Monocular Vision-based Detection of a Flying Bird

Dezhen Song and Yiliang Xu

Abstract

To assist nature observation, we take on the challenge of detecting the species of a flying bird using a single camera. We study the bird flying data and find that a bird body axis is an invariant dimension during flight. We then develop a model-based detection approach that verifies the body axis information with the known bird flying dynamics. As a commonly used method, an extended Kalman filter (EKF) cannot be directly applied because the EKF would not converge due to the high measurement error introduced by image segmentation and the limited observation data due to the high flying speed of the bird. To cope with the problem, we develop a novel Probable Observation Data Set (PODS)-based EKF method. First, we prove that the EKF converges when there is no measurement error, and the new PODS-EKF searches the measurement error range for all probable observation data that ensures the convergence of the corresponding EKF. The detection is based on whether the set PODS is non-empty and the corresponding velocity is within the known bird flying velocity profile. The algorithm has been extensively tested using both simulated inputs and physical experiments. The results are satisfying and have shown the bird detector has less than 7% false negative rate and 90% area under the receiver operating characteristic (ROC) curve.

Index Terms

monocular vision, autonomous observatory, nature observation, bird detection

I. Introduction

Observing nature in harsh and inhospitable environments for a long period of time has been a major challenge for natural scientists. Our group focuses on developing autonomous robotic observatories to address this problem. As a recent development, a camera was installed in the middle of a forest to assist ornithologists to search and document bird activities as illustrated in Fig. 1. Due to power and communication constraints, it is often prohibitive to install multiple cameras at the same location. The challenging problem is how to automatically detect and identify the bird species using the monocular vision system under the noisy background and varying lighting conditions of the nature environment.

This work is supported in part by the National Science Foundation under IIS-0534848/0643298, in part by Arkansas Electric Cooperatives Corp., in part by U.S. Fish and Wildlife Service, and in part by Microsoft.

D. Song, and Y. Xu are with the Computer Science Department, Texas A&M University, College Station, TX 77843, USA (email: *dzsong@cs.tamu.edu*).

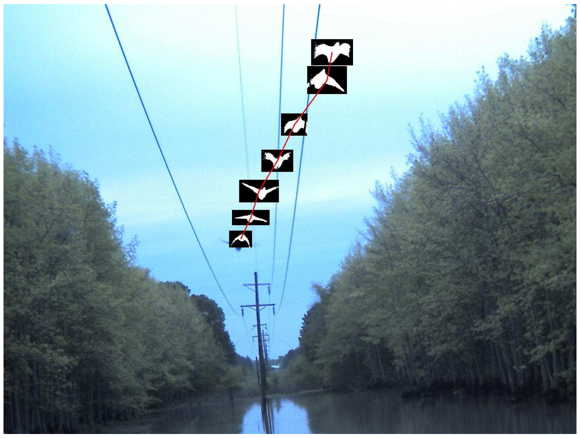

Fig. 1. An example of a video sequence of a flying bird that is captured in Bayou DeView in eastern Arkansas. This sequence is generated by superimposing the segmented bird images from consecutive video frames on the top of the background frame.

As illustrated in Fig. 1, the input of the problem is a segmented motion sequence of an object from consecutive video frames. The output of the problem is to determine whether the motion sequence is caused by a targeted bird species. To address the problem, we study the bird flying data and find that a bird body axis is an invariant dimension during flying. We also notice that the bird body axis is often parallel to that of the tangent line of the bird flying trajectory. Based on the observations, we develop a bird body axis filter that extracts the length and the orientation of the bird body axis.

We then develop a model-based detection approach that verifies the body axis information with the known bird flying dynamics. As a commonly used method, an extended Kalman filter (EKF) cannot be directly applied because the EKF would not converge due to the high measurement error introduced by image segmentation and the limited observation data due to the high flying speed of the bird. The sample bird sequence in Fig. 1 only contains seven data points. To cope with this problem, we develop a probable observation data set (PODS)-based EKF and an approximate computation scheme. Based on the fact that the EKF converges when there is no measurement error, the new PODS-EKF searches the measurement error range for all probable observation data that ensures the convergence of the corresponding EKF. The detection is based on whether the PODS is non-empty and the corresponding velocity is within the known bird flying velocity profile. We show that the PODS-EKF theoretically ensures a zero false negative rate. We have evaluated our biometric bird filtering algorithm using the simulated data set and field test data. In physical experiments, we report how our algorithm has been applied for the detection of rock pigeons on Texas A&M campus and for the search of the ivory-billed woodpeckers in eastern Arkansas. The physical experiment results shown a promising result of 90.0% area under the receiver operating characteristic (ROC) curve.

The rest of the paper is organized as follows. Section II reviews the related works. The definition of the bird filtering problem and necessary assumptions are presented in Section III. In Section IV and V, we model the bird filtering problem and propose the PODS-EKF method. The simulation on random input data and the experiments on real data are presented in Section VII before we conclude in Section VIII.

II. Related Work

Using vision to detect a flying bird relates to fields of remote nature cameras, vision-based motion detection, and Kalman filter-based visual tracking and recognition.

Remote electronic nature camera systems have existed since 1950s. Gysel and Davis [1] build an early video camera based on remote wildlife observation system to study rodents. Cameras have been deployed by biologists to observe feeding behavior, species presence, and population parameters [2]–[7]. Commercial remote camera systems such as Trialmaster [2] and DeerCam have been developed since 1986 and have been widely used in wildlife observation and hunting sports. The Internet enables webcam systems that allow the general public to access live robotic cameras. Thousands of webcams have been installed around the world, for example, to observe elephants [8], tigers [9], bugs [10], birds/squirrels [11], [12], and swans [13]. Many other examples can be found in [14]. However, existing remote cameras do not have the ability to autonomously recognize animal species and are just a mere recording/observation device. Our objective is to develop intelligent remote robotic observatories that are capable of automatically classifying and documenting activities with respect to the targeted animal species.

Recent development in vision-based motion detection has greatly advanced the robustness of the motion detection in noisy environments. Motion detection segments moving objects from their background in a video sequence. To address the background noise, researchers propose many statistics-based background models such as temporal average [15], median absolute deviation (MAD) [16], adaptive Gaussian estimation [17], mixed Gaussian model, parameter estimation [18], non-parameter estimation [19], and Kalman filter compensation [20]. Our work builds on the nonparametric background subtraction algorithm proposed in [**?**] to segment the moving foreground objects. Moreover, our algorithm advances the mere motion-detection to bird species detection by utilizing the motion information across multiple frames and the known bird flying dynamics.

The fundamental technique we used in the bird detection is the extended Kalman filer. Kalman filter, extended Kalman filter, and their variations can be viewed as model-based detection methods and are powerful in object recognition involving the motion of the camera or the object observed [21]. The Kalman filter-based methods verify the detected motion information from video frames with the prior known dynamic models. Since the methods utilize the information across consecutive video frames, their robustness to errors make them ideal methods for poor illumination conditions and outdoor environments [22]. Hence, the Kalman-filter has seen a wide range of applications in object recognition and tracking [23] such as vehicles [24], pedestrians [25], human faces [26], and even human eyes [27]. However, there is no existing work regarding how to detect a flying bird. Most existing work assumes rigid objects such as vehicles or land markers [24] and objects with regular and known shapes. Also, the existing work does not need to worry about the convergence of Kalman filter or its variations because an ample amount of observation data is available. Unfortunately, those conditions do not hold for the detection of a flying bird. Since a flying bird is a non-rigid, deformable, irregular, and highly dynamic object, our approach has to take on those challenges to develop a new class of filtering algorithms.

