3D Pose Estimation of Bats in the Wild

Mikhail Breslav[1], Nathan Fuller[2], Stan Sclaroff[1], and Margrit Betke[1]
[1] Department of Computer Science and [2] Department of Biology
Boston University
{breslav, nwfuller, sclaroff, betke} @bu.edu

Abstract

Vision-based methods have gained popularity as a tool for helping to analyze the behavior of bats. Though, for bats in the wild, there are still no tools capable of estimating and subsequently analyzing articulated 3D bat pose. We propose a model-based multi-view articulated 3D bat pose estimation framework for this novel problem. Key challenges include the large search space associated with articulated 3D pose, the ambiguities that arise from 2D projections of 3D bodies, and the low resolution image data we have available. Our method uses multi-view camera geometry and temporal constraints to reduce the state space of possible articulated 3D bat poses and finds an optimal set using a Markov Random Field based model.

Our experiments use real video data of flying bats and gold-standard annotations by a bat biologist. Our results show, for the first time in the literature, articulated 3D pose estimates being generated automatically for video sequences of bats flying in the wild. The average differences in body orientation and wing joint angles, between estimates produced by our method and those based on gold-standard annotations, ranged from $16^\circ - 21^\circ$ *(i.e.,* $\approx 17\% - 23\%$*) for orientation and* $14^\circ - 26^\circ$ *(i.e.,* $\approx 7\% - 14\%$*) for wing joint angles.*

1. Introduction

Our work is motivated by the desire biologists and aerospace engineers have to understand how bats fly and why they behave the way they do. New questions are being asked about bat behavior: How do large numbers of bats behave as a group? What is their behavior when they forage? How do bats maneuver through dense vegetation while avoiding obstacles? To aid in answering these types of questions, researchers have looked to video data for clues. With advancements in computer vision research, automated video analysis tools have helped with the quantitative analysis of video data, saving biologists time and labor. Bats flying in the wild are shown in Figure 1.

Figure 1. Bats emerging from a cave in Texas at sunset.

Vision-based methods have been used to detect bats in visible and thermal-infrared video [3, 20, 23], track them in 3D [3, 20, 23], and analyze their kinematics, behaviors, and flight trajectories [3, 20]. An important distinction among these works is whether they deal with data captured in a laboratory or in the wild, and whether they model a bat as a point or a 3D articulated body.

Works that deal with video data of bats in the wild have modeled bats as points instead of articulated bodies [20, 23]. To uncover detailed flight behavior of bats in the wild, however, scientists would like to retrieve their 3D articulated motion. Works that do model bats as 3D articulated bodies have relied on data of bats in confined laboratory spaces, where the motion of a bat can be captured up close in great detail [3, 11, 17]. The work of Bergou et al. [3] modeled a bat with a 52 degree of freedom (DOF) articulated model, whose parameters are estimated from real data. Their data was obtained by first attaching tape markers to various landmarks on the bat, then placing the bat in a confined flight corridor where cameras can be positioned as close as necessary, and finally capturing flight data using high frame rate (1000 fps) cameras. This high quality data, from multiple cameras, served as input to a 3D tracking algorithm whose output led to a set of 3D pose estimates. While this method and similar approaches may offer relatively accu-

rate 3D pose estimates, they have several limitations. First, these methods are not suited for bats in the wild, where the use of tape markers is especially impractical. Second, there is no guarantee that bat behavior observed in confined laboratory spaces is representative of behaviors exhibited in the wild. As a result, analyses performed on data obtained in a laboratory may be misleading or limited in scope. Lastly, laboratory experiments tend to be performed on a small number of bats, one at a time, which is of limited use for studies on group behavior.

Our proposed work bridges the gap between methods that use 3D articulated bat models and methods that work on data of bats in the wild. In particular, we propose a model-based framework for estimating the 3D pose of bats in the wild, given multiple low resolution camera views. We call the system we designed based on this framework 3D-PEB for 3D Pose Estimation of Bats in the wild.

3D pose estimation for bats in the wild is a new problem that can build on a substantial amount of work that exists on 3D pose estimation for people. Several survey papers detail the large body of work that exists for human hand pose estimation, head pose estimation, and full body pose estimation [8, 13, 14, 15]. Among this large body of work, one popular class of approaches is the model-based approach. These works leverage a 3D graphics model by generating a large amount of synthetic views labeled with the 3D pose that generated them. These labeled views can then be used directly to compare with an input view. One example of such an approach is the 3D hand pose estimation work by Athitsos & Sclaroff [2]. In their work, a synthetic articulating hand model was used to generate a large set of views. Each view of the hand, encoded by edge based features, represented the appearance of a posed hand from the viewpoint of an orthographic camera. 3D hand pose was estimated for a novel input image by finding the view whose edges match best to the input. Our approach is also model-based.

