields a low likelihood as from (4), then the estimate is naturally ignored. However, if the estimate provides a high likelihood, then the particle filter is "nudged" closer to high likelihood regions during the re-sampling step. This quality allows for the nudged VPPF to be robust to possible errors in the relative transformation estimation, as poor estimates may lead to insufficient samples near the true pose, and to possible errors in f_{pose}. The full nudged particle filter is detailed in Algorithm 3.2.

Algorithm 3.2 (View Prediction Particle Filter).

$I_\tau \leftarrow$ grab_new_image()
$\hat{\boldsymbol{u}}_\tau, \hat{\boldsymbol{V}}_\tau \leftarrow$ get_relative_transform(I_τ)
for $\xi_{\tau,j} \in \Xi_\tau$ **do**
  $\hat{\boldsymbol{t}}^+_{\tau,j}, \hat{\boldsymbol{R}}^+_{\tau,j} \leftarrow$ propagate$(\hat{\boldsymbol{u}}_\tau, \hat{\boldsymbol{V}}_\tau, \boldsymbol{\epsilon}_j, \boldsymbol{\psi}_j)$ ▷ As in (9)
end for
$\tilde{\boldsymbol{x}}_\tau \leftarrow f_{pose}(I_\tau)$ ▷ Get max. likelihood estimate
$\Xi^+_\tau \leftarrow$ add_particle$(\Xi_\tau, \tilde{\boldsymbol{x}}_\tau)$ ▷ Add nudging particle
for $\xi_{\tau,j} \in \Xi^+_\tau$ **do**
  $L_j \leftarrow L(\hat{\boldsymbol{x}}^+_{\tau,j}; I_\tau)$ ▷ Compute likelihoods using (4)
end for
$\Xi_{\tau+1} \leftarrow$ resample$(\Xi^+_\tau, \{L_j\}_{j=1}^N)$ ▷ Return new particles

4. Loop Closure Smoothing

In this section, we describe our proposed approach to solving Problem 2.2. We first describe a novel approach to computing the volume of intersection using the D-S Theory of evidence and the particle filter, then describe an algorithm for finding a minimum volume fused ellipsoid that covers the volume of intersection.

4.1. *Dempster-Shafer Ellipsoidal Intersections*

Using (8), $\mathcal{E}_1$ can be translated to $\bar{\mathcal{E}}_T = \mathcal{E}(\boldsymbol{A}_1, \boldsymbol{b}_1 + \boldsymbol{A}_1\boldsymbol{y}_T)$ with associated particles $\bar{\xi}_{\tau,i} = (w_{1,i}, (\hat{\boldsymbol{t}}_{1,i} + \boldsymbol{y}_T, \hat{\boldsymbol{R}}_{1,i}))$. To fuse ellipsoids $\mathcal{E}_T$ and $\bar{\mathcal{E}}_T$ and the associated particle sets Ξ_T and $\bar{\Xi}_T$, we leverage the D-S rule of combination to compute the volume of ellipsoidal intersection and use the volume computations to define a new mass assignment function m'_T.

D-S theory allows for the combination of probability masses that have been allocated to sets of events instead of individual events, as usually seen in Bayesian formulations of probability. We define two mass assignment functions m_T and $\bar{m}_T$ for the four sets of being inside or outside $\mathcal{E}_T$ and being inside or outside $\bar{\mathcal{E}}_T$ using the particle memberships for each ellipsoid,

$$m_T(A) = \begin{cases} \sum_{i \in \mathcal{N}_T} w_{T,i}, & \text{if } A = \mathcal{E}_T \\ 1 - \sum_{i \in \mathcal{N}_T} w_{T,i}, & \text{if } A = \mathcal{E}_T^C \end{cases}$$
$$\bar{m}_T(A) = \begin{cases} \sum_{i \in \bar{\mathcal{N}}_T} w_{1,i}, & \text{if } A = \bar{\mathcal{E}}_T \\ 1 - \sum_{i \in \bar{\mathcal{N}}_T} w_{1,i}, & \text{if } A = \bar{\mathcal{E}}_T^C \end{cases} \tag{12}$$

in which $\mathcal{E}_T^C$ and $\bar{\mathcal{E}}_T^C$ refer to the set complements of $\mathcal{E}_T$ and $\bar{\mathcal{E}}_T$. The combination of m_T and $\bar{m}_T$ is then achieved using the D-S rule of combination.[10,34] For two given mass assignments m_1 and m_2, the combined mass assignment $m_{1,2}$ is

$$m_{1,2}(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} m_1(B) m_2(C)$$
$$K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C). \tag{13}$$

where K is a normalization factor based on any mass allocated to any events that do not intersect between m_1 and m_2.