Our group has developed systems and algorithms [28], [29] for networked robotic cameras for a variety of applications such as construction monitoring [30], distance learning [31], panorama construction [32], and nature observation [33]. Our previous work [34] details how to build an autonomous nature observation system using motion detection. We learn that mere motion detection cannot save the biologists from the herculean task of image sorting, which inspires this work.

III. Problem Description

A. Problem Context

Our system is a monocular vision system. To observe birds at a distance, the camera is equipped with a telephoto lens, which has a narrow field of view (FOV). For example, the camera used in our experiments has only $20°$ horizontal and $15°$ vertical FOVs, respectively. The position of objects with respect to the camera is based on a camera coordinate system. The camera coordinate system (CCS) is a right hand, 3D Cartesian coordinate system with its origin at the camera center, its z-axis along the optical axis, and its x-y plane parallel to the imaging plane. The x-axis and y-axis of the CCS are parallel to the u-axis and the v-axis of the image coordinate system, respectively.

From the knowledge provided by ornithologists, we know that a flying bird is usually an adult bird. A bird does not change its size once reaching its adulthood. Birds of the same species share a similar size and flying velocity range. This biological information allows us to distinguish the targeted specie from other moving objects.

B. Assumptions

To establish the bird detection problem, we also have the following assumptions,

- A fixed and pre-calibrated camera is used. Hence we know the accurate perspective projection matrix.
- In the captured image, the foreground motion zone has been segmented using the method in [34] prior to the bird filtering process. Although the motion could be caused by any objects, it must have a size of 25×25 pixels or larger in area. The assumption is proposed by ornithologists because they believe this is the requirement for a human to read an image to positively identify a bird. We name a motion zone satisfying this size requirement as a salient motion zone.
- There is only one salient motion zone in each frame because we only detect the specie of a singulated bird at this stage.
- The motion segmentation process has a segmentation error as much as half a pixel when computing the boundary of the moving object.
- We assume that the bird is flying along a straight line with a constant velocity when captured by the camera. Considering that the camera only has a narrow FOV and the size of the bird on the image has to be large enough to satisfy the salient motion constraint, the length of the motion sequence is usually short given the

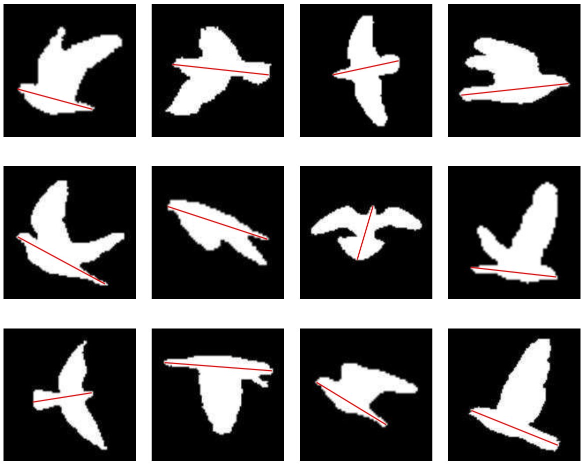

Fig. 2. Segmented bird flying poses. The white pixels in the binary map indicate the segmented salient motion zone. Bird body axes are overlaid on top of the segment image.

fast flying speed of the bird. In fact, the duration for a bird to fly across the camera FOV is around 1-2 seconds for most cases. Since the duration is short, the assumption usually holds.

C. Inputs and Output

The input of the problem is a sequence of n images, $F_1, F_2, ..., F_n$, which contain a moving object of any type. The frames are obtained by a motion detection process and the salient motion zone on each frame has been segmented as detailed in [34]. Each frame is time-stamped. Based on the information provided by ornithologists, we also know the body length l_b and the absolute flying velocity range $\mathcal{V} = [v_{min}, v_{max}]$ of the targeted bird species. The output of the problem is to determine whether the motion sequence is caused by the targeted bird species or not.

IV. Modeling a Flying Bird

To develop a bird filter, the key is to extract the bird flying information from the segmented bird motion sequence and associate the information with the known flying models and the prior information regarding the targeted specie. Let us first observe the motion sequence of the flying bird to investigate how to extract the bird flying information.

A. Bird Body Axis Filter

As detailed in [34], we segment the moving object from its background and obtain a set of motion sequence. Fig. 2 illustrates different flying poses of a pigeon. At first glance, it is unclear how to utilize this information because bird poses are not a simple discrete set of states. The wing configurations of the bird vary dramatically from frame to frame. The shape of the bird changes significantly as well.

As we scrutinize the collected flying pose data, we find that a bird does not bend or extend its body during the flight as illustrated in Fig. 2. Hence, we have,

Conjecture 1 (Invariant Body Length): A flying bird maintains a fixed body length during flight.

The conjecture has been confirmed by ornithologists and our data. This is an important finding because it provides an entry point to attack the bird detection problem. The ornithologists also utilize the bird body length as an important index to classify birds because adult birds from the same species share the same body length with little variance. Hence the problem becomes how to extract the body axis orientation and length of a flying bird from the segmented motion sequence.

It is nontrivial to extract the bird body axis and length from the isolated poses in Fig. 2 because a bird is a non-rigid and deformable object. However, if we superimpose the segmented bird flying pose data to the background image as illustrated in Fig. 1, a new finding appears:

Conjecture 2 (Body Axis Orientation): The orientation of the body axis of a flying bird is always close to the tangent line of its flying trajectory.

To validate our conjecture, we analyze 61 bird motion sequences with a total of 341 segmented birds that we have collected in past years. The result confirms the conjecture. The mean orientation difference is $0.8°$ and the standard deviation is $\sigma_b = 8.3°$. This means that the body axis orientation of a flying bird is usually very close to the tangent line of the flying trajectory, which inspires us to develop a bird body axis filter (BBAF) to extract bird body axes from the segmented motion zone.

Let us define the bird body line segment in the image frame as

$$\mathbf{z} = [u^h, v^h, u^t, v^t]^T, \tag{1}$$

where (u^h, v^h) is the head position and (u^t, v^t) is the tail position. From **z**, we can compute the body axis orientation

$$\theta = \text{atan2}(u^h - u^t, v^h - v^t),$$

and the body axis length

$$l = \sqrt{(u^h - u^t)^2 + (v^h - v^t)^2}.$$

Note that l is different from l_b. l is the projection of l_b on the image plane and is in units of pixels.

We know that the slope of the tangent line of the trajectory can be extracted easily based on the position of the salient motion zone on the background image. The red line in Fig. 1 is the approximate trajectory generated by linking the geometric center of each motion zone. The tangent line of the approximate trajectory can serve as an initial solution for the bird body axis orientation. However, since variance $\sigma_b \neq 0$, some refinements are required.

Define B as the boundary pixel set of the motion zone (i.e the boundary pixel set of the white pixels in each block in Fig. 2). Define $\bar{\theta}$ as the orientation of the corresponding tangent line of the flying trajectory. We set the search range for the orientation as $[\bar{\theta} - \sigma_b, \bar{\theta} + \sigma_b]$. Hence the solution to following optimization

$$\mathbf{z} = \arg \max_{\theta \in [\bar{\theta} - \sigma_b, \bar{\theta} + \sigma_b]; (u^h, v^h) \in B; (u^t, v^t) \in B} l \tag{2}$$

gives us the bird body axis orientation and length.