Another class of 3D pose estimation methods are those that learn a regressor or classifier from a motion capture dataset. An example of one of these methods was the work by Agarwal & Triggs [1] for 3D body pose estimation. Their training data consisted of images of different full body poses, along with the joint angles that parameterize the 3D body pose. Then, a Relevance Vector Machine (RVM) was trained to learn a mapping from shape context based features to 3D body pose. Unfortunately, motion capture data is not readily available for the bats we studied, limiting the applicability of this class of methods.

Some 3D pose estimation methods, like ours, take advantage of multiple cameras. The work on 3D human upper body pose estimation by Hofmann & Gavrila [10] used the view in each camera to hypothesize a set of 3D pose candidates. The 3D pose candidates were reprojected into other camera views to evaluate a likelihood. Additionally, like our work, temporal constraints were used.

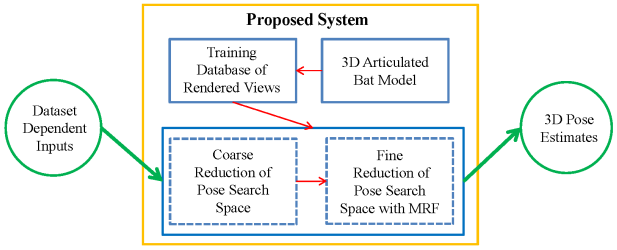


Figure 2. Overview of the proposed 3D-PEB system. Input and output are shown with green circles, system components with blue rectangles, and system dependencies with arrows.

More recently, depth cameras have been used for 3D pose estimation [19]. Currently, depth cameras are not viable to record bats in the wild, so such approaches have limited applicability to our problem.

Our main contribution is the 3D-PEB system for estimating the articulated 3D pose of bats in the wild, a problem for which we are the first to offer a vision-based approach. The second contribution of our work is a description of the challenges that arise when estimating 3D pose for non-human articulated bodies, which often lack good 3D models and training data, and solutions to address them. The third contribution is a set of 3D pose estimates over time, describing what the body and wings of individual bats are doing during an emergence of a group of bats from a cave. Our fourth contribution is a 3D graphics model of the *Tadarida brasiliensis* bat, along with calibrated images of bats, both of which are available online for public use.[1] Lastly, we introduce a part-based feature for use with low resolution images of bats.

2. Methods

Our proposed system 3D-PEB is made up of four components, shown in Figure 2. The system takes dataset-dependent inputs and produces articulated 3D bat pose estimates as output. The training database of rendered views only depends on the 3D articulated bat model, and is created once with the purpose of being reused. The training database of rendered views and the dataset-dependent inputs jointly serve as input to the subsystem responsible for reducing the pose search space. This subsystem includes one component that yields a coarse reduction of the pose search space, and another component which uses a Markov Random Field (MRF) to produce a fine reduction of the pose search space. The final system output is a set of articulated 3D pose estimates. This framework is detailed in the following subsections.

[1] http://www.cs.bu.edu/faculty/betke/research/3dpeb/

2.1. Dataset Dependent Inputs

3D-PEB operates on inputs extracted from video datasets. For the purposes of our specific application, datasets are videos of bats recorded by two or three cameras. We make the assumption that among the many bats flying through the field of view of the cameras, some remain unoccluded in each view for some interval of time. The task of 3D-PEB is to estimate the 3D pose of these individual bats for an interval of time represented by n frames. To reduce the scope of this work, we assume the following to be inputs to 3D-PEB:

- Segmentations $\{S_{i,j}\}$ of the bat for cameras $\{i\} \subset \{1, 2, 3\}$, for all frames $j \in [1 \cdots n]$.
- A description of the relative camera geometry, typically obtained from a calibration procedure.
- A trajectory of the 3D position of the bat $\{\mathcal{T}_j\}$, $j \in [1 \cdots n]$

These inputs can be obtained using a camera calibration tool [9], and algorithms for detecting bats in thermal video [20] and producing 3D tracks [23]. A sample input segmentation and a typical camera setup are shown in Fig 5.

2.2. 3D Graphics Model

Model-based methods that estimate the 3D pose of an articulated body require a 3D graphics model of that articulated body. In the case of bats, there are many different species, so each would potentially require its own 3D graphics model. In this work, we focused on the bat species *Tadarida brasiliensis* for which 3D graphics models are not readily available. To acquire a model, we built our own using Blender [6], a free 3D modeling tool.

We based the overall shape of our model, seen in Figure 3a, on illustrations by biologists [4, 11]. The size of the model was determined by the average wingspan (≈ 284 mm) and aspect ratio (9) for *Tadarida brasiliensis*, measured by biologists [4]. To aid in modeling the articulation of the wings, we referenced illustrations [11, 21] and video materials [11]. We designed a model with two joints per wing, each having one degree of freedom, roughly corresponding to the elbow and the wrist, see Fig. 3b. Our model is sufficiently powerful to approximate the flying motion of a bat, typically characterized by a wingbeat cycle consisting of a downstroke and an upstroke. We define ten key wing configurations that represent a sampling of a full wingbeat cycle, see Figure 3c. The wings are assumed to move in a symmetric manner, so one pair of angles $[\theta_1, \theta_2]$ is sufficient to describe the configuration (articulation) of both wings.

