3D Reconstruction of Bat Flight Kinematics from Sparse Multiple Views

Attila J. Bergou Sharon Swartz Kenneth Breuer Gabriel Taubin
Brown University
Providence RI, USA
{attila_bergou,sharon_swartz,kenneth_breuer,gabriel_taubin}@brown.edu

Abstract

In this paper we present a novel method to reconstruct the 3D posture of flying bats from sparse multiple view video. Specifically, we incorporate biomechanical and geometric knowledge about bats into an articulated model. We then estimate the bats time-varying pose by tracking a set of known markers using a Square Root Unscented Kalman filtering method augmented with video optical flow information. Our method scales easily to multiple views, elegantly handles missing and occluded markers, and has a versatility in the type and complexity of the tracking model. To demonstrate the performance of the reconstruction method, we apply our technique to estimate the parameters of a 52 degree of freedom articulated model of a bat from a real-world flight sequence. We further assess our algorithms performance by quantifying its ability to recover model parameters accurately for a realistic simulated flight sequence.

1. Introduction

Motion is ubiquitous in Biology: all living organisms exhibit motions at some level from the cellular to the macroscopic. For animals in particular, motion determines how they forage, migrate, mate, where they live and how they escape predators. Because of the biological significance of animal movement and the increased availability of high quality cameras for motion capture, there is growing interest in quantifying animal locomotion from multiple view video. Despite this interest, 3D motion reconstruction methods predominantly focus on the capture and reconstruction of human motion. As such, there are fewer sophisticated methods for analysis of animal locomotion.

In this paper, we detail a method to reconstruct the flight kinematics of freely flying bats from multiple view video. The genus of bats, containing 25% of all mammals [30], has recently garnered interest from both the engineering and biological communities because of the exceptional flight characteristics these animals demonstrate. Bats are adept at maneuvering and at the same time are highly efficient fliers

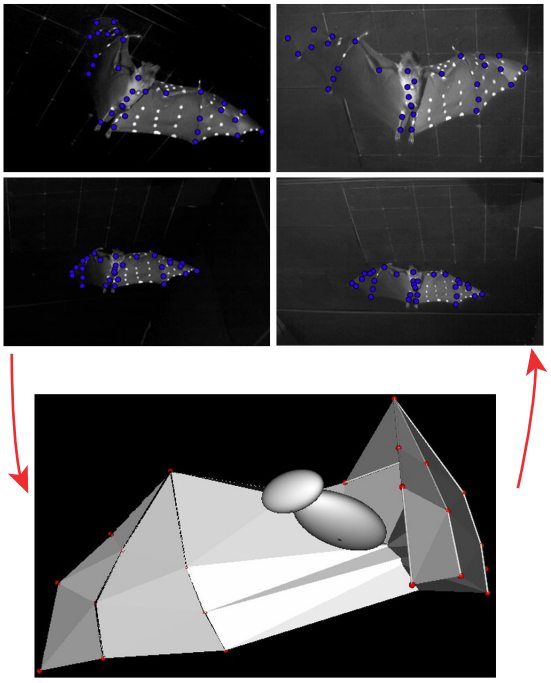

Figure 1. We develop a model based method to track the highly articulated flight of bats from sparse multiple view video.

[24]. These flight abilities largely stem from bats' precise ability to control their highly articulated wings. The study of the flying capabilities of these animals as a consequence requires accurate reconstruction of their wing and body kinematics. The biomechanics of bat wings, however, makes their 3D motion reconstruction particularly challenging: the elaborate wing beats of bats result from the coordination of more than 24 wing joints as well as the deformation of a thin membrane covering their wing skeleton [24].

Previous investigations of bat flight kinematics have relied on hand digitizing video markers and direct triangulation of their 3D positions (e.g. [31, 27]). Due to the relatively sparse camera arrays used to reconstruct these kinematics and the large scale motions that bat wings ex-

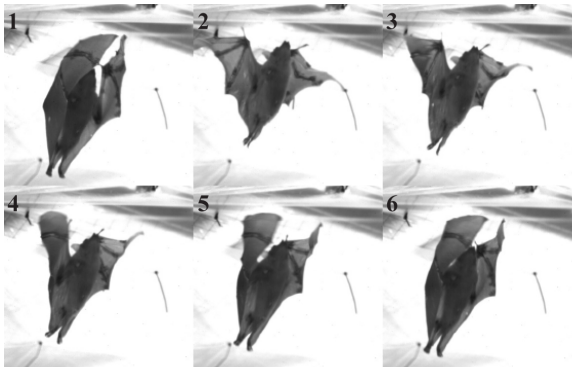

Figure 2. A landing sequence of a fruit bat, *Cynopterus Brachyotis*.