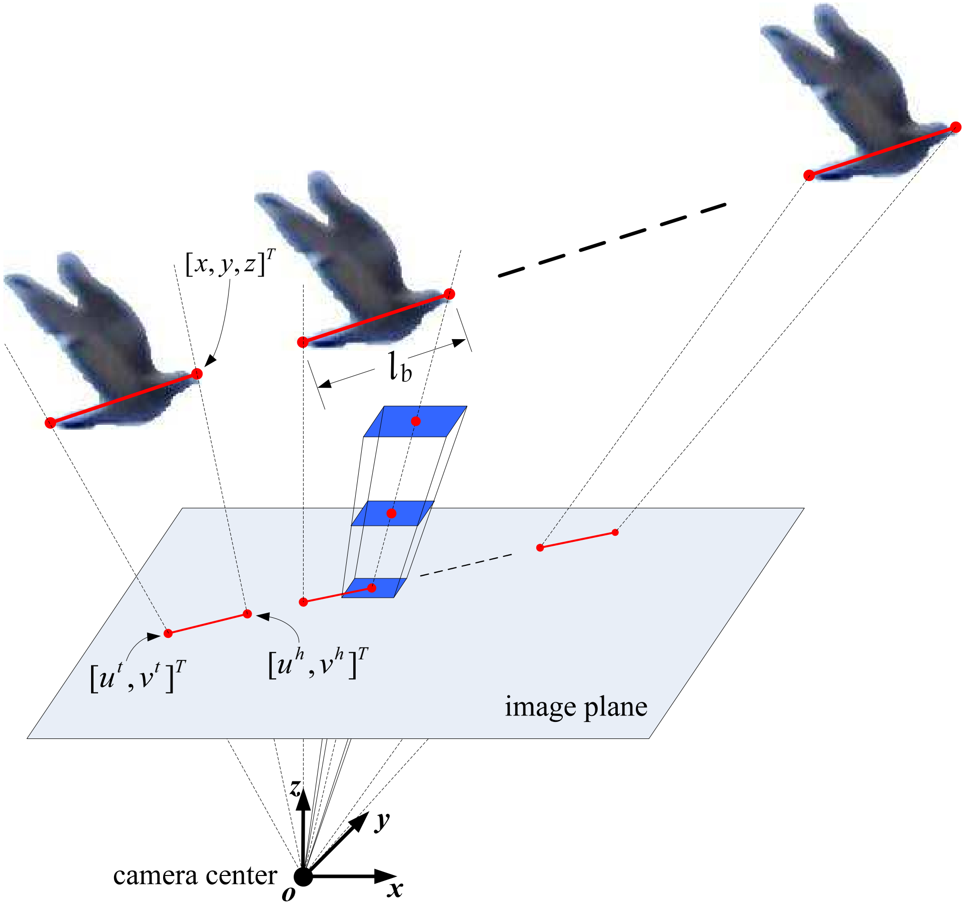

Fig. 3. An illustration of bird detection. When a bird flies across the camera FOV, the corresponding motion sequence can be used to extract a set of moving line segments that correspond to the body axis of the bird. The line segments are then verified using an EKF based on the known profile from the targeted specie. The segmentation error of the end of body axis are uniformed distributed in the u-v image plane and can be represented as an inverse pyramid when the error range is back-projected from the camera center to the 3D space.

B. Bird Flying Dynamics

To determine whether the motion information is caused by the targeted specie, we need to establish a bird flying model in the image frame. Let $\mathbf{p} = [x, y, z]^T$ denote the head position of the bird body axis and $\mathbf{v} = [\dot{x}, \dot{y}, \dot{z}]^T$ denote its velocity in the CCS. Since the bird flies along a straight line with a constant velocity, we have

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \end{bmatrix} = [\dot{x}, \dot{y}, \dot{z}, 0, 0, 0]^T = \begin{bmatrix} \mathbf{v} \\ \mathbf{0} \end{bmatrix}, \tag{3}$$

where the state variable $\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \end{bmatrix} \in \mathbb{R}^6$ describes the position and velocity of the bird head. Defining $\mathbf{x}_{tail} = [x^t, y^t, z^t]^T$ as the position of the bird tail, and we have

$$\mathbf{x}_{tail} = \begin{bmatrix} x - \dot{x} l_b / \|\mathbf{v}\| \\ y - \dot{y} l_b / \|\mathbf{v}\| \\ z - \dot{z} l_b / \|\mathbf{v}\| \end{bmatrix}. \tag{4}$$

As illustrated in Fig. 3, the relationship between the measurement data $\mathbf{z}$ defined in (1) and the corresponding state $\mathbf{x}$ can be described using the pin-hole camera model,

$$\mathbf{z} = \begin{bmatrix} fx/z \\ fy/z \\ fx^t/z^t \\ fy^t/z^t \end{bmatrix} = \begin{bmatrix} fx/z \\ fy/z \\ f\frac{x\|\mathbf{v}\| - l_b\dot{x}}{z\|\mathbf{v}\| - l_b\dot{z}} \\ f\frac{y\|\mathbf{v}\| - l_b\dot{y}}{z\|\mathbf{v}\| - l_b\dot{z}} \end{bmatrix} + \mathbf{w} = h(\mathbf{x}) + \mathbf{w}, \tag{5}$$

where f is the focal length of the camera divided by the side length of a square pixel on the CCD sensor and $\mathbf{w}$ represents the measurement noise.

V. Probable Observation Data Set-based EKF Method

A. Extended Kalman Filter

Eq. (2) provides the bird flying information extracted from images. The nonlinear dynamic system described by (5) captures the prior known information regarding the targeted specie. If the motion is caused by the targeted specie, then the bird body axis information provided by (2) should follow the nonlinear dynamic system described by (5), which can be validated using an EKF to track the states of the moving object.

Eqs. (3) and (5) can be re-written in a discrete-time form,

$$\mathbf{x}(k+1) = A(k+1)\mathbf{x}(k) + \mathbf{q}(k), \tag{6a}$$

$$\mathbf{z}(k) = h(\mathbf{x}(k)) + \mathbf{w}(k), \tag{6b}$$

where $\mathbf{q}(k) \in \mathbb{R}^6$ and $\mathbf{w}(k) \in \mathbb{R}^4$ represent the white Gaussian transition and measurement noises at time k with covariance matrix $Q(k) \in \mathbb{R}^{6\times6}$ and $W(k) \in \mathbb{R}^{4\times4}$, respectively,

$$\mathbf{q}(k) \sim \mathcal{N}(0, Q(k)),$$

$$\mathbf{w}(k) \sim \mathcal{N}(0, W(k)),$$

and $A(k+1)$ is the state transition matrix at time $k+1$,

$$A(k+1) = \begin{bmatrix} \mathbf{I}_{3\times3} & \Delta T(k+1|k)\mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix},$$

where $\Delta T(k+1|k)$ is the time interval between time k and time $k+1$.

We define $P \in \mathbb{R}^{6\times 6}$ as the covariance matrix for the state variable $\mathbf{x}$. The EKF for the system in (6) can be implemented as a state prediction step $\hat{\mathbf{x}}(k|k-1),\ \hat{P}(k|k-1)$ and measurement correction step $\hat{\mathbf{x}}(k|k),\ \hat{P}(k|k)$ recursively as follows,

$$\hat{\mathbf{x}}(k|k-1) = A(k)\hat{\mathbf{x}}(k-1|k-1), \tag{7a}$$

$$\hat{P}(k|k-1) = A(k)\hat{P}(k-1|k-1)A^T(k) + Q(k), \tag{7b}$$

$$K(k) = \frac{\hat{P}(k|k-1)H^T(k)}{H(k)\hat{P}(k|k-1)H^T(k) + W(k)}, \tag{7c}$$

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + K(k)(\mathbf{z}(k) - h(\hat{\mathbf{x}}(k|k-1))), \tag{7d}$$

$$\hat{P}(k|k) = (\mathbf{I}_{6\times 6} - K(k)H(k))\hat{P}(k|k-1), \tag{7e}$$

where $K(k) \in \mathbb{R}^{6\times 4}$ is the "Kalman gain" at time k and $H(k) \in \mathbb{R}^{4\times 6}$ is the Jacobian matrix of the function $h(\cdot)$ in (5) with respect to $\mathbf{x}$.