2.3. Training Database of Rendered Views

We used our 3D graphics model to render a large set of labeled views, collectively referred to as the 'training database.' A good training database should capture the variation in appearance of a bat as its articulated 3D pose changes. In this paper, the term '3D pose,' which we use interchangeably with 'articulated 3D pose,' means a description of how the body of a bat is oriented in 3D, together with the configuration of its wings. Specifically, the 3D model is assumed to be centered at the origin of a world coordinate system where its 3D orientation is described by a unit quaternion. The wing configuration is specified by a pair of angles $[\theta_1, \theta_2]$ (Sect. 2.2). A fixed orthographic camera model is used to render the appearance of the bat. This rendered view v_i is labeled with the 3D pose of a bat p_i, and constitutes a single training sample $d_i = (v_i, p_i)$.

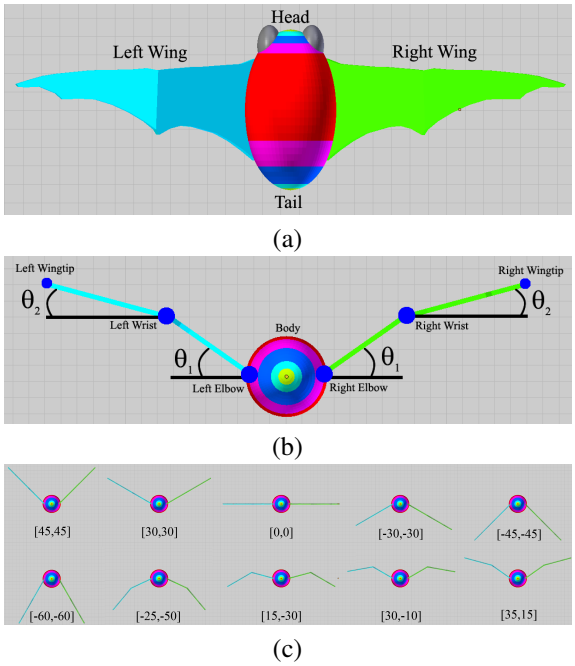


Figure 3. Proposed 3D graphics model of a *Tadarida brasiliensis* bat (a) Top down view (b) Back view of the model where the articulation of the elbow determines angle θ_1 and the wrist determines angle θ_2. (c) Video-based observations were used to manually define 10 key wing configurations. These 10 configurations, represented by a pair of angles (in degrees), describe a sampling of a typical wingbeat cycle. Shown from left to right, and top to bottom, is the downstroke phase followed by the upstroke phase.

Ideally, our training database would need to satisfy the following property: for any possible view v_{in}, generated by a camera observing 3D pose p_{in} at test time, there should be at least one training sample d_j with a similar view v_j produced by a similar 3D pose p_j. In essence, this property can be satisfied for some definition of 'similar' if the training database acts as a good approximation to an ideal database, which would be infinite in size and contain all possible views of a flying bat.

To approximate an ideal training database, we densely

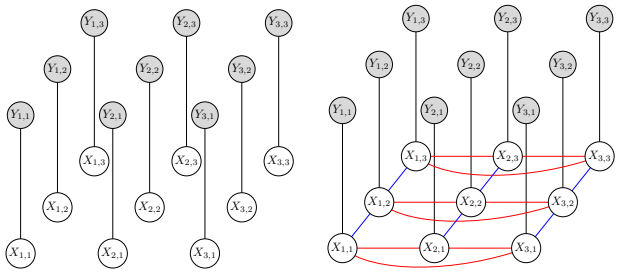


Figure 4. Undirected graphical models are shown for the case of 3 time frames and 3 cameras, where observed nodes are gray and hidden nodes are white. (a) Basic model. (b) Basic model with additional constraints represented by edges.

sampled the space of 3D orientations and the space of wing configurations. If a sphere of views is considered, then sampling the polar angle every 10° for 360°, elevation every 10° for 180°, and camera roll every 10° for 360° yields a total of 23,328 views. Instead of moving the camera, we equivalently kept the virtual camera fixed and rotated the bat according to 23,328 randomly [12] generated unit quaternions. To keep the database size relatively small, but sufficient, we kept the wing configurations to one of ten discrete parameters (Fig. 3c). All together, 23,328 orientations for each of the 10 different wing configurations yielded a database of 233,280 views. In practice, the view v_i associated with each training sample d_i can be represented by a d-dimensional feature vector f_i (Sect. 3.1). The training samples stored have the form $d_i = (f_i, p_i)$.