Applying (13) to the ellipsoidal intersections, we find the new mass assignment $\tilde{m}_T$ for four possible events de-

pending on the intersections of the two ellipsoids

$$\tilde{m}_T(A) = \begin{cases} m_T(\mathcal{E}_T)\bar{m}_T(\bar{\mathcal{E}}_T^C), & \text{if } A = \mathcal{E}_T \cap \bar{\mathcal{E}}_T^C \\ m_T(\mathcal{E}_T^C)\bar{m}_T(\bar{\mathcal{E}}_T), & \text{if } A = \mathcal{E}_T^C \cap \bar{\mathcal{E}}_T \\ m_T(\mathcal{E}_T^C)\bar{m}_T(\bar{\mathcal{E}}_T^C), & \text{if } A = \mathcal{E}_T^C \cap \bar{\mathcal{E}}_T^C \\ m_T(\mathcal{E}_T)\bar{m}_T(\bar{\mathcal{E}}_T), & \text{if } A = \mathcal{E}_T \cap \bar{\mathcal{E}}_T \end{cases} \tag{14}$$

which allocates the mass associated with each event as the product between the two events. We now define a new mass assignment m'_T composed from $\tilde{m}_T$ and a given ellipsoid $\mathcal{E}'_T = \mathcal{E}(\boldsymbol{A}'_T, \boldsymbol{b}'_T)$ using the intersecting particles of Ξ_T and $\bar{\Xi}_T$.

The assignment m'_T is given by

$$\begin{aligned} m'_T(\mathcal{E}'_T) &= \sum_{A \subseteq \mathcal{E}'_T} \tilde{m}_T(A) \frac{\text{vol}(\mathcal{E}'_T \cap A)}{\text{vol}(A)} \\ &= \tilde{m}_T(\mathcal{E}_T \cap \bar{\mathcal{E}}_T^C) \frac{\text{vol}(\mathcal{E}'_T \cap \mathcal{E}_T \cap \bar{\mathcal{E}}_T^C)}{\text{vol}(\mathcal{E}_T \cap \bar{\mathcal{E}}_T^C)} \\ &+ \tilde{m}_T(\mathcal{E}_T^C \cap \bar{\mathcal{E}}_T) \frac{\text{vol}(\mathcal{E}'_T \cap \mathcal{E}_T^C \cap \bar{\mathcal{E}}_T)}{\text{vol}(\mathcal{E}_T^C \cap \bar{\mathcal{E}}_T)} \\ &+ \tilde{m}_T(\mathcal{E}_T^C \cap \bar{\mathcal{E}}_T^C) \frac{\text{vol}(\mathcal{E}'_T \cap \mathcal{E}_T^C \cap \bar{\mathcal{E}}_T^C)}{\text{vol}(\mathcal{E}_T^C \cap \bar{\mathcal{E}}_T^C)} \\ &+ \tilde{m}_T(\mathcal{E}_T \cap \bar{\mathcal{E}}_T) \frac{\text{vol}(\mathcal{E}'_T \cap \mathcal{E}_T \cap \bar{\mathcal{E}}_T)}{\text{vol}(\mathcal{E}_T \cap \bar{\mathcal{E}}_T)} \end{aligned} \tag{15}$$

where the intersecting volume is given by

$$\text{vol}(A \cap B) = \sum_{i \in A \cap B} w_i \tag{16}$$

or the set of particle weights such that the particle is a member of both events. Our final fused ellipsoid $\mathcal{E}'_T$ is then the solution to an optimization problem

$$\begin{aligned} \min_{\boldsymbol{A}'_T, \boldsymbol{b}'_T} \quad & -\log(\det(\boldsymbol{A}'^{-1}_T)) \\ \text{s.t.} \quad & m'_T(\mathcal{E}(\boldsymbol{A}'_T, \boldsymbol{b}'_T)) \geq p \\ & \boldsymbol{A}'_T \succ 0. \end{aligned} \tag{17}$$

To solve (17), we use a modified form of the ellipsoidal peeling algorithm originally developed in.[30]