hibit in-flight, the reconstructed kinematic data often contain gaps where insufficient information exists to directly triangulate marker positions. For example, in Figure 2, we show an image sequence by [26] depicting a landing fruit bat, *Cynopterus Brachyotis*. The complex wing motion results in significant self-occlusion that makes direct tracking of marker locations tedious and direct triangulation of marker locations often impossible.

The wings of bats are homologous to human arms, hands, and legs: most joints present in human appendages are also present in the wings of bats. As such, there is a close relationship between the tracking of flight kinematics for bats and the tracking of whole body kinematics for humans. We therefore build upon techniques developed for tracking whole body articulated motion of humans to track the highly articulated body and wing motion of bats. The approaches to human motion tracking generally fall into two categories: model based methods and model free methods [25, 23, 13]. Model free methods (e.g. [1, 6, 8, 12, 15, 2]) generally rely upon learning associations between captured images and a large database of predefined motions. In practice, these methods perform very well: for example accurately recovering more than 50 degrees of freedom in real world tests [1]. For bats, however, no kinematic database that contains a sufficient sampling of motions exists, nor can feasibly be generated. Therefore such methods cannot be used to track bat flight kinematics.

A significant body of work exists on model based methods to track articulated kinematics. These methods are either bottom-up (e.g. [5, 16, 21]) and directly identify body parts in images to infer 3D posture or top-down (e.g. [10, 14, 7, 18]) and projected model information into the image space to infer 3D model posture from the projection disparity. Bottom-up approaches are precluded for the reconstruction of bat flight kinematics because the membrane covering bat wings does not allow enough information to be gained directly from images. For top-down approaches, stochastic methods such as the Kalman filtering (e.g. [9, 29]) or particle filtering (e.g. [11, 19, 32]) are often used to approximate the distribution of model states.

Our method of tracking the in-flight posture of bats is a top-down model based approach built upon the Square Root Unscented Kalman Filter [22]. The primary contribution of this work is the development of a practical and versatile tracking framework to estimate the posture of the highly articulated flight kinematics of bats. Tracking is accomplished through the extraction of a sparse set of known image features and the Square Root Unscented Kalman Filtering augmented by optical flow to recover 3D body pose. To the best of our knowledge, this paper presents the first use of the square root unscented Kalman filtering method for a complex tracking application. In addition, we believe the manner we integrate feature tracking with optical flow with Kalman filtering is unique. The fusion of these two methods is superior to either method alone. Our method demonstrates the feasibility of tracking highly articulated motions by detecting only a sparse set of image features. Because our approach is quite general and scales to models with a large number of degrees of freedom, we believe that it can be widely applied to study animal and human locomotion.

2. Filming and Experimental Setup

The flight sequences in this paper are filmed using the methods described in [26, 27]. The experiments were done in an enclosed flight corridor with several species of bats. The flight sequence that we analyze in this paper was filmed using four calibrated Photron FASTCAM 1024 PCI cameras running at 1000 frames per second. At this frame rate, these cameras record 1024 x 1024 pixel 8-bit grayscale images. The high speed of the cameras makes experimental lighting difficult. In turn, the lack of scene lighting makes detection of natural markers on the bats skin difficult. Therefore, to aid tracking, the wings of bats were temporarily marked prior to flights. We make use of a subset of 32 markers that are attached to the bat. Two of these markers are placed at the center of the bats body and denote the longitudinal axis of its body. The other 30 markers are split evenly between the bats two wings and placed at the bats wing joints and wing tips. The markers are shown in the images of the bat in Fig. 1 by the blue dots and also in Fig. 4 by the red and blue circles. The marker placement is chosen to uniquely specify the state of the articulated bat model in §3.1.

We note that although the capture of the bat flight sequences are performed in real-time, the filming equipment necessarily restricts processing to be performed off-line. Because the processing of these videos is not time-critical a degree of human interaction in tracking is acceptable.