2.4. Pose Estimation via Graphical Model

We model the problem of articulated 3D bat pose estimation with an undirected graphical model. For each unique pair of cameras and frame numbers (i, j), $i \in \{1, 2, 3\}$, $j \in [1, \cdots, n]$, there are two types of nodes in the graph. The first type of node is labeled by the random variable $X_{i,j}$ and it represents the true 3D pose of the bat at frame number j. An $X_{i,j}$ is associated with every camera i. Since the true 3D pose of a bat is not directly accessible, $X_{i,j}$ is considered to be a hidden variable or state. The second type of node is labeled by the random variable $Y_{i,j}$ and it represents the appearance or observation of a bat as imaged by camera i at frame number j. Typically $Y_{i,j}$ will be a feature vector extracted from a segmented bat. All together, this model will have cn hidden state nodes $\{X_{i,j}\}$, and cn observation nodes $\{Y_{i,j}\}$ (c is the number of cameras used); for convenience, we write $X = \{X_{i,j}\}$, and $Y = \{Y_{i,j}\}$. Figure 4a shows an example of this undirected graphical model for the case of three frames and three cameras.

Each hidden state $X_{i,j}$ in the model has a state space $Z_{i,j}$ initialized to be Z, the set of 3D poses spanned by the training database D, where $|Z| = 233,280$. From here on out we will express the state space $Z_{i,j}$ by a set of training samples, and it is to be understood that the state space is equivalent to the 3D poses spanned by those samples. Additionally, we assume that the training samples and features Y are represented by the same feature and that a suitable distance measure is available to compare them.

Our goal is to use our observations (features) Y to infer the most likely set of hidden states (3D poses) X. However, two problems arise that prevent the direct use of this graphical model. First, the state space is quite large which is problematic when the number of frames n grows. Second, appearance information alone can be highly ambiguous; for example, a bat flying away from the camera may appear similar to a bat flying towards the camera. To deal with problems of search size and ambiguities, we propose two strategies to progressively shrink the state space. These strategies, detailed below, leverage the dataset dependent inputs along with the training database.

2.5. Rule-based Coarse Reduction of Search Space

To reduce the state space of each node $X_{i,j}$, we designed a rule-based system. The first rule takes advantage of the input 3D trajectory $\mathcal{T}$ to bias the state space of a hidden state $X_{i,j}$. The heading of a bat at frame number j is given by the vector $h_j = [\mathcal{T}_{j+1} - \mathcal{T}_j]$ for $j = 1$ and $h_j = [\mathcal{T}_j - \mathcal{T}_{j-1}]$ for $j > 1$. The first rule eliminates 3D poses that represent a bat flying in a direction very different than the heading h_j. The second rule, defined for frame numbers $j > 1$, biases the state space of a hidden state $X_{i,j}$ to be nearby in pose to those from the previous time $X_{i,j-1}$. The third rule eliminates upside down poses from the state space, since they do not occur in our datasets. The fourth rule reduces the state space to only have 3D poses which viewed from camera i look similar to $Y_{i,j}$. This represents the assumption that nearby poses in pose space appear similar in image space, and feature space, when viewed from the same camera.

The second rule requires that the state space is initialized well at frame one. To obtain a sufficiently accurate initialization, we use the first rule with the assumption that the back of the bat points as much towards the sky as possible while maintaining its heading. Application of the last rule reduces the state space to size k, by choosing the k samples with appearance closest to $Y_{i,j}$, as defined by a feature distance measure (details in Sect. 3.1). After application of the four rules to all frames, all hidden states $X_{i,j}$ will have a state space of size k.

2.6. Fine Reduction with a Markov Random Field

After our method has reduced the state space of each node to k candidates, it chooses the single best candidate for each node. To help guide the choice of best candidates, we propose two constraints. The first constraint is a temporal smoothness constraint which introduces a penalty when the

estimated 3D pose of a bat changes too much from one time frame to the next. The second constraint is a camera geometry constraint which favors 3D pose estimates that are consistent with the camera geometry. The undirected graphical model in Figure 4b models these added constraints; vertical edges in blue represent temporal constraints, and horizontal edges in red represent camera geometry constraints.

One way to find the most probable 3D poses for the hidden states in this undirected graphical model is to compute a maximum a posteriori (MAP) estimate. This can be formulated as finding the values of X that maximize $P(X|Y)$, where $P(X|Y) \propto P(Y|X)P(X)$. Here $P(X)$ is a Markov Random Field (MRF) prior, which encodes our two constraints, and $P(Y|X)$ is the likelihood that observations Y came from hidden states X. The forms we selected for the MRF prior and likelihood function depend on our design of the MRF model [5, 7], which we first introduce.