Recall that $\hat{\mathbf{x}}(k|k) = \begin{bmatrix} \hat{\mathbf{p}}(k|k) \\ \hat{\mathbf{v}}(k|k) \end{bmatrix}$. For the n-image motion sequence, the predicted $\hat{\mathbf{x}}(n|n)$ contains the bird velocity $\hat{\mathbf{v}}(n|n)$. The decision of accepting or rejecting the moving object as a member of the targeted specie is defined as the following indicator function,

$$I(\mathbf{Z}^{1:n}) = \begin{cases} 1 \text{ (accept)} & \text{if} \quad \|\hat{\mathbf{v}}(n|n)\| \in \mathcal{V} \text{ and EKF converges}, \\ 0 \text{ (reject)} & \text{otherwise}, \end{cases} \tag{8}$$

where $\mathbf{Z}^{1:n} = \{\mathbf{z}(1), \mathbf{z}(2), ..., \mathbf{z}(n)\}$ is the set of body axes across n-frames. $\mathbf{Z}^{1:n}$ is also referred to as the observed data. Eq. (8) basically states that the moving object is a member of the targeted specie if the EKF converges to the desired absolute velocity range $\mathcal{V}$.

B. *EKF Convergence*

As indicated in (8), automatically determining whether the EKF converges or not is necessary. Define the estimated state set as $\mathbf{X}^{1:n} = \{\hat{\mathbf{x}}(1|1), \hat{\mathbf{x}}(2|2), ..., \hat{\mathbf{x}}(n|n)\}$. Since velocity convergence implies position convergence, we determine the convergency of the EKF by inspecting the velocity component of $\mathbf{X}^{1:n}$,

$$\varepsilon(\mathbf{X}^{1:n}) = \sum_{k=2}^{n} \omega(k)\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|, \tag{9}$$

where $\omega(k) > 0$ is the weighting factor at time k. $\omega(k)$ is a monotonically-increasing function of k, which gives more weight to later states. $\omega(k)$ is usually pre-generated using simulated random inputs across the entire possible parameter range without measurement error (i.e. $W(k) = \mathbf{0}_{4\times 4}$). Setting $W(k) = \mathbf{0}_{4\times 4}$ is to ensure EKF convergency, which will be explained later in the paper. We repeat the EKF with randomized inputs for over 10^6 times to observe the quantity of $\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|$. Then the weighting factor is,

$$\omega(k) = \frac{1}{\sigma_v(\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|)}, \tag{10}$$

where function $\sigma_v(\cdot)$ computes the stand deviation of the variable over the total simulated trials. When the EKF converges, $\sigma_v(\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|)$ appears as a decreasing function of k. Hence $\omega(k)$ is an increasing function of k.

Therefore, $\omega(k)$ is a function of k but not a function of the current absolute velocity difference $\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|$ in (9). If $\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\| \to 0$, then $\varepsilon(\mathbf{X}^{1:n})$ is smaller than that of the case $\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\| \nrightarrow 0$. To determine the EKF convergence, we employ a threshold δ on $\varepsilon(\mathbf{X}^{1:n})$ and introduce a new indicate variable,

$$I_{\text{EKF}}(\mathbf{X}^{1:n}) = \begin{cases} 1 \text{ (converge)} \;\; \text{if} \quad \varepsilon(\mathbf{X}^{1:n}) < \delta, \\ 0 \text{ otherwise.} \end{cases} \tag{11}$$

Then the decision-making in (8) is re-written as,

$$I(\mathbf{Z}^{1:n}) = \begin{cases} 1 \text{ (accept)} \;\; \text{if} \quad \|\hat{\mathbf{v}}(n|n)\| \in \mathcal{V} \text{ and } I_{\text{EKF}}(\mathbf{X}^{1:n}) = 1, \\ 0 \text{ (reject)} \;\; \text{otherwise.} \end{cases} \tag{12}$$

The underlying condition for (12) to be an effective bird detection mechanism is that if observation $\mathbf{Z}^{1:n}$ is caused by the targeted bird species then the convergence of the EKF has to be guaranteed. Unfortunately, the condition usually does not hold due to two reasons: n is small and the measurement noise $\mathbf{w}(k)$ is too big. n is the number of images that contain the moving object. Due to the fact that the bird flies very fast, the bird can only stay in the camera FOV for less than 1 second for most of the time. Actually, $n < 11$ for most cases in our experiments. The measurement noise covariance matrix $W(k)$ is directly determined by the image segmentation error. Even at 0.5 pixel level, its relative range is 5% for a bird body length of 10 pixels. For the nonlinear deterministic discrete time system in (6), the large $W(k)$ means the EKF either fails to converge or converges very slowly according to [35]. The bird detection mechanism would have a close to 100% false negative rate if the simple EKF implementation is used, which makes it useless.

C. Probable Observation Data Set-based EKF Method

Since simply applying EKF cannot address the bird detection problem, a new approach is required. Let us assume there is no measurement noise (i.e. $W(k) = \mathbf{0}_{4\times 4}$) and no state transition noise $Q(k) = \mathbf{0}_{6\times 6}$. At each time k, the EKF in (7) is a system of equations with four inputs, which is the dimensionality of $\mathbf{z}$, and six outputs, which is the dimensionality of $\mathbf{x}$. We also know that matrix A introduces two constraints: the constant velocity and the linear trajectory. Therefore, the equation system can be solved within one step. The convergence of the EKF is not a problem when there is no noise provided that the bird does not fly in a degenerated trajectory (i.e. flying along the optical axis of the camera).

Although $Q(k) \neq \mathbf{0}_{6\times 6}$ for most cases, the state transition noise $\mathbf{q}(k)$ is often very small, which leads to the following lemma,

Lemma 1: The EKF described in (7) converges when $W(k) = \mathbf{0}_{4\times 4}$.

Proof: We skip the proof because our system in (6) is a linear time-invariant discrete time system with a nonlinear observer. The convergence of its EKF can be proved by directly applying the results in [35]. ■

This is also confirmed in our experiments in which the EKF converges nicely under 7 periods (i.e. $n \leq 7$).