4.2. *Ellipsoidal Peeling*

The ellipsoidal peeling algorithm is used to solve a k-covering minimum volume ellipsoid (MVE) problem in which given a set of N points, the volume of an ellipsoid that covers $k < N$ points is minimized. While the method has no guarantee of finding the minimum volume ellipsoid, the algorithm generally yields good solutions in practice.[35] Problems of this form naturally appear in robust machine learning (cluster analysis) and robust optimization problems, in which an outlier-free dataset is sought.[35,36] For a set of points $\boldsymbol{x}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$, the k-covering MVE problem is

$$\begin{aligned} \min_{\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{\alpha}} \quad & -\log(\det(\boldsymbol{A}^{-1})) \\ \text{s.t.} \quad & \alpha_i \|\boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{b}_\tau\| \leq 1, \quad \forall i \in \{1, 2, \ldots, N\} \\ & \alpha_i \in \{0, 1\}, \quad \forall i \in \{1, 2, \ldots, N\} \\ & \|\boldsymbol{\alpha}\|_1 = k \\ & \boldsymbol{A} \succ 0 \end{aligned} \tag{18}$$

in which $\boldsymbol{A} \in \mathbb{R}^{n \times n}_{++}$ and $\boldsymbol{b} \in \mathbb{R}^n$ define the parameters for an ellipsoidal set of dimension n and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^\intercal$ is an integer-valued vector in which each α_i denotes the inclusion of the point $\boldsymbol{x}_i$.

While solving (18) is an open problem, finding a minimum volume ellipsoid that covers all points can be formulated as a convex program[37]

$$\begin{aligned} \min_{\boldsymbol{A}, \boldsymbol{b}} \quad & -\log(\det(\boldsymbol{A}^{-1})) \\ \text{s.t.} \quad & \|\boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{b}\| \leq 1, \quad \forall i \in \{1, 2, \ldots, N\} \\ & \boldsymbol{A} \succ 0. \end{aligned} \tag{19}$$

and a solution using (19) known as the ellipsoidal peeling algorithm. Leveraging the Gaussian likelihood

$$L(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp(-\frac{1}{2}((\boldsymbol{x} - \boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}))}{(2\pi)^{n/2} \det(\boldsymbol{\Sigma})^{1/2}} \tag{20}$$

the approach finds the minimum volume ellipsoid by iteratively fitting normal distribution parameters (μ, Σ) and removing minimum likelihood points. Then (19) is solved using the remaining points.

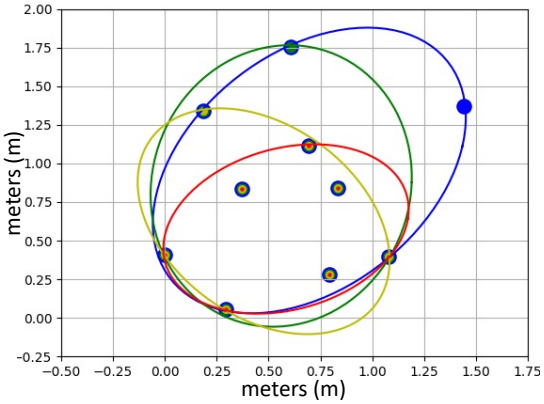

Fig. 3. Visualization of four iterations of the modified ellipsoidal peeling algorithm. Particle weights are higher for particles closer to the origin. Particle super-imposed colors indicate which colored sets each particle is a member of.

In this work we use a modified form of the ellipsoidal peeling algorithm shown in Algorithm 4.1 in which a weighted mean and covariance are used to find the minimum likelihood points.

Algorithm 4.1 (Modified Ellipsoidal Peeling).

1: $\boldsymbol{X} \leftarrow \{\boldsymbol{x}_i, i = 1, 2, \ldots, N\}$ ▷ Set of particle locations

$\boldsymbol{W} \leftarrow \{w_i, i = 1, 2, \ldots, N\}$ ▷ Set of particle weights
$p \leftarrow$ get_weight_criteria()
$found \leftarrow$ False
while $\neg found$ **do**
  $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow$ get_weighted_params$(\boldsymbol{X}, \boldsymbol{W})$
  $\boldsymbol{L} \leftarrow$ gaussian_likelihood$(\boldsymbol{X}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$
  $i_{min} = \arg\min_i L_i$ ▷ Find lowest likelihood
  $\boldsymbol{X}', \boldsymbol{W}' \leftarrow$ delete$(\boldsymbol{X}, \boldsymbol{W}, i_{min})$
  $p' \leftarrow \sum_{w \in \boldsymbol{W}'} w$
  if $p' \leq p$ **then** ▷ End search if total p is satisfied
    $found \leftarrow$ True
  else
    $\boldsymbol{X}, \boldsymbol{W} \leftarrow \boldsymbol{X}', \boldsymbol{W}'$ ▷ Update interior points
  end if
end while
$(\boldsymbol{A}, \boldsymbol{b}) \leftarrow$ solve$(\boldsymbol{X})$ ▷ Solve (19) with remaining $\boldsymbol{X}$

A visual example of the algorithm is also presented in Fig. 3 in which points closer to the origin have higher weights. As shown in the example, minimum volume ellipsoids are found such that points further from the origin are excluded.