MRF Model. A Markov Random Field model is characterized by the Markov property which states that a node is conditionally independent of all other nodes in the graph given its neighbors. Additionally, a Markov Random Field model may be described by a joint distribution that factors in a special way. In particular, if C is a clique and the set of random variables belonging to that clique is x_C, then the joint distribution $P(x)$ factors as $P(x) = \frac{1}{Z}\prod_c \psi_C(x_C)$, where Z is a normalization constant, and $\psi_C(x_C)$ is a potential function defined on the maximal cliques of the graph. Maximal cliques in this work are of size 2, so potential functions are defined on pairs of hidden states connected by edges. We chose the potential function for temporal edges $e_t = (x_{i,j}, x_{i,j'})$ to be $\psi^T(e_t) = e^{-D_T(e_t)}$ and for camera edges $e_c = (x_{i,j}, x_{i',j})$ to be $\psi^C(e_c) = e^{-D_C(e_c)}$.

MRF Prior. Our choice of potential functions yields the prior $P(x) \propto e^{-(\sum_{e_t} D_T(e_t) + \sum_{e_c} D_C(e_c))}$, where the temporal distance D_T is chosen to reflect the change D_O in orientation over time and the change D_W not-consistent with a typical wing beat cycle. The sum $D_T(P_i, P_j) = D_O(P_i, P_j) + D_W(P_i, P_j, \theta)$ expresses this distance, where P_i and P_j are 3D poses consecutive in time and θ is a tuning parameter. If unit quaternion q_1 represents the first orientation, and unit quaternion q_2 the following, we define D_O by the angle between the two unit quaternions. The distance D_W between two wing configurations is given by a cost matrix that encodes a typical wing beat cycle by penalizing two wing configurations which either do not follow the correct order or are not nearby in the wing beat cycle. The distance D_C due to camera geometry is defined similarly to D_T, but with a different cost matrix that penalizes wing configurations proportional to their difference.

MRF Likelihood Function. The likelihood function $P(Y|X)$ is defined on pairs of variables connected by edges e_o. From conditional independence assumptions, the likelihood simplifies to $P(Y|X) = \prod_{v \in \mathcal{V}} P(Y_v|X_v)$. The individual likelihoods corresponding to each node are chosen as $P(Y_v|X_v) = e^{-D_f}$, where D_f is the difference between the feature representation of the observation and the feature representation of the view corresponding to the 3D pose X_v. The feature and feature distance used in our experiments are discussed in Section 3.1. Combining the likelihood and prior the posterior is given by $P(X|Y) \propto e^{-(\sum_{e_t} D_T(e_t) + \sum_{e_c} D_C(e_c) + \sum_{e_o} D_f(e_o))}$. Final 3D pose estimates are obtained using max-product loopy belief propagation (messages initialized to 0) to yield an approximate MAP estimate. Additional implementation details are available in [18, 22].

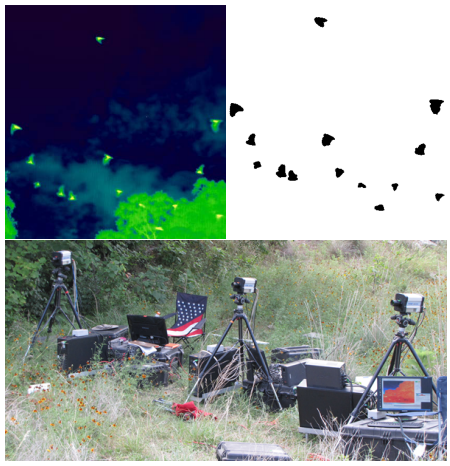


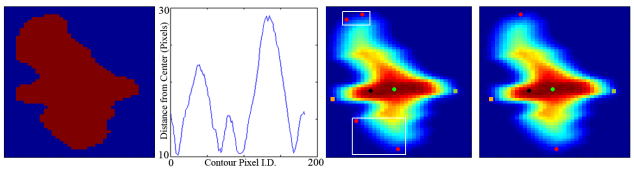
Figure 5. Sample data of bats in the wild. **Top:** A frame from an infrared video showing bats flying in the wild and the corresponding segmentation to the right. **Bottom:** A typical field setup of 3 cameras with baselines between 1 m and 2 m.

3. Experiments

Our experiments were based on videos taken of *Tadarida brasiliensis* emerging from a cave in Texas. Three FLIR SC8000 thermal infrared cameras were used to record data at a frame rate of 131.5 fps at a resolution of 1024×1024 pixels through a 25 mm lens. Typical camera baselines for field experiments were in the range of 1 to 2 m (Fig. 5 bottom). Segmentation of the bats from video was performed by modeling the background intensity with a single Gaussian component per pixel. Pixels with outlier intensities were labeled as belonging to a bat. An infrared image with its corresponding segmentation is shown in Fig. 5 top. After segmenting a bat, our method computes a corresponding part-based feature vector, as described in the next section.