At first glance, this result is useless because we cannot get rid of the measurement noise. However, this result opens the door to a new approach. Define the observation data without measurement error as $\mathbf{Z}^{1:n*} = [\mathbf{z}^*(1), \mathbf{z}^*(2), ..., \mathbf{z}^*(n)]^T$. Although we do not have $\mathbf{Z}^{1:n*}$, we know it is within the segmentation error range of $\mathbf{Z}^{1:n}$. For the k-th image, the measurement data is

$$\mathbf{z}(k) = [u^h(k), v^h(k), u^t(k), v^t(k)]^T.$$

Define the error-free measurement data at time k as

$$\mathbf{z}^*(k) = [u^{h*}(k), v^{h*}(k), u^{t*}(k), v^{t*}(k)]^T.$$

Recall that the segmentation error is within 0.5 pixels. Define

$$S_1(k) = [u^h(k) \pm 0.5], \qquad S_2(k) = [v^h(k) \pm 0.5],$$
$$S_3(k) = [u^t(k) \pm 0.5], \qquad S_4(k) = [v^t(k) \pm 0.5],$$

and the segmentation error range at time k as $\mathbb{S}(k)$. Hence,

$$\mathbf{z}^*(k) \in S_1(k) \times S_2(k) \times S_3(k) \times S_4(k) = \mathbb{S}(k). \tag{13}$$

We partition the entire segmentation error range set $\{\mathbb{S}(k), k = 1, 2, ..., n\}$ according to the convergence of the EKF using (11). Define the probable observation data set(PODS) $\mathbb{Z}^{1:n}$ as the set of observation data $\mathbf{Z}^{1:n}$ that satisfies the condition for the EKF convergence,

$$\mathbb{Z}^{1:n} = \{\mathbf{Z}^{1:n} | z(k) \in \mathbb{S}(k), k = 1, ..., n, \text{ and } \varepsilon(\mathbf{X}^{1:n}) \leq \delta\}. \tag{14}$$

Hence $\mathbf{Z}^{1:n*} \in \mathbb{Z}^{1:n}$. Each $\mathbf{Z}^{1:n}$ in the PODS is likely to be $\mathbf{Z}^{1:n*}$ and hence it is named as the probable observation data. For a given PODS $\mathbb{Z}^{1:n}$, there is a corresponding estimated state set $\mathbb{X}^{1:n}$, which contains a set of all possible estimated velocities at time n, which is defined as $\mathbb{V}$,

$$\mathbb{V} = \{\|\hat{\mathbf{v}}(n|n)\| \text{ such that } \mathbf{X}^{1:n} \in \mathbb{X}^{1:n}\}.$$

Then the decision making for our PODS-based EKF (PODS-EKF) method can be written as,

$$I(\mathbf{Z}^{1:n}) = \begin{cases} 1 \text{ (accept)} \quad \text{if} \quad \mathbb{V} \cap \mathcal{V} \neq \emptyset \text{ and } \mathbb{Z}^{1:n} \neq \emptyset, \\ 0 \text{ (reject)} \quad \text{otherwise.} \end{cases} \tag{15}$$

Hence we have the following lemma,

Lemma 2: If the non-degenerated observation data $\mathbf{Z}^{1:n}$ is triggered by a bird of the targeted specie, then $I(\mathbf{Z}^{1:n}) = 1$.

Proof: Since $\mathbf{Z}^{1:n}$ is triggered by the targeted specie, its corresponding $\mathbf{Z}^{1:n*}$ ensures the convergence of the EKF according to Lemma 1.

Define $\mathbf{X}^{1:n*}$ as the corresponding estimated states for $\mathbf{Z}^{1:n*}$. Hence

$$\varepsilon(\mathbf{X}^{1:n*}) < \delta \rightarrow \mathbb{Z}^{1:n} \neq \emptyset,$$

because $\mathbf{Z}^{1:n*} \in \mathbb{Z}^{1:n}$.

Following our naming convention, $\hat{\mathbf{v}}^*(n|n)$ is the velocity component of $X^*(n|n) \in \mathbf{X}^{1:n*}$. Since the observation data is not degenerated, $\|\hat{\mathbf{v}}^*(n|n)\| \in \mathcal{V}$. We also know $\|\hat{\mathbf{v}}^*(n|n)\| \in \mathbb{V}$ by definition, $\mathbb{V} \cap \mathcal{V} \neq \emptyset$ holds. Since both conditions are satisfied, $I(\mathbf{Z}^{1:n}) = 1$. ■

Lemma 2 ensures that the PODS-EKF method theoretically has a zero false negative rate in the bird detection, which is a very desirable property.

D. Approximate Computation for PODS-EKF

Computing the PODS $\mathbb{Z}^{1:n}$ is nontrivial. It is possible to use conventional searching methods such as a binary search. However, this would be very time consuming. Note that we actually do not need $\mathbb{Z}^{1:n}$ because all we need to know is whether the conditions $\mathbb{V} \cap \mathcal{V} \neq \emptyset$ and $\mathbb{Z}^{1:n} \neq \emptyset$ hold or not. This allows an approximation method. For a given observation $\mathbf{Z}^{1:n}$, we define the following optimization problem,

$$\widetilde{\mathbf{Z}}^{1:n} = \arg \min_{z(k) \in \mathbb{S}(k); k=1,\ldots,n} \varepsilon(\mathbf{X}^{1:n}), \tag{16}$$

where $\widetilde{\mathbf{Z}}^{1:n}$ is the optimal solution to the minimization problem above. Actually, (16) is a typical nonlinear optimization problem with the error range $z(k) \in \mathbb{S}(k); k = 1, ..., n$ and the EKF in (7) as constraints. There are many numerical methods from nonlinear programming that can be used here [36].We apply a sequential quadratic programming (SQP) method [37]. Define $\widetilde{\mathbf{X}}^{1:n} = \{\widetilde{\mathbf{x}}(1|), \widetilde{\mathbf{x}}(2|2), ..., \widetilde{\mathbf{x}}(n|n)\}$ as the estimated states corresponding to $\widetilde{\mathbf{Z}}^{1:n}$. We have the following lemma,

Lemma 3: $\varepsilon(\widetilde{\mathbf{X}}^{1:n}) > \delta \iff \mathbb{Z}^{1:n} = \emptyset$.

Proof: Since (16) is a minimization problem, $\widetilde{\mathbf{X}}^{1:n}$ yields the minimal $\varepsilon(\mathbf{X}^{1:n})$, namely,

$$\varepsilon(\widetilde{\mathbf{X}}^{1:n}) > \delta \iff \varepsilon(\mathbf{X}^{1:n}) > \delta, \forall \mathbf{X}^{1:n} \in \mathbb{X}^{1:n} \tag{17}$$

$$\iff \mathbb{Z}^{1:n} = \emptyset. \tag{18}$$

■

It is worth mentioning that this method is an approximation in computation because the nonlinear programming solver often falls in a local minimum instead of a global minimum.

Now we want to determine whether $\mathbb{V} \cap \mathcal{V} \neq \emptyset$. If we view the EKF output $\hat{\mathbf{v}}(n|n)$ as a function of $\mathbf{Z}^{1:n}$, it is continuous and differentiable with respect to each entry in $\mathbf{Z}^{1:n}$. Since $\mathbb{Z}^{1:n}$ is actually very small, the variance of the velocity in the set $\mathbb{V}$ is very small. Instead of comparing $\mathbb{V}$ to $\mathcal{V}$, we select a value in $\mathbb{V}$ to check if it is in $\mathcal{V}$. Define $\widetilde{\mathbf{v}}(n|n)$ as the velocity component of $\widetilde{\mathbf{x}}(n|n) \in \widetilde{\mathbf{X}}^{1:n}$. The chosen value is the $\|\widetilde{\mathbf{v}}(n|n)\|$ because it is readily available. Therefore, the approximation is

$$\|\widetilde{\mathbf{v}}(n|n)\| \in \mathcal{V} \iff \mathbb{V} \cap \mathcal{V} \neq \emptyset.$$

Due to the approximation, the zero false negative rate cannot be guaranteed. However, the false negative rate is still very low under the approximation as shown in the experimental results.

VI. Algorithm

We summarize our PODS-EKF based bird detection algorithm below in Algorithm 1. Note that the approximate computation of the PODS-EKF is used here.