To apply Algorithm 4.1 to the loop closure problem, we use both Ξ_T and $\bar{\Xi}_T$ with weights that have been re-computed using (15). Going backwards in time from $\tau = T$, we use the transformed ellipsoid $\bar{\mathcal{E}}_\tau = \mathcal{E}(\boldsymbol{A}'_{\tau+1}, \boldsymbol{b}'_{\tau+1} - \boldsymbol{A}'_{\tau+1}\boldsymbol{r}_\tau)$ as our measurement to solve for the ellipsoid $\mathcal{E}'_\tau$, for time-steps $\tau = 1, 2, \ldots, T-1$ in which $\boldsymbol{r}_\tau = \mathbb{E}[f(\boldsymbol{x} \mid I_1, I_2, \ldots, I_{\tau+1})] - \mathbb{E}[f(\boldsymbol{x} \mid I_1, I_2, \ldots, I_\tau)]$. Particles at each time-step are also re-sampled using the newly fitted ellipsoids. Letting $\mathcal{N}'_\tau$ be the set of particle indices from Ξ_τ and $\bar{\Xi}_\tau$ that lie inside of $\mathcal{E}'_\tau$ and p' be the total weight of particles inside $\mathcal{E}'_\tau$, re-sampling is performed such that the proportion of points inside and outside $\mathcal{E}'_\tau$ matches p', i.e., after re-sampling from Ξ_τ and $\bar{\Xi}_\tau$, $|\mathcal{N}'_\tau| / (|\mathcal{N}'_\tau| + |\mathcal{N}'^C_\tau|) = p'$.

An example using Algorithm 4.1 to solve Problem 2.2 is shown in Fig. 4 for three levels of confidence $p = 0.5, 0.7, 0.9$ on a simulated 2D particle filter with initial ellipses that have been found with $p = 0.7$. For low confidence $p = 0.5$, the refined tube is significantly tighter but frequently does not contain the true trajectory. For medium confidence $p = 0.7$, the refined tube bounds the true trajectory less tightly but with greater frequency. In the final case when $p = 0.9$, the estimated ellipses are extremely conservative but entirely contain the trajectory.

5. Experimental Results

In this section we validate the proposed monocular vision-based localization strategy using images collected from a lighter-than-air blimp, shown in Fig. 5. This platform, called the GT-MAB, was autonomously flown to collect $\sim 144.5k$ training images and $\sim 36k$ testing images. All experiments presented in this section are based on using a subset of data from the testing dataset. The pose-labeled dataset was also used in our previous work.[17]

Fig. 5. The GT-MAB research blimp used to collect the pose-labeled image dataset.

For our experiments, $\mathbb{E}\big[f(\boldsymbol{x} \mid I_1, I_2, \ldots, I_\tau\big]$ is used as the point estimate and we assume relative transformation noise parameters are $\boldsymbol{\Sigma}_t = 0.02\boldsymbol{I}_3$ where $\boldsymbol{I}_3$ is the identity matrix of size 3×3 and $a = \frac{\pi}{2}$. The likelihood parameter γ is chosen as 4 and the loop-closure measurement is made using a motion-capture system for these experiments. Initial ellipsoids are found using $p = 0.98$ while smoothed ellipsoids are found using $p = 0.7$. Fig. 6 shows the position and quaternion error using the non-nudged and nudged versions of the particle filter both before and after smoothing.

The inclusion of the nudging particle and smoothing operation provides significant position tracking improvement with the nudged filter yielding an improvement of $\sim$25% and the smoothing operation yielding an improvement of $\sim$33%. However, the smoothing operation appears to worsen the quaternion tracking performance. This is likely due to a few samples that are heavily weighed in the loop closure process belonging far from the mean of the particle filter. As a result, the smoothing operation potentially gives more weight to outlier particles which may have better position estimates but worse rotation estimates. While some rotation accuracy is lost, the filter is still extremely accurate after smoothing with a small increase of $\sim 5^\circ$ inaccuracy.