3.1. Comparing Part-based Features

To reliably compare real thermal infrared images to training samples from our approximate 3D graphics model, we chose the "high level representation" of a part-based feature. It is an 8-dimensional vector that encodes the



(a) (b) (c) (d)

Figure 6. Part-based feature: (a) Segmented bat. (b) Boundary contour unwrapped into a 1D signal from which points of high curvature are extracted. (c) Overlaying the thermal view are clusters of wingtip points shown as red dots enclosed by white rectangles. The green dot is the body center. The 2D heading is depicted by a vector from the green dot to the black dot. (The head and tail of the bat are also detected but not used in this work.) (d) Non-maximum suppression yields one wingtip per wing.

overall structure of the bat body by storing the 2D position of the wingtips relative to the body center. The body center is approximated by the warmest pixel on the bat [16] and can easily be detected across poses. To detect wingtips, our method extracts the boundary contour of a bat and chooses the points with large curvature, characteristic of wingtips, as candidates. Candidates are then clustered and non-maximum suppression is used to select up to two wingtip positions. Our part-based feature is augmented with a 2D projection of the 3D heading of the bat, obtained from the 3D trajectory associated with that individual bat. The 3D trajectories were generated by using a Kalman filter to track the 2D position of a bat in each view. These 2D positions were combined with camera geometry information (relative rotation and translation) obtained from a DLT based calibration routine [9] to reconstruct a 3D trajectory. The process of feature extraction is shown in Fig. 6. It should be noted that the same feature representation is extracted for all training samples. In this case, the process is trivial since the color of the wingtips can be specified and the 3D heading of the model is always known.

To compare features, we chose a simple distance measure that combines a wingtip position distance d_{wt} with a 2D heading distance d_h. The distance between two wingtips is a combination of the difference in their angles (relative to the body center) and the difference in their distances from the body center. The 2D heading distance is the angle between the two vectors. When one of the features has a different number of wingtips detected than the other, a penalty is added for the mismatch. When two wingtips are detected in each feature, the resulting data association problem is solved by choosing the pair among the two possible pairs that minimizes the feature distance.

3.2. Experimental Methodology and Results

We designed an experiment where we compared the results of a baseline algorithm and the results of 3D-PEB to those obtained from manual annotations. The baseline algorithm B is derived from 3D-PEB by removing the first three rules from the coarse reduction stage of 3D-PEB and keeping the MRF model unmodified. In this experiment, the state space for each node was set to $k = 100$. For this value of k our Matlab implementation took ≈ 10 minutes to run, with most of the time spent on k-nearest neighbor (k-nn) retrieval, and computation of the potential functions. Relative to this, the time taken by the max-product algorithm to converge was negligible.

Table 1. E_o, E_{θ_1}, and E_{θ_2} represent differences in orientation and wing angles, computed by comparing baseline algorithm B and the proposed system 3D-PEB (noted by P), on two datasets with 183 and 93 frames (cn) respectively, with the gold standards established by two annotators, expert annotator A_1 and non-expert annotator A_2.

Id	# fr.	Alg.	Expert Ann. A_1			Annotator A_2		
			E_o	E_{θ_1}	E_{θ_2}	E_o	E_{θ_1}	E_{θ_2}
1	183	B	70°	45°	47°	51°	43°	53°
		P	17°	17°	14°	16°	16°	25°
2	93	B	59°	55°	53°	59°	63°	63°
		P	20°	22°	18°	21°	26°	26°

The experiment was designed to compare the automatic 3D pose estimates from 3D-PEB to a gold standard and thus examine the accuracy of our proposed algorithm. To obtain quantitative gold standard annotations of real image data, we built an annotation tool and asked two annotators to label 3D pose in an image of a bat by manipulating the 3D pose of our bat model until it matched a presented image. All controls on the annotation tool were discretized to 5° increments to make annotation less labor intensive. Annotator A_1 was an expert bat biologist, annotator A_2 a graduate student with less expertise.

As a measure of potential inaccuracies in the automated estimation of 3D pose, we report the differences in orientation and wing joint angles between automatically-produced estimates and those that were based on manual annotations. If unit quaternion q_e is the estimated orientation and q_a the gold-standard orientation, then the difference E_o is defined by the angle, in degrees, between the two unit quaternions. The maximum possible difference in orientation is 90°. The differences in the wing angles are defined by the difference $E_{\theta_1} = |\theta_{1e} - \theta_{1a}|$ in the elbow angle and the difference $E_{\theta_2} = |\theta_{2e} - \theta_{2a}|$ in the wrist angle. The maximum possible difference in wrist or elbow angle is 180°.

Experimental results are based on estimating the 3D pose of 276 frames, taken from 2 different videos, representing the 3D pose of several bats across time and cameras. The first video (Id 1), is a bat emergence where the 3D pose of bats change steadily over time. The second video, (Id 2), is a more chaotic bat emergence where the bats are changing their 3D pose rapidly.