Algorithm 1: PODS-EKF based Bird Detection Algorithm

```
input : $\bar{F}_i$ with segmented salient motion blocks, $i = 1, 2, ..., n$.
output: TRUE or FALSE for the targeted specie.
for each $\bar{F}_i$ do
    calculate the geometric center point $C_i$ of the bird;
Connect $C_i,\ i = 1, 2, ..., n$ to generate a piecewise linear trajectory;
Obtain $\bar{\theta}$ from the trajectory;
for each $\bar{F}_i$ do
    Obtain $\mathbf{z}(i)$ using the BBAF in (2);
solve the constrained nonlinear optimization problem in (16);
if $\|\widetilde{\mathbf{v}}(n|n)\| \in \mathcal{V}$ AND $\varepsilon(\widetilde{\mathbf{X}}^{1:n}) < \delta$ then
    return TRUE;
else
    return FALSE;
```

VII. Experiments

We have implemented the PODS-EKF algorithm and tested the algorithm on both the simulated data and the real data from field experiments. The computer used in the test is a desktop PC with Intel Core 2 Duo 2.13GHz CPU and 2GB RAM. The PC runs Microsoft Windows XP. The BBAF has been implemented using Microsoft Visual C++. The PODS-EKF filter has been implemented using Matlab v7.0. We chose Arecont Vision 3100 high resolution networked video cameras as the imaging devices. The camera runs at 11 frames per second with a resolution of 3 Mega-pixel per frame. The lens for the camera is a Tamron auto iris vari-focus lens with a focal length range of 10-40 mm. We have adjusted the lens to ensure a 20° horizontal FOV.

A. Bird Body Axis Filter Test

The first thing we want to verify is whether the BBAF is capable of extracting the bird body axis from the noisy data. We compare the output of BBAF with a ground truth case on 61 bird motion sequences with a total of 341

TABLE I

TWO SPECIES USED IN THE EXPERIMENTS

Species	l_b (cm)	$\mathcal{V}$ (km/h)
Rock pigeon	34	$[25, 55]$
IBWO	48	$[30, 60]$

segmented birds. The ground truth case is a human's choice in bird body axes. The difference between the BBAF output and the ground truth has a mean of 0.3° and a standard deviation of 3.7°. The statistics test shows that the two populations come from the same distribution with statistic significance, which is satisfying.

B. Simulation

The second test we have conducted is to test the performance of our PODS-EKF using the simulated inputs. The simulated inputs allow us to test the bird detection performance under a full range of possible changes in the parameter settings, which is usually unavailable in a physical experiment setting.

1) Generating a random input: Let us introduce how we generate a random input $\mathbf{Z}^{1:n}$. First, four random numbers are generated as the coordinates of two random points in the image plane. These two image points determine a straight line in the image. The straight line and the camera center determine a motion plane in which the motion sequence will be generated. We know that the camera FOV is a pyramid with its top vertex at the camera center. The plane intersects with two faces of the pyramid. The fifth random binary number is generated to choose one of the two faces as the initial face through which the bird enters the camera FOV. The chosen face intersects with the motion plane and yields a line segment. We generate the sixth random number as a point on this line segment. This point is used as the initial position of the bird. This line segment's corresponding line divides the motion plane into two halves. We are interested in the half motion plane that intersects with the pyramid. The seventh random number in the range of $[0, \pi)$ is generated as the pitch angle of the bird heading on the half motion plane. Finally, the eighth random number is used to generate the velocity of the bird. Hence, 8 random numbers determine a complete trajectory of a flying bird. By projecting the trajectory back to the image plane with predefined time stamps and the preset bird body length, we obtain a random input $\mathbf{Z}^{1:N}$.

2) EKF Convergence without measurement error: An immediate step in simulation is to verify if the EKF converges without measurement noise. Although Lemma 1 ensures the convergence in theory, it is unclear how many steps it would take. We simulate two types of birds in the test: rock pigeons and the ivory-billed woodpeckers (IBWO). The former is a common bird that is easy to be found in Texas and the later is a rare bird which our system is used to search for. As indicated in Table I, a rock pigeon is a relatively small bird and an IBWO is a large bird. The two species represent a pretty good coverage of different bird species.

For each species, we generate 10^6 different sets of random inputs to test the EKF. We record the values of $\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|,\ k = 2, 3, ..., n$ for the first 11 steps. We choose $n = 11$ because that is the typical number in our physical experiments. The average values of $\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|$ over the 10^6 trials are as

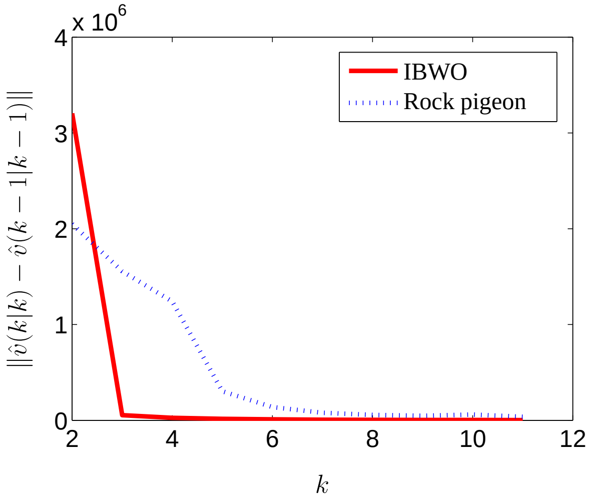

Fig. 4. Average $\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|$ vs. time k.

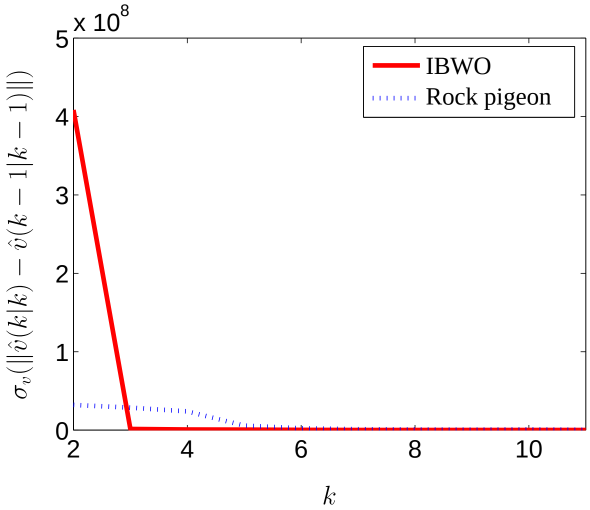

Fig. 5. $\sigma_v(\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|)$ vs. time k.

shown in Fig. 4. It is clear that the EKF converges within seven steps. Recall $\sigma_v(\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|)$ is the standard deviation of $\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|$. Its inverse $\omega(k)$ is the weighting factor in (11). Fig. 5 illustrates how $\sigma_v(\|\hat{\mathbf{v}}(k|k) - \hat{\mathbf{v}}(k-1|k-1)\|)$ decreases as k increases.

3) Performance of PODS-EKF under simulated inputs: Now we are ready to evaluate the performance of the PODS-EKF bird detection algorithm. The targeted specie is the IBWO. We generate the set of random inputs to mimic the IBWO that have a similar size and a similar flying velocity. We set a velocity range of 20 km/h to 70 km/h with an incremental step of 5 km/h and a bird size range from 40 cm to 60 cm with an incremental step of 2 cm. We also observe how the algorithm performs under different EKF convergence thresholds δ. For each setting of the input data, 20 trials are carried out. Fig. 6 demonstrates how the rates of false positive (FP) and false negative

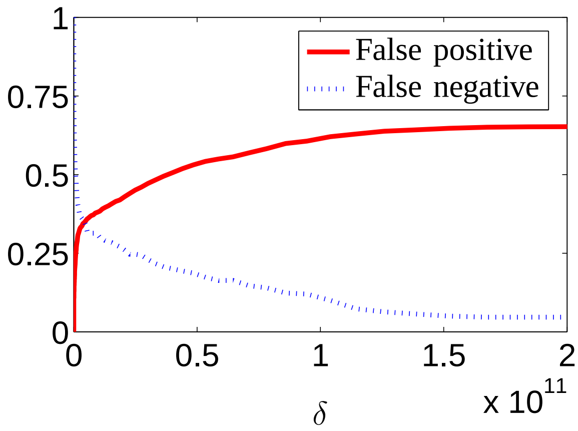

Fig. 6. False positive and false negative rates with different δ.