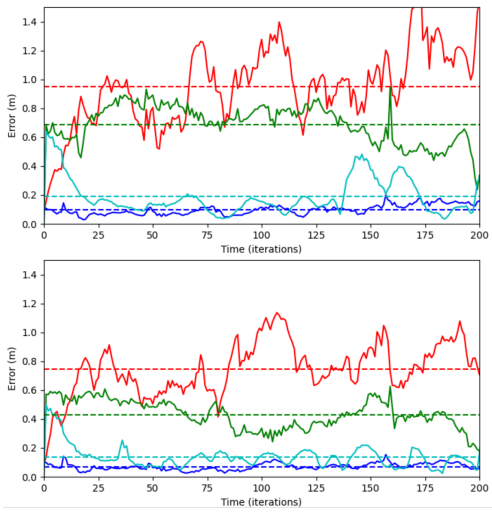

Fig. 6. Experimental tracking results, solid red indicates posi-

tion error (in meters), solid green indicates smoothed position error (in meters), solid blue indicates quaternion error (in radians), solid cyan indicates smoothed quaternion error (in radians). Dashed lines correspond to the average error of each color. **Top**: VPPF with an average position and quaternion error of $\sim 0.97m$ and $\sim 5.21^\circ$. After smoothing achieves an average position and quaternion error of $\sim 0.71m$ and $\sim 11.28^\circ$. **Bottom**: Nudged VPPF with an average position and quaternion error of $\sim 0.78m$ and $\sim 5.05^\circ$. After smoothing achieves an average position and quaternion error of $\sim 0.46m$ and $\sim 9.92^\circ$.

We consider another set of experiments here to demonstrate the benefit of a nudging particle when the initial estimated distribution is incorrect. Such a case might occur if a user gives the agent a confident but incorrect estimate of its initial pose. Normally, particle filters can handle these scenarios by starting with a highly varying initial distribution.[29] In the case where the initial distribution has low-variance though, it is difficult for the particle filter to re-converge on the correct pose. By leveraging the nudging particle though, we can correct the tracking error by pushing our particle distribution closer to the true pose.

5.1. *Wrong Initial Distribution*

Our proposed approach also allows for resiliency in the event the initial particle distribution in accurate. In normal cases, the view prediction particle filter will be unable to track the pose of the camera. However, using the nudged particle filter, we can re-identify the pose of the camera with ease despite having the wrong initialization. Fig. 7 shows the results of running the VPPF and the nudged VPPF when under incorrect initialization. In these experiments, the initial distribution is centered around the origin with a set of particles with positions normally distributed with covariance matrix $\mathbf{\Sigma}_0 = 0.02\boldsymbol{I}_3$ and with quaternions distributed using $a = 0.04$.

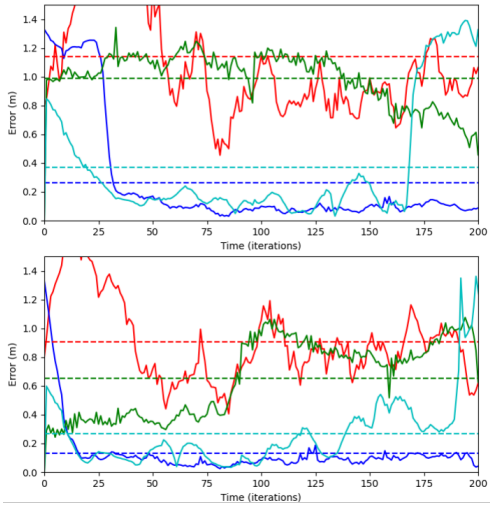

Fig. 7. Experimental tracking results, solid red indicates position error (in meters), solid green indicates smoothed position error (in meters), solid blue indicates quaternion error (in radians), solid cyan indicates smoothed quaternion error (in radians). Dashed lines correspond to the average error of each color. **Top**: Wrongly initialized VPPF with an average position and quaternion error of $\sim 1.18m$ and $\sim 14.9^\circ$. After smoothing achieves an average position and quaternion error of $\sim 0.99m$ and $\sim 21.77^\circ$. **Bottom**: Wrongly initialized nudged VPPF with an average position and quaternion error of $\sim 0.92m$ and $\sim 8.02^\circ$. After smoothing achieves an average position and quaternion error of $\sim 0.68m$ and $\sim 14.33^\circ$.

We see that the nudged VPPF recovers from the incorrect initial distribution and is able to recover the position with reasonable accuracy ($\sim 0.68m$) while the VPPF struggles to recover the correct position ($\sim 0.99m$).