Quantitative results are summarized in Table 1. Our pro-

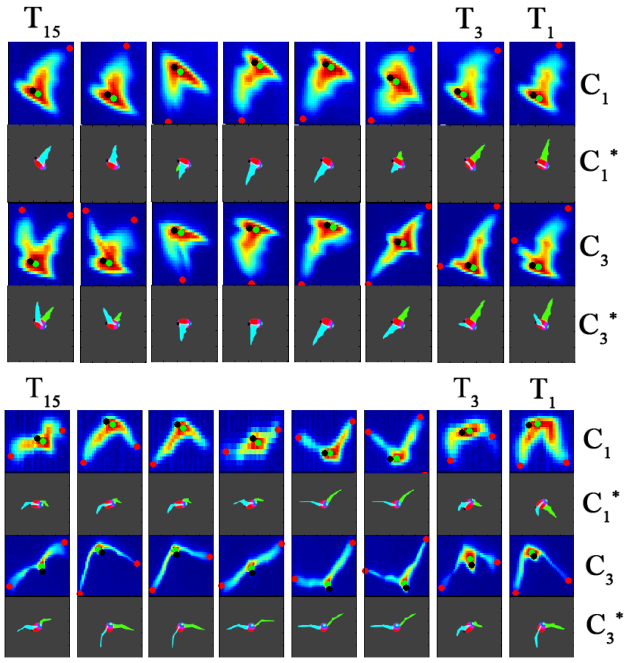


Figure 7. **Top:** The rows labeled C_1 and C_3 show frames of a flight sequence from video 1, seen from camera 1 and 3, respectively, with the part-based feature of the bat overlaid. The 3D pose estimates produced by 3D-PEB are shown in rows C_1^* and C_3^*. From right to left, every other frame is shown ($T_1 - T_{15}$). **Bottom:** Estimates for a flight sequence from video 2.

posed algorithm yielded a mean difference in orientation in the range of $16^\circ - 21^\circ$, or $\approx$ $17\% - 23\%$, considering the maximum of 90° difference to be 100%. For the wing angles, the mean differences range from $14^\circ - 26^\circ$, which is $\approx$ $7\% - 14\%$. The baseline algorithm has a significantly higher difference on average, with the mean orientation difference in the range of $51^\circ - 70^\circ$, or $\approx$ $56\% - 77\%$, and mean wing angle difference in the range of $35^\circ - 71^\circ$, which is $\approx 25\% - 35\%$.

In addition to quantitative results, we provide qualitative results in Fig. 7. The 3D pose estimates our system produced for 2 particular bats, selected from two different videos, are shown on even numbered rows and can be compared visually to the 3D poses observed in the input, shown on odd numbered rows. Input thermal infrared images are at their original resolution. In the model, recall that the left wing is colored blue and the right wing is colored green, with the head of the bat colored black.

4. Discussion

Our results are the product of many system components and reflect challenges in both the general problem of 3D pose estimation and those specific to bats and the datasets used for our experiments.

A general 3D pose estimation method must find a way to deal with ambiguities arising when multiple 3D poses map to nearly identical projections. In our work, ambiguities are dealt with by filtering out upside down poses and 3D poses that disagree with the heading of the bat. The errors obtained by baseline algorithm B help validate the need for these particular rules. The baseline algorithm performs significantly (≈ 3 times) worse than 3D-PEB because, without filtering, many incorrect 3D poses remain in the state space, and the MRF model is forced to find the best solution among mostly incorrect candidates.

Many of our design decisions have been influenced by the application domain of bats. Bats in the wild fly quickly, so keeping them in the field of view for a reasonable amount of time requires placing the cameras at a distance from their flight paths, yielding relatively low spatial and temporal recording resolutions. At a spatial resolution of approximately 30 by 30 pixels per bat, its overall body shape can be seen with some coarse detail on the wings, head, and tail (Fig 7). At a frame rate of 131.5 fps, the wingbeat cycle can be seen in a choppy fashion. This level of detail in the input video influenced both the 3D bat model we developed and the features we used.

Consequently, a fundamental challenge has been to compare images generated from our 3D bat model with images generated from a real, fast moving, articulated bat. We initially chose low level features based on moments, and contours, but these failed to capture semantic similarity across these two domains. We observed this failure when a k-nn retrieval produced few good matches and many bad ones. Our part-based feature, which captures the overall body and wing structure, was more successful at matching across domains. Detecting the body parts of bats in sequences of low resolution images is a largely unexplored problem and solutions here could accelerate progress in this application domain. More generally, accurate graphics models may not be available for a variety of animals and other researchers may find part-based features beneficial for relating projections from approximate graphics models to real image data.

Explicit experiments were not carried out to determine robustness of our estimates with respect to input segmentations. Segmentation quality was not a major concern for the datasets used because the background was mostly sky and slow moving.

The frames chosen from both videos, for evaluation, are typical of the thousands of other frames in those videos. The bats sampled for our experiments, span a variety of 3D poses, including less complex and more complex ones.

The evaluation of our method relies on the labor-intensive process of obtaining a gold standard. A domain expert was required, who had limited time, making a larger-scale experiment difficult to carry out. This hurdle should be considered by anyone wishing to analyze video of wild

animals, whose motions and articulations are complex and not amenable to typical marker based motion capture approaches.