TABLE II

EXPERIMENTAL RESULTS FROM THE ROCK PIGEON DETECTION EXPERIMENT.

	pigeon	not pigeon
predicted pigeon	27	24
predicted not pigeon	2	66

(FN) change according to δ. As shown in the figure, the algorithm is not sensitive to the selection of the threshold δ after $\delta > 1.25 \times 10^{11}$, which is desirable. The false negative rate can be reasonably controlled to less than 5%, which again confirms that the approximation in Section V-D is reasonable. The false positive rate is around 65%, which is a little high. However, considering the fact that we are comparing the bird with objects similar in size and velocity, this result is not surprising. In fact, the algorithm should behave better in real tests where the noise from moving objects has a much larger range in both size and velocity. On the other hand, the monocular system has its problem in detecting objects close to its optical axis, which also contributes to the high false positive rate.

C. Physical Experiments

We have conducted two field experiments: detecting flying rock pigeons and assisting the search of the legendary ivory-billed woodpecker.

1) Detecting a flying pigeon: In this experiment, the targeted specie is rock pigeons. With a camera setup in room 311B of H.R. Bright Bldg. from May 2005 to October 2005 and another camera setup in Bayou DeViewm AR from Oct. 2006 to Oct. 2007, we have collected 119 events with $n > 7$ for each motion sequence. 29 of the sequences are rock pigeons while the other 90 are not pigeons, which are image sequences of typical environment noises such as vibrating trees, falling leaves, flying insects, and other bird species. The PODS-EKF filtering algorithm is applied to the data set with the threshold $\delta = 1.25 \times 10^8$. The outcome of the algorithm is summarized in Table II.

Table II indicates that our filtering algorithm can achieve very low false negative $(2/29 = 6.9\%)$. This is

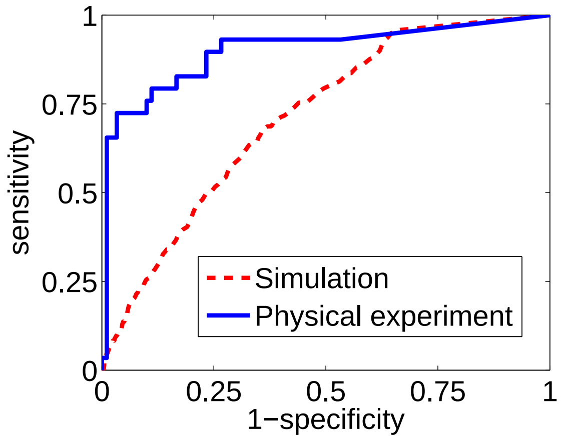

Fig. 7. The ROC curves using the data from both the simulation and the rock pigeon experiments. The corresponding δ range for the simulation data is $[6.51 \times 10^4, 9.64 \times 10^{11}]$ and the corresponding δ range for the rock pigeon experiment is $[1.32 \times 10^6, 9.03 \times 10^8]$.

very important for the purpose of finding birds of targeted species. The false positive rate is 26.7%. The overall performance is actually better than that of the simulation results. This is due to the fact that it is much easier for the algorithm to distinguish the targeted specie from noises such as flying insects and falling leaves. As illustrated in Fig. 7, we also draw ROC curves using the data from both the simulation and physical experiments with the rock pigeon. The area under the ROC curve of the simulation data is 71.2% under the simulation and the area under the ROC curve from the rock pigeon experiment is 90.0%, which again shows that the algorithm performs much better in physical experiments.

2) Assisting the search of the legendary ivory-billed woodpecker in Arkansas: Since October 2006, our team began to assist the search for the thought-to-be-extinct ivory-billed woodpecker (IBWO). The IBWO is the largest woodpecker in North America and was last seen over 60 years ago. Sporadic sightings have been reported in past decades but no definite evidences such as a clear picture of the bird have been available. In October 2006, we installed a camera system in Bayou DeView wildlife refuge in Arkansas, where sightings of the bird were reported in 2004, in order to capture any possible activities of the IBWO. Due to the low false negative rate, our PODS-EKF algorithm is very desirable for this type of application. Fig. 8 illustrates the setup. The system monitored the sky from Oct. 2006 to Oct. 2007. Although we have detected several species of similar size, no IBWO has been captured.

VIII. Conclusion and Future Work

We reported our development of a bird detection algorithm. To extract the invariant information from a flying bird, a BBAF that reports the bird body length and orientation was developed. We then developed a model-based detection approach that verifies the body axis information with the known bird flying dynamics. We show that an extended Kalman filter (EKF) cannot be directly applied because the EKF would not converge due to the high

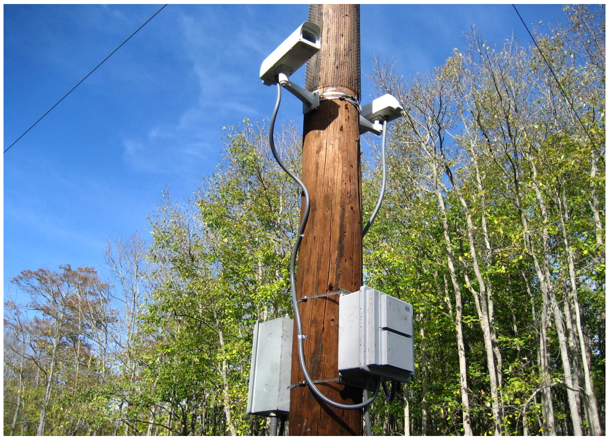

Fig. 8. Assisting the search of the IBWO.

measurement error introduced by image segmentation and the limited observation data due to the high flying speed of the bird. Instead, we developed a novel Probable Observation Data Set (PODS)-based EKF method. The detection is based on whether the PODS is non-empty and the corresponding velocity is within the known bird flying velocity profile. The algorithm has been extensively tested using both simulated inputs and physical experiments. The results were satisfying and the PODS-EKF bird detector has less than 7% false negative rate and 90% area under the ROC curve in physical experiments.

In the future, an immediate extension is to consider the case without the linear flying trajectory and/or the constant velocity. We will consider the simultaneous detection of a flock of birds using a single camera or multiple cameras. It is also interesting to consider the use of a robotic camera to combine tracking with detection. A pan-tilt-zoom robotic camera can give a closer view of a flying bird, which reduces the measurement error at a price of increasing the state transition error and the nonlinearity of the system. We will investigate how to achieve the best tradeoff. We plan to utilize multiple cameras or moving cameras in the detection.