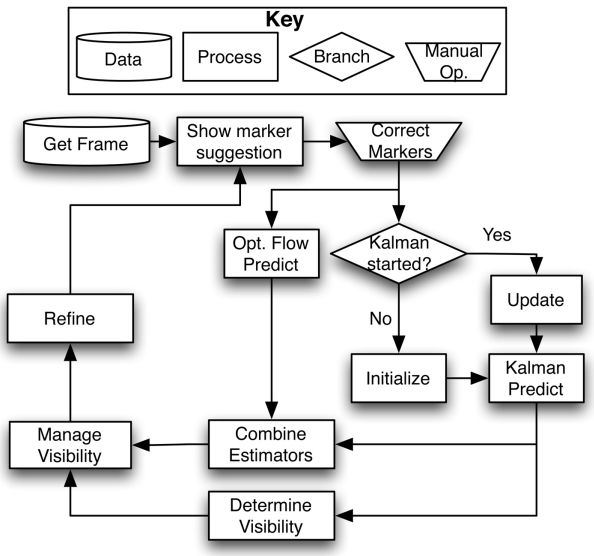

Figure 3. High-level flowchart describing tracking method.

3. Description of Method

To reconstruct the wing and body kinematics of bats, we developed a versatile framework where known biomechanical constraints can be incorporated into tracking. We anticipate that our tracking method will be used to reconstruct the flight kinematics of a variety of bat species. Consequently, our model-based tracking framework accommodates generalized articulated tracking models. We only consider estimating flight kinematics from multiple views. Accordingly, we require a minimum of two cameras for tracking.

We phrase tracking as a non-linear state estimation problem. Namely, we estimate the state of an articulated bat model by measuring the locations of visible named markers in a sequence of images from a number of calibrated cameras. We consider a model parametrized by body position, orientation, wing joint angles, and geometric parameters specifying the dimensions of the model. In what follows, we refer to model state - i.e. the stacked vector of orientation and dimensional parameters - by the variable q. We refer to measurements - i.e. the stacked vector of image marker coordinates - by the variable x. We assume that the systems dynamics $q_{n+1} = f(q_n, w)$ and the mapping from state vector to measurement vector $x_n = h(q_n, v)$ can be approximated by a known stochastic functions f and h, respectively. In what follows w and v are assumed to be zero mean Gaussian random variables with known covariance. To track, we estimate q_n for each time-step n given measurement x_n and prior measurements.

To estimate the time-varying probability distribution (pdf) of q, we use the infrastructure provided by the Square

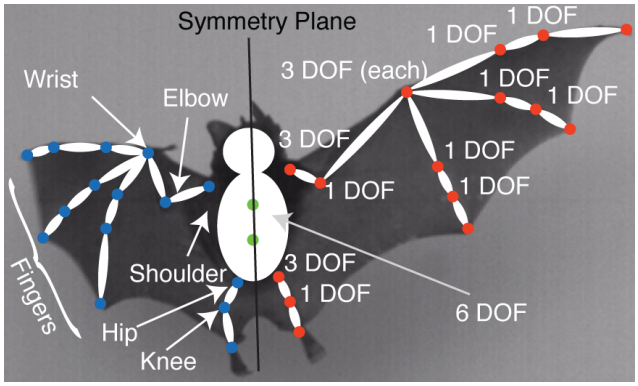

Figure 4. Skeletal model of a bat with 52 degrees of freedom.

Root Unscented Kalman filtering (SRUKF) [22]. Like the Extended Kalman (EKF) and Unscented Kalman filtering (UKF) methods [28], the SRUKF method generalizes the Kalman filter state estimation method to nonlinear problems. The EKF, UKF, and SRUKF all estimate the state pdf by estimating its time-varying mean and covariance. The EKF applies the Kalman filtering framework by directly linearizing the dynamics and measurement equation about the current state estimates. The UKF and SRUKF use a sample of generated sigma-points to statistically linearize f and h. This latter method has been shown to produce superior accuracy in estimating state covariances to the direct linearization methods [28]. Whereas UKF and EKF both propagate the state pdf using a state mean and covariance matrix, SRUKF propagates the state mean and the Cholesky decomposition of the covariance, enhancing both the efficiency and the numerical properties of the method.

The high level flow chart for the tracking of bat flight kinematics is shown in Figure 3. We explain the components of of methods in §3.1 – §3.7.