5. Conclusion

The goal of our work has been to design and build a system which can automatically estimate the articulated 3D pose of bats flying in the wild. In the context of the challenges discussed, as well as the novelty of our results, we find our automated estimates more than suitable for describing coarse 3D pose of flying bats in the wild. Our paper describes both the process and the considerations taken in developing a system for estimating the 3D pose of a wild animal. We are offering a 3D graphics model of *Tadarida brasiliensis* for public use, as well as calibrated images of bats, and encourage others to experiment with bat images. This is also the first time in the literature, quantitative and qualitative results are shown for the automatic 3D pose estimation of bats in the wild.

Future work includes looking at how sensitive our system is to errors in camera calibration, and experimenting with different potential functions. We intend to use our system to analyze interesting maneuvers performed by bats. Collaborations with domain experts may lead to applying more detailed 3D bat models on higher quality bat videos, and uncovering new patterns in 3D pose that yield insight into the obstacle avoidance and foraging behaviors of bats.

Acknowledgments. This material is based in part upon work supported by the Air Force Office of Scientific Research, the National Science Foundation (1337866, 0910908, 0855065), and the Office of Naval Research (N00014-10-100952). We would like to thank Jaclyn Aliperti, Ali Irwin, Tom Kunz, Amy Norris, Leslie Pepin, Kenneth Sebesta, and Diane H. Theriault for their help with field experiments, and the Bamberger Ranch Preserve for site access. We thank the Image and Video Computing group at BU for their valuable feedback.

References

[1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *PAMI*, 28(1):44–58, 2006. 2

[2] V. Athitsos and S. Sclaroff. 3d hand pose estimation by finding appearance-based matches in a large database of training views. In *IEEE Workshop Cues in Communication*, 2001. 2

[3] A. J. Bergou, S. Swartz, K. Breuer, and G. Taubin. 3d reconstruction of bat flight kinematics from sparse multiple views. In *IEEE Workshop on Dynamic Shape Capture and Analysis*, pages 1618–1625, 2011. 1

[4] J. M. Birch. Comparing wing shape of bats: The merits of principal-components analysis and relative-warp analysis. *J. Mammal.*, 78(4):pp. 1187–1198, 1997. 3

[5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006. 5

[6] Blender. http://www.blender.org. 3

[7] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *CVPR*, 1998. 5

[8] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *CVIU*, 108:52–73, 2007. 2

[9] T. L. Hedrick. Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. *Bioinspir. Biomim.*, 3, 2008. 3, 6

[10] M. Hofmann and D. Gavrila. Multi-view 3d human pose estimation in complex environment. *IJCV*, 96(1):103–124, 2012. 2

[11] T. Y. Hubel, N. I. Hristov, S. M. Swartz, and K. S. Breuer. Changes in kinematics and aerodynamics over a range of speeds in *Tadarida brasiliensis*, the Brazilian free-tailed bat. *J. R. Soc. Interface*, 2012. 1, 3

[12] J. J. Kuffner. Effective sampling and distance metrics for 3D rigid body path planning. In *IEEE ICRA*, pages 3993–3998, 2004. 4

[13] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81(3):231 – 268, 2001. 2

[14] T. B. Moeslund, A. Hilton, and V. Krger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(23):90 – 126, 2006. 2

[15] E. Murphy-Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *PAMI*, 31:607–626, 2009. 2

[16] J. D. Reichard and S. R. Fellows. Thermoregulation during flight: Body temperature and sensible heat transfer in free-ranging Brazilian free-tailed bats (*Tadarida brasiliensis*). *Physiol. Biochem. Zool.*, 83(6):pp. 885–897, 2010. 6

[17] D. K. Riskin, J. Iriarte-Daz, K. M. Middleton, K. S. Breuer, and S. M. Swartz. The effect of body size on the wing movements of pteropodid bats, with insights into thrust and lift production. *J. Exp. Biol.*, 213(23):4110–4122, 2010. 1

[18] M. Schmidt. Ugm: Matlab code for undirected graphical models. http://www.di.ens.fr/~mschmidt/Software/UGM.html, 2013. 5

[19] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, Jan. 2013. 2

[20] D. H. Theriault, Z. Wu, N. I. Hristov, S. M. Swartz, K. S. Breuer, T. H. Kunz, and M. Betke. Reconstruction and analysis of 3D trajectories of Brazilian free-tailed bats in flight. Technical Report BUCS-2010-027, 2010. 1, 3

[21] P. Watts, E. J. Mitchell, and S. M. Swartz. A computational model for estimating the mechanics of horizontal flapping flight in bats: Model description and validation. *J. Exp. Biol.*, 204(16):2873–2898, 2001. 3

[22] Y. Weiss. *Comparing the mean field method and belief propagation for approximate inference in MRFs*. The MIT Press, 2001. 5

[23] Z. Wu, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke. Tracking a large number of objects from multiple views. In *ICCV*, 2009. 8 pp. 1, 3