5.2. *Comparison to Backwards Simulation*

We provide details here on the performance of our smoothing strategy when compared to that of the Backwards Sim-

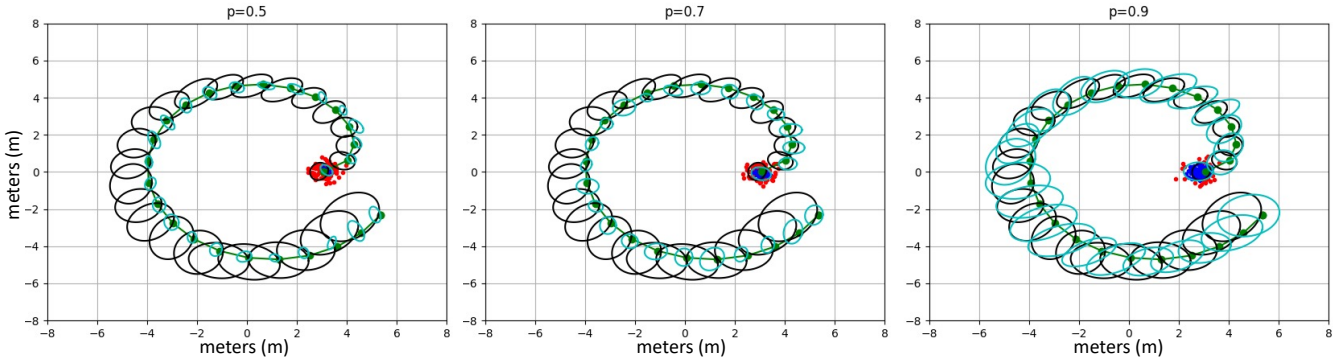

Fig. 4. The green trajectory indicates the true position of a simulated camera being tracked by the particle filter with initial ellipses shown in black. Newly found ellipses are shown in cyan with the final particle set shown with red points lying on the exterior and blue points lying on the interior of the final ellipse. Initial ellipsoids are found using $p = 0.7$.

ulation smoothing strategy as detailed in.[7] The Backwards Simulation strategy leverages a history of particle distributions and iterates backwards in time starting from the final state to find a set of smoothed particle distributions. Leveraging the motion model $p(\boldsymbol{x}_{t+1}, \boldsymbol{x})$, particle weights are re-weighed as the trajectory is simulated backwards until the initial time-step. The algorithm is then repeated for as many samples as desired. The full approach is briefly detailed in Algorithm 5.1. For this comparison study, N backward simulation samples are collected and the mean of the smoothed particles are used to compare the smoothing strategies.

Algorithm 5.1 (Backwards Simulation).

Initial particles: $\{w_t^{(i)}, \boldsymbol{x}_t^{(i)} : i = 1, \ldots, N, t = 1, \ldots, T\}$
for $i = 1, 2, \ldots, N$ **do**
  $\tilde{\boldsymbol{x}}_T^{(i)} \leftarrow$ sample_particle$(\{(w_t^{(i)}, \boldsymbol{x}_t^{(i)})\}_{i=1}^N)$
  for $k = T-1, T-2, \ldots, 0$ **do**
    $w_{k|k+1}^{(i)} \leftarrow w_k^{(i)} p(\tilde{\boldsymbol{x}}_{k+1} \mid \boldsymbol{x}_k^{(i)})$
    $\tilde{\boldsymbol{x}}_k^{(i)} \leftarrow$ sample_particle$(\{(w_{k|k+1}^{(i)}, \boldsymbol{x}_k^{(i)})\}_{i=1}^N)$
  end for
end for
Result: $\{w_{t|t+1}^{(i)}, \tilde{\boldsymbol{x}}_t^{(i)} : i = 1, \ldots, N, t = 1, \ldots, T\}$

Using the VPPF to generate the initial particles, the Backwards Simulation strategy is compared with two cases. In the first case, the Backwards Simulation strategy is implemented as in Algorithm 5.1 and directly compared with the Dempster-Shafer smoothing strategy. In the second case, the Backwards Simulation strategy is implemented after applying the loop closure method detailed in Section 4 and compared with Dempster-Shafer smoothing strategy with loop closure. As shown in Fig. 8, both strategies appear to generate very similar smoothed trajectory errors. However, the Dempster-Shafer smoothing strategy only requires a single pass backwards over the particle sets while the Backwards Simulation strategy requires N passes backwards.