Acknowledgements

Thanks to K. Goldberg for insightful discussions. Thanks to N. Qin for her implementation of bird motion detection. We are grateful to J. Fitzpatrick and R. Rohrbaugh of the Cornell Ornithology Lab, J. Yi, and J. Liu for providing inputs for system design and providing their help technically and logistically in field experiments. Thanks to B. Fine and T. Southard for their help in proofreading the manuscript. Thanks to R. Volz for his insightful input in the selection of camera parameters and to George Lee and Junku Yuh of NSF for their support. Thanks to H. Lee, C. Kim, and H. Wang for their contribution to the Networked Robot Lab at Texas A&M University. Thanks to Richard Crosset, Billy Culbreath and the U.S. Fish and Wildlife Service. Thanks to Robert Johnson and the Arkansas Electric Cooperatives Corp.. Thanks to Patsy Arnett and the support from Brinkley Convention Center, and to Mary Harlin and her family for providing space for our wireless antenna in Arkansas.

References

[1] L. W. Gysel and E. M. J. Davis, “A simple automatic photographic unit for wildlife research,” *Journal of wildlife management*, vol. 20, pp. 451–453, 1956.

[2] T. E. Kucera and R. H. Barrett, “The trailmaster camera system for detecting wildlife,” *Wildlife Society Bulletin*, vol. 21, pp. 505–508, 1993.

[3] K. Iida, R. Takahashi, Y. Tang, T. Mukai, and M. Sato, “Observation of marine animals using underwater acoustic camera,” *Japanese Journal of Applied Physics*, vol. 45, no. 5B, pp. 4875–4881, 2006.

[4] T. L. Cutler and D. E. Swann, “Using remote photography in wildlife ecology: a review,” *Wildlife Society Bulletin*, vol. 27, no. 3, pp. 571–581, 1999.

[5] M. D. Sanders and R. F. Maloney, “Causes of mortality at nests of ground-nesting birds in the upper waitaki basin, south island, new zealand: a 5-year video study,” *Biological Conservation*, vol. 106, no. 2, pp. 225–236, 2002.

[6] M. M. Stake and D. A. Cimprich, “Using video to monitor predation at black-capped vireo nests,” *BioOne*, vol. 105, no. 2, pp. 348–357, 2003.

[7] S. B. Lewis, P. DeSlmone, K. Titus, and M. R. Fuller, “A video surveillance system for monitoring raptor nests in a temperate rainforest environment,” *Northwest Science*, vol. 78, no. 1, pp. 70–74, 2004.

[8] AfricaWebCams, “http://www.zulucam.org/,” 2005.

[9] tigherhomes.org, “http://www.tigerhomes.org/animal/web-cams.cfm,” 2005.

[10] D. E. Weber, B. Grosser, and G. Fried, “http://bugscope.beckman.uiuc.edu/,” 2005.

[11] newyorkwild.org, “http://www.newyorkwild.org/webcams/webcams.htm,” 2005.

[12] JamesReserve, “http://www.jamesreserve.edu/webcamsphp.lasso,” 2005.

[13] SwanCam, “http://www.osage.net/ mccb/trumpeterswan.html,” 2005.

[14] wildcam.com, “http://www.wildcam.com/,” 2005.

[15] N. Friedman and S. Russell, “Image segmentation in video sequences: a probabilistic approach,” in *Proceedings of 13th Conference on Uncertainty in Artifical Intelligence*, 1997, pp. 1–3.

[16] Y. H. Yang and M. D. Levine, “The background primal sketch: an approach for tracking moving objects,” in *Machine Vision and Applications*, vol. 5, 1992, pp. 17–34.

[17] M. Kohle, d. Merkl, and J. Kastner, “Clinical gait analysis by neural networks: Issues and experiences,” in *Proceedings of IEEE Symposium on Computer-Based Medical Systems*, vol. 5, 1997, pp. 138–134.

[18] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, “Using adaptive tracking to classify and monitor activies in a site,” in *Proceedings of IEEE Symposium on Computer-Based Medical Systems*, vol. 5, 1997, pp. 138–134.

[19] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction,” in *6th European Conference on Computer Vision. Dublin, Ireland*, June 2000, pp. 751–767.

[20] C. Ridder, O. Munkelt, and H. Kirchner, “Adaptive background estimation and foreground detection using kalman-filtering,” in *Proceedings of International Conference on Recent Advances in Mechatronics*, 1995, pp. 193–199.

[21] G. L. Foresti, “Object recognition and tracking for remote video surveillance,” *IEEE Transactions on Circuits and Systems for Vieo Technology*, vol. 9, no. 7, pp. 1045–1062, October 1999.

[22] ——, “Object detection and tracking in time-varying and badly illuminated outdoor environments,” *Opt. Eng.*, vol. 37, no. 9, pp. 2550–2564, September 1998.

[23] A. Lipton, H. Fujiyoshi, and R. Patil, “Moving target classification and tracking from real-time video,” in *The Fourth IEEE Workshop on Applications of Computer Vision, Princeton, NJ, USA*, October 1998, pp. 8–14.

[24] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos, “Switching kalman filter-based approach for tracking and event detection at traffic intersections,” in *The 13th Mediterranean Conference on Control and Automation, Limassol, Cyprus*, June 2005, pp. 1167–1172.

[25] F. Xu, X. Liu, and K. Fujimura, “Pedestrian detection and tracking with night vision,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 63–71, March 2005.

[26] C. Seong, B. Kang1, J. Kim, and S. Kim, “Effective detector and kalman filter based robust face tracking system,” in *Advances in Image and Video Technology, Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006, pp. 453–462.

[27] W. Abd-Almageed, M. Fadali, and G. Bebis, “A non-intrusive kalman filter-based tracker for pursuit eye movement,” in *American Control Conference, Anchorage, Alaska, US*, May 2002, pp. 1443–1447.

[28] D. Song, F. van der Stappen, and K. Goldberg, “Exact algorithms for single frame selection on multi-axis satellites,” *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 1, pp. 16–28, Jan. 2006.

[29] D. Song and K. Goldberg, “Approximate algorithms for a collaboratively controlled robotic camera,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1061–1070, Nov. 2007.

[30] D. Song, Q. Hu, N. Qin, and K. Goldberg, “Automating high resolution panoramic inspection and documentation of construction using a robotic camera,” in *IEEE Conference on Automation Science and Engineering, Edmonton, Canada*, Aug. 2005.

[31] K. Goldberg, D. Song, and A. Levandowski, “Collaborative teleoperation using networked spatial dynamic voting,” *The Proceedings of The IEEE*, vol. 91, no. 3, pp. 430–439, March 2003.

[32] N. Qin, D. Song, and K. Goldberg, “Aligning windows of live video from an imprecise pan-tilt-zoom robotic camera into a remote panoramic display,” in *IEEE International Conference on Robotics and Automation (ICRA), Orlando, Florida*, May 2006.

[33] D. Song, N. Qin, and K. Goldberg, “Systems, control models, and codec for collaborative observation of remote environments with an autonomous networked robotic camera,” *Autonomous Robots*, vol. 24, no. 4, pp. 435–449, May. 2008.

[34] D. Song, N. Qin, Y. Xu, D. Luneau, and K. Goldberg, “System and algorithms for an autonomous observatory assisting the search for the ivory-billed woodpecker,” in *IEEE International Conference on Automation Science and Engineering (CASE), Washington DC*, August 2008.

[35] M. Boutayeb, H. Rafaralahy, and M. Darouach, “Convergence analysis of the extended kalman filter used as an observer for nonlinear deterministic discrete-time systems,” *IEEE Transactions on Automatic Control*, vol. 42, no. 4, pp. 581–586, April 1997.

[36] M. Bazaraa, H. shelrali, and C. Shetty, *Nonlinear Programming: Theory and Algorithms*. John Wiley and Sons, Inc, 1993.

[37] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, “Procedures for optimization problems with a mixture of bounds and general linear constraints,” *ACM Transactions on Mathematical Software*, vol. 10, no. 3, pp. 282–298, September 1984.