3.1. Tracking Model

Our tracking method is generalized to work with a variety of articulated models. For the results presented in this paper, we use the bat model shown in Fig. 4. The "generic" bat is modeled with a combined 52 degrees of freedom (dof) and optional equality and inequality constraints on joint angles. We model the bat's body as rigid with 6 dofs. The wing skeleton is modeled as 13 bones connected by 13 joints. Each bone's coordinate frame is specified relative to the previous bone with the rotational dofs specified in Fig. 4. We do not explicitly model the bats wrist, which is very small, and instead attach the fingers directly to the bats forearm. In addition to the above state variables, our model contains geometric parameters that specify the lengths of the bat bones and geometry of the body. The geometry of the model is thus determined during tracking.

The bat's pose is specified using a kinematic chain spec-

ified by an exponential map formulation [7, 5]. The transformation matrix, g_i, specifying the orientation of bone i relative to the body is therefore formed recursively by

$$g_i = \prod_{j \in \lambda(i)} g_j \tag{1}$$

where g_j's are the transformation matrices of bones further up the kinematic chain specified by $\lambda(j)$. The form of the exponential map we use makes the wing bone angles, and body orientation angles equivalent to Euler angles.

To integrate with our framework, our tracking model also specifies: 1. The process noise and nonlinear measurement noise covariance matrices; 2. A - possibly nonlinear - function that specifies how the model state evolves between time-steps; and 3. A - possibly nonlinear - function that specifies the mapping from the model state to measurements, which are in this case the markers detected in each video frame. These are discussed in §3.3.1.

3.2. Initialization Step

Model initialization is performed interactively. Through a graphical interface the user estimates initial marker locations. The best estimate of each marker location is specified by clicking its corresponding image location. These user estimates are then refined by the algorithm by: 1. Constraining corresponding measurements between different camera views to lie along epipoles 2. Refining user click locations to the nearest SURF or Harris image feature [3, 17]. The user can optionally adjust the covariance matrix for uncertain measurements by dragging an ellipse around the specified marker mean to define the estimated measurement spread, though this is not required. This interaction results in a vector of initial image coordinates x and a corresponding (diagonal) covariance matrix M_x. We use x and M_x to numerically solve the constrained minimization

$$\underset{\tilde{q}}{\operatorname{argmin}}\ \tilde{E}(\tilde{q}, x) \equiv \underset{q,\lambda}{\operatorname{argmin}}\ \big(E(q, x) + \lambda^T C(q)\big) \tag{2}$$

for the most probable model state vector q. In the above equation λ is the vector of Lagrange multipliers, $\tilde{q} = [q^T, \lambda^T]^T$ is the augmented state vector, and $C(q)$ is the vector function of model state constraints. For the bat model described in §3.1, $C(q)$ is the single quaternion normalization constraint for the bats body degrees of freedom, while the energy function

$$E(q, x) = (x - P(q))^T M_x^{-1} (x - P(q)), \tag{3}$$

is the Mahalanobis distance between the estimated distribution of marker locations and the image space projection $P(q)$ of the model marker locations for a particular configuration q.

Once the minimization has converged to a particular model state q^*, we propagate the uncertainty from the initial image marker coordinates to state space. We treat the minimization in Eq. 2 as an implicit function

$$f(x) = q, \tag{4}$$

that maps from image marker space to state space. Then we propagate uncertainty using the linearized approximation

$$M_q \approx \nabla_q f^T M_x \nabla_q f \mid_{q=q^*} \tag{5}$$

where we evaluate the Jacobian $\nabla_q f$ implicitly

$$\nabla_{\tilde{q}} f = -H_{\tilde{q}}(E(\tilde{q}, x))^{-1} \nabla_{\tilde{q}} \nabla_x E(\tilde{q}, x), \tag{6}$$

using the hessian $H_{\tilde{q}} E$ with respect to the augmented state, and the cross-derivative $\nabla_{\tilde{q}} \nabla_x E$.

3.3. Prediction Step

To determine the apriori state estimate q_{n+1}^-, corresponding apriori measurement x_{n+1}^-, and corresponding covariance matrices from the posteriori state estimate q_n^+ at time-step n we perform two distinct forms of predictions. The first prediction is the standard SRUKF prediction step and is described in §3.3.1. The second prediction is formed from a Lukas Kanade feature tracker and is described in §3.3.1. The latter method only results in predictions for x_{n+1}^-. In §3.3.3 we describe how these predictions are combined to form an improved state estimate.