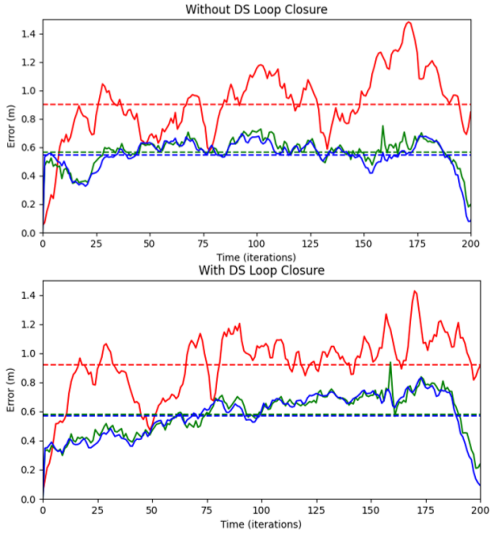

Fig. 8. Comparisons between the Dempster-Shafer based smoothing strategy and the Backwards Simulation smoothing strategy. Red indicates the initial particle trajectory error, blue indicates the smoothed trajectory error using the Dempster-Shafer strategy and red indicates the smoothed trajectory error using the Backwards Simulation strategy. Top shows the comparison without a Dempster-Shafer loop closure, bottom shows the comparison with a Dempster-Shafer loop closure.

6. Conclusions

In this work we proposed a vision-based localization approach based on using particle filters and deep-learning based image-to-pose and pose-to-image predictions. The strategy also allows for the construction and smoothing of ellipsoidal confidence tubes that contain the position of the camera with a given probability p. We note that compared to the Backwards Simulation strategy, our approach achieves comparable results but for less computational effort.

In our future work we plan to extend our proposed method to support loop closures on the rotation space of the trajectory as well. We will also consider an extension of our work to a multi-agent swarming scenario in which multiple agents using the VPPF can provide loop closure measurements to each other and can collaboratively improve their localization performance.

ACKNOWLEDGMENT

T. Lin and S. Coogan were partially supported by the NSF under grant 1924978. Zhang's research work is supported by ONR grants N00014-19-1-2556 and N00014-19-1-2266; AFOSR grant FA9550-19-1-0283; NSF grants CNS-1828678, S&AS-1849228 and GCR-1934836; and NOAA grant NA16NOS0120028.

References

[1] G. Fink, H. Xie, A. F. Lynch, and M. Jagersand, "Dynamic visual servoing for a quadrotor using a virtual camera," *Unmanned Systems*, vol. 5, no. 01, pp. 1–17, 2017.

[2] J. Su, M. Coombes, C. Liu, Y. Zhu, X. Song, S. Fang, L. Guo, and W.-H. Chen, "Machine learning-based crop drought mapping system by uav remote sensing rgb imagery," *Unmanned systems*, vol. 8, no. 01, pp. 71–83, 2020.

[3] Z.-H. Wang, K.-Y. Qin, T. Zhang, and B. Zhu, "An intelligent ground-air cooperative navigation framework based on visual-aided method in indoor environments," *Unmanned Systems*, vol. 9, no. 03, pp. 237–246, 2021.

[4] T. F. Filippova, "Ellipsoidal estimates of reachable sets for control systems with nonlinear terms," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 355–15 360, 2017.

[5] M. E. Villanueva, R. Quirynen, M. Diehl, B. Chachuat, and B. Houska, "Robust mpc via min–max differential inequalities," *Automatica*, vol. 77, pp. 311–321, 2017.

[6] L. Yan, T. Stouraitis, and S. Vijayakumar, "Decentralized ability-aware adaptive control for multi-robot collaborative manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2311–2318, 2021.

[7] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, no. 3.

[8] P. E. Jacob, F. Lindsten, and T. B. Schön, "Smoothing with couplings of conditional particle filters," *Journal of the American Statistical Association*, 2019.

[9] E. Ye, G. Würsching, S. Steyer, and M. Althoff, "Offline dynamic grid generation for automotive environment perception using temporal inference methods," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5501–5508, 2021.

[10] R. R. Yager, "Generalized dempster–shafer structures," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 428–435, 2018.

[11] R. Wen, M. A. Siddiqui, and T. Williams, "Dempster-shafer theoretic learning of indirect speech act comprehension norms," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 06, 2020, pp. 10 410–10 417.

[12] E. Koksalmis and Ö. Kabak, "Sensor fusion based on dempster-shafer theory of evidence using a large scale group decision making approach," *International Journal of Intelligent Systems*, vol. 35, no. 7, pp. 1126–1162, 2020.