3.3.1 Kalman Filter Prediction

We estimate the dynamics of the bat's state by an acceleration level model

$$\begin{aligned} q_{n+1}^- &= q_n^+ + u_n^+ \Delta t + a\frac{(\Delta t)^2}{2} \\ u_{n+1}^- &= u_n^+ + a\Delta t, \end{aligned} \tag{7}$$

similar to [9]. We thus estimate the articulated model's velocity in addition to the state. We model the acceleration a of joint angles a as a zero-mean gaussian random variable with covariance matrix M_a that is estimated from 135 flight sequences published in [27]. We have also generalized Eq. 7 to include higher derivatives at the cost of additional state variables. To estimate dimensional parameters of the model, which remain constant in time, we use $q_{n+1}^- = q_n^+ + n$ where n is a small variance gaussian random variable [28].

To map from model state to projected image space measurements, we first compute the 3D locations of the labeled marker positions (red and blue dots in Fig. 4) using the kinematic chain transformation matrices in Eq. 1. We then use the camera projection matrices to project 3D marker locations to image space (see §3.3.1). Specifically we implement this method using the function h

$$x_{n+1}^- = h(q_{n+1}^-, v_n^{(1)}) + v_n^{(2)}, \tag{8}$$

where h(q) is the noise free projection of a model state to image markers, and $v_n^{(1)}$, $v_n^{(2)}$ are modeled nonlinear and additive noise terms, respectively. The additive noise term is used to model uncertainty of locating the image space locations of markers. During the prediction step, these are estimated from the covariance matrix of measurements in the previous time-step. During the update step they are recomputed using the current state covariance. Additional measurement uncertainty, for example directly modeling the movement of markers relative to the bat, can be modeled using the nonlinear noise term.

3.3.2 Optical Flow Prediction

To complement the Kalman prediction step, we use the Lukas Kanade feature tracker [4] to provide a separate estimate of the movement of feature points in the image. To predict the location of features at timestep $n+1$, we first initialize the Lukas Kanade tracker with the location of defined feature points at time-step n and apply the algorithm. Broadly, this algorithm uses the optical flow in the image to predict the movement of specified feature points. For features that are successfully found at timestep $n+1$ we set the prediction mean μ to be the recovered location. We assign μ a diagonal covariance M_μ with a pre-configured variance. The value of M_μ determines the relative weight of the Kalman prediction and the optical flow prediction in the combined prediction.

3.3.3 Combining Estimators

Our algorithm contains two different measurement predictions: the Kalman prediction step (see §3.3.1) and the Lukas Kanade feature tracker (see §3.3.2). To combine these two predictions, we treat them as unbiased estimators and propagate their optimal linear combination [28]. This is equivalent to convolving the pdf of the two predictions. Specifically, for the two estimators with respective mean and covariance μ_1, M_1 and μ_2, M_2 their linear combination yields

$$\begin{aligned} \mu_* &= M_2 M_{1+2}^{-1} \mu_1 + M_1 M_{1+2}^{-1} \mu_2 \\ M_* &= M_1 - M_1 M_{1+2}^{-1} M_1, \end{aligned} \tag{9}$$

where μ_* and M_* are the mean and covariance of the combined estimator, and $M_{1+2} = M_1 + M_2$. When either the Kalman measurement prediction or the optical flow prediction contain missing values, we treat these columns as having infinite covariance. In effect this means that if an element is missing in either measurement vector μ_1 or μ_2 then the combined estimate yields the value of the other. When an element is missing in both μ_1 and μ_2 it is also missing in their combined estimate.

3.4. Managing Marker Visibility

After the prediction step the estimated image space positions of markers are ensured to be visible. This is done by directly checking whether the markers fall within view of the cameras, as well as checking whether the ray coming from the camera to the marker intersects a simple mesh model of the bat at any other position other than the marker. When the estimated position of a marker is determined not to be visible it is marked as such.

3.5. Measurement and Refine Step

For the refine step an automated form of marker measurement is performed. For each image space marker location that is predicted to be visible in a camera view we determine the closest SURF [3] or Harris corner [17] feature. In practice, we find that Harris features generally work well and are the features used for this presentation. When features are not found sufficiently close to the prediction, or two features are too close together they are marked as missing. When features are found, we assign the feature location as the mean of the feature distribution and assign a user specified covariance to it.

3.6. User Interaction

Once a prediction and refinement step is complete, the user is presented a view of the current frames from the calibrated camera sequence. Markers are shown as ellipsoids with the estimated mean and covariance each marker distribution. Markers that are missing are not shown. The user is then able to edit the location and estimated distribution by selecting which marker to edit and clicking on a specific image. We rerun the refinement step on user specified markers to estimate the nearest corners. The level of interaction depends heavily on bat wing motions and camera placement, however, interaction on average interaction is necessary every 5 or more frames and often no interaction is necessary for dozens of consecutive frames.

3.7. Update Step

We perform the standard SRUKF update step by first regenerating apriori measurement estimates using the process noise specified by measurement covariances of the current measurement step. This ensures that the Kalman update is performed with the proper weight afforded to measurements. Missing measurements are treated as having infinite covariance along the unknown measurement direction. Once apriori state and measurement estimates are available, we update the apriori state q_{n+1}^- using the Kalman update to generate the posteriori state estimate q_{n+1}^+. Constraints are then enforced by projection [20].

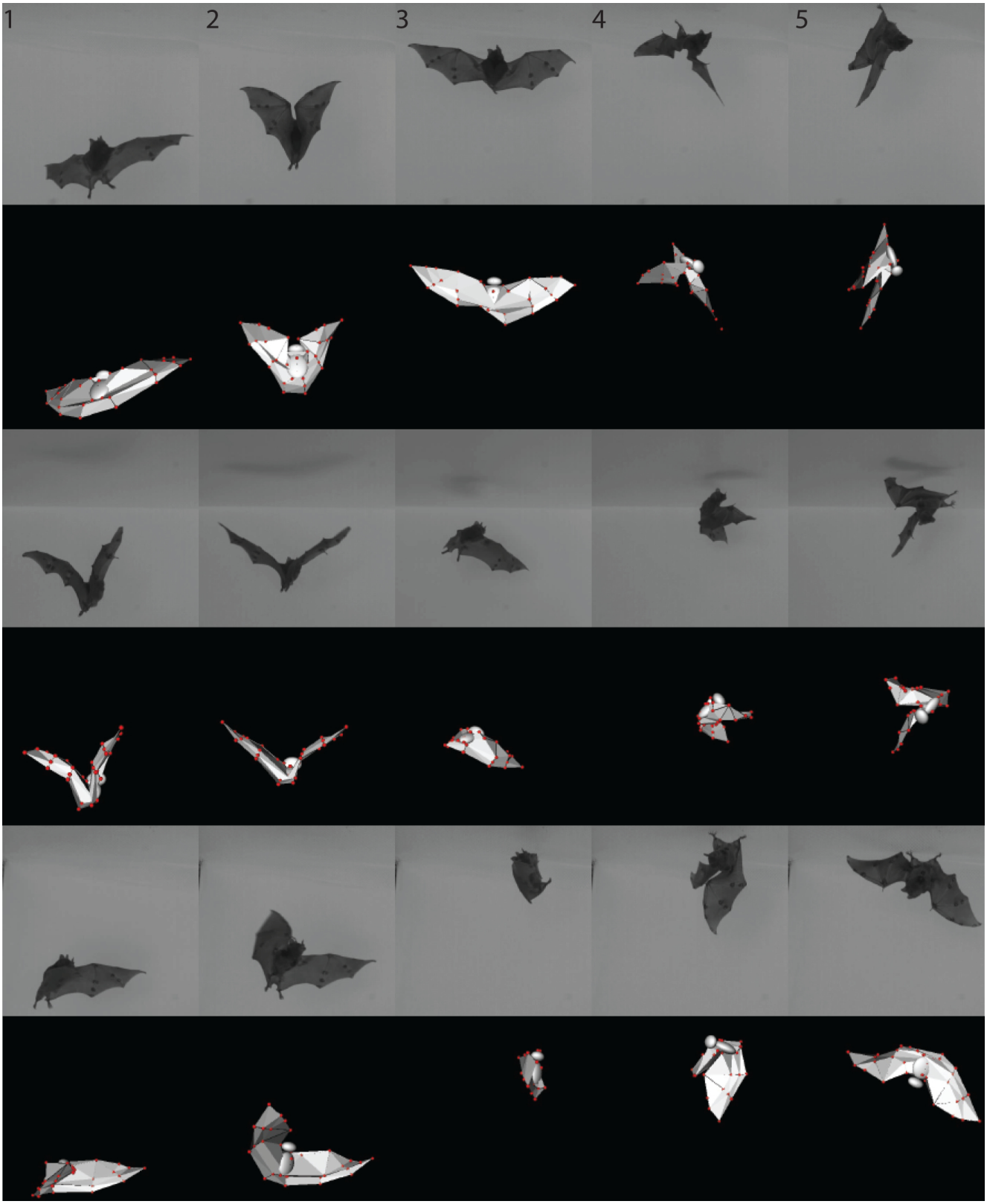

Figure 6. Real-world tracking of bats performing complex aerial rotations analyzed using the 52 degree of freedom articulated bat model.

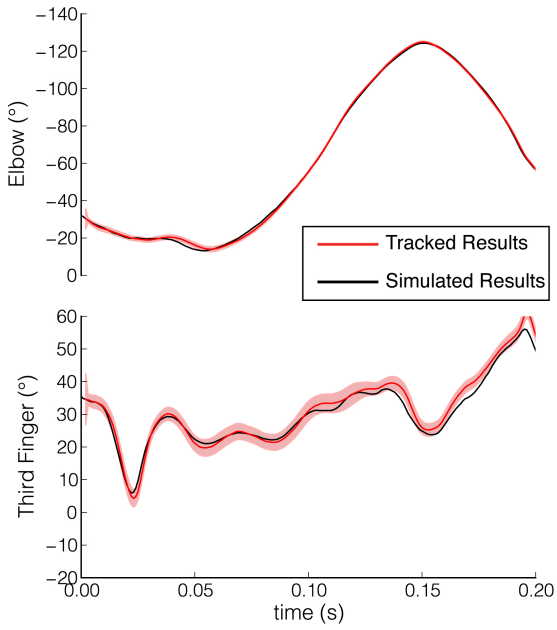

Figure 5. Evaluation of tracker performance on a simulated wing beat for the right wing elbow and distal joint of the third finger.

4. Results

No database of ground truth bat flight kinematics exists. Therefore our tracking method cannot be evaluated in comparison to known ground truth. Furthermore, previous attempts at tracking and analyzing the complex motions of these animals have focused on directly working with image space or 3D marker positions. Therefore, comparisons to previous work are not applicable for this study. To evaluate the performance of our tracking method given these restrictions, we show the results of applying the method to both real world and simulated bat.

To evaluate the algorithm's performance on simulated wing beat kinematics, we first simulate a 200 time-step realistic bat wing-beat on the articulated bat model shown in Fig. 4. We then compute 3D marker locations and project these locations into image space. The motion of the simulated bat is then reconstructed using our tracking algorithm from these image space markers. In Fig. 5 we show a comparison of the simulated (black line) and the reconstructed (red area) joint angle kinematics for the elbow and the distal finger joint. The area around the wing angle kinematic represent the standard deviation of the recovered wing angle kinematics as propagated through our algorithm. We find that the wing angle kinematics of both proximal and distal joint angles are recovered to better than 3° with only slight degradation in the tracking quality of the distal joint compared to the proximal one. Because the simulated kinematics represent - an admittedly very idealized, ground truth - these results demonstrate given high enough quality measurements, tracking converges to a true representation of the wing kinematics.

To evaluate the algorithms on real world performance wing kinematics we have tracked more than a dozen of flight sequences of 500 or more frames each using models of varying complexity. We show the results of tracking three complex flight sequences in Fig. 6. Five representative frames are chosen from each flight sequence. The quality of posture recovery is evaluated by first mapping the posture onto an articulated model of the bat, and then rendering this model from a similar view as the original frame. We find that although there is significant marker occlusion in all cameras throughout the tracking sequence, the algorithm can reconstruct the complex aerial rotations and highly articulated wing movements of these bats so that the rendered view compares favorably with the original frames. Videos of these, and other, flight sequence tracking example can be found in the supplementary materials.

5. Future Work

Our current implementation of the bat tracking system is based on a Kalman filtering formalism. As such, we implicitly demand that the probability distribution describing the state of our system is unimodal and can be approximated as a gaussian. Both of these assumptions can be violated during the tracking of an articulated model. In practice, violation of distribution unimodality has the largest potential to cause tracking errors. Namely, when the motions of the bats are highly occluded in the different camera views there may be a multiple model states that are equally likely. In this case, the Kalman filter will converge to one of the state and information is lost about the distribution. As such, a fruitful extension to this tracking system is to transition state propagation to a more sophisticated method based on particle filtering such as Condensation [19].

We are also currently working on extending the current marker based tracking system to allow for the detection and incorporation of additional features. The most promising line of inquiry is the incorporation of silhouettes into tracking. With this addition tracking is anticipated to become more robust during highly occluded portions of flight sequences as the shadows significantly restrict the possible joint angle configurations.

6. Conclusions

This paper presents a novel method to track the highly articulated kinematics of bats from multiple view video. We show that using a Square Root Unscented Kalman Filtering method augmented with a Lucas Kanade optical feature tracker, we are able to use a sparse set of robust feature points to accurately reconstruct the 3D posture of bats. Al-

though we gear our method towards the tracking of bat flight kinematics, our methods are not tightly coupled to the type of model being tracked and should therefore be widely applicable. Adding to the versatility of the method is its ability to easily base tracking off of any number of cameras.

The authors wish to thank D. Riskin, L. Reimnitz, J. Cheney, and C. Schunk for help with experiments and valuable discussions. This work was supported by the AFOSR MURI program on bio-inspired flight and NSF grants numbers CCF-0729126, IIS-0808718, and CCF-0915661.

References

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE T Pattern Anal*, 28:44–58, 2005. 2

[2] A. Balan, L. Sigal, M. Black, J. Davis, and H. Haussecker. Detailed human shape and pose from images. In *Computer Vision and Pattern Recognition*, 2007. 2

[3] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *Proceeds of the 9th European Computer Vision*, 3951:404–417, 2006. 4, 5

[4] J.-Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm. Technical report, Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999. 5

[5] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *Int J Comput Vis*, 56:179–194, 2004. 2, 4

[6] S. Y. Cheng and M. M. Trivedi. Articulated human body pose inference from voxel data using a kinematically constrained gaussian mixture model. In *Computer Vision and Pattern Recognition*, 2007. 2

[7] S. Corazza, L. Mündermann, A. Chaudhari, T. Demattio, C. Cobelli, and T. Andriacchi. A markerless motion capture system to study musculoskeletal biomechanics: Visual hull and simulated annealing approach. *Ann Biomed Eng*, 34:1019–1029, 2006. 2, 4

[8] S. A. Dadgar, J.-C. Nebel, and D. Makris. 3D pose estimation from silhouettes in cyclic activities encoded by a dense gaussians mixture model. In *International Conference on Computer Vision Theory and Applications*, 2010. 2

[9] J. M. del Rincon, D. Makris, C. O. Urunuela, and J.-C. Nebel. Tracking human position and lower body parts using Kalman and particle filters constrained by human biomechanics. *IEEE T Syst Man Cy B*, 41:37, 2011. 2, 4

[10] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *International Conference on Computer Vision*, pages 716–721, 1999. 2

[11] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *Int J Comput Vis*, 61:185–205, 2005. 2

[12] A. Elgammal and C. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *Computer Vision and Pattern Recognition*, volume 2, 2004. 2

[13] A. Erol, G. Bebis, M. Nicolescu, R. Boyle, and X. Twombly. A review on vision-based full dof hand motion estimation. In *Computer Vision and Pattern Recognition*, 2005. 2

[14] E. Fontaine, F. Zabala, M. Dickinson, and J. Burdick. Wing and body motion during flight initiation in drosophila revealed by automated visual tracking. *J Exp Biol*, 212:1307–1323, 2009. 2

[15] A. Fossati, M. Dimitrijevic, V. Lepetit, and P. Fua. From canonical poses to 3D motion capture using a single camera. *IEEE T Pattern Anal*, 32:1165–1181, 2010. 2

[16] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *Computer Vision and Pattern Recognition*, 2010. 2

[17] C. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, 1988. 4, 5

[18] G. Hua and Y. Wu. A decentralized probabilistic approach to articulated body tracking. *Comput Vis Image Und*, 108:272–283, 2007. 2

[19] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *Int J Comp Vis*, 29:5–28, 1998. 2, 7

[20] S. Julier and J. LaVoila. On Kalman filtering with nonlinear equality constraints. *IEEE T Signal Proces*, 55:2774–2784, 2007. 5

[21] P. Kohli, J. Rihan, M. Bray, and P. Torr. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *Int J Comput Vis*, 79:285–298, 2008. 2

[22] R. V. D. Merwe and E. Wan. The square-root unscented Kalman filter for state and parameter-estimation. In *IEEE T