[13] N. Nesa and I. Banerjee, "Iot-based sensor data fusion for occupancy sensing using dempster–shafer evidence theory for smart buildings," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1563–1570, 2017.

[14] K. H. Hui, M. H. Lim, M. S. Leong, and S. M. Al-Obaidi, "Dempster-shafer evidence theory for multi-bearing faults diagnosis," *Engineering Applications of Artificial Intelligence*, vol. 57, pp. 160–170, 2017.

[15] K. Verbert, R. Babuška, and B. De Schutter, "Bayesian and dempster–shafer reasoning for knowledge-based fault diagnosis–a comparative study," *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 136–150, 2017.

[16] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[17] L. Seguin, J. Zheng, A. Li, Q. Tao, and F. Zhang, "A deep learning approach to localization for navigation on a miniature autonomous blimp," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*. IEEE, 2020, pp. 1130–1136.

[18] D. Rosenbaum, F. Besse, F. Viola, D. J. Rezende, and S. Eslami, "Learning models for visual 3d localization with implicit mapping," *arXiv preprint arXiv:1807.03149*, 2018.

[19] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "Camnet: Coarse-to-fine retrieval for camera relocalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2871–2880.

[20] V. Balntas, S. Li, and V. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 751–767.

[21] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative

poses using convolutional neural network," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 929–938.

[22] E. Brachmann and C. Rother, "Expert sample consensus applied to camera re-localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7525–7534.

[23] ——, "Visual camera re-localization from rgb and rgb-d images using dsac," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[24] H. Jo, H. M. Cho, S. Jo, and E. Kim, "Efficient grid-based rao–blackwellized particle filter slam with inter-particle map sharing," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 714–724, 2018.

[25] P. Dawson, R. Gailis, and A. Meehan, "Detecting disease outbreaks using a combined bayesian network and particle filter approach," *Journal of theoretical biology*, vol. 370, pp. 171–183, 2015.

[26] Ö. D. Akyildiz, D. Crisan, and J. Míguez, "Parallel sequential monte carlo for stochastic gradient-free nonconvex optimization," *Statistics and Computing*, vol. 30, no. 6, pp. 1645–1663, 2020.

[27] Ö. D. Akyildiz and J. Míguez, "Nudging the particle filter," *Statistics and Computing*, vol. 30, no. 2, pp. 305–330, 2020.

[28] H. C. Yeong, R. T. Beeson, N. Namachchivaya, and N. Perkowski, "Particle filters with nudging in multi-scale chaotic systems: with application to the lorenz'96 atmospheric model," *Journal of Nonlinear Science*, vol. 30, no. 4, pp. 1519–1552, 2020.

[29] I. Bukhori, Z. Ismail, and T. Namerikawa, "Detection strategy for kidnapped robot problem in landmark-based map monte carlo localization," in *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*. IEEE, 2015, pp. 75–80.

[30] V. Barnett, "The ordering of multivariate data," *Journal of the Royal Statistical Society: Series A (General)*, vol. 139, no. 3, pp. 318–344, 1976.

[31] T. H. Nguyen, T.-M. Nguyen, M. Cao, and L. Xie, "Loosely-coupled ultra-wideband-aided scale correction for monocular visual odometry," *Unmanned Systems*, vol. 8, no. 02, pp. 179–190, 2020.

[32] Q. Li, J. P. Queralta, T. N. Gia, Z. Zou, and T. Westerlund, "Multi-sensor fusion for navigation and mapping in autonomous vehicles: Accurate localization in urban environments," *Unmanned Systems*, vol. 8, no. 03, pp. 229–237, 2020.

[33] Q. Tao, J. Wang, Z. Xu, T. X. Lin, Y. Yuan, and F. Zhang, "Swing-reducing flight control system for an underactuated indoor miniature autonomous blimp," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 1895–1904, 2021.

[34] G. Shafer, "Dempster-shafer theory," *Encyclopedia of artificial intelligence*, vol. 1, pp. 330–331, 1992.

[35] S. D. Ahipaşaoğlu, "Fast algorithms for the minimum volume estimator," *Journal of Global Optimization*, vol. 62, no. 2, pp. 351–370, 2015.

[36] L. Vandenberghe and S. Boyd, "Applications of semidefinite programming," *Applied Numerical Mathematics*, vol. 29, no. 3, pp. 283–299, 1999.

[37] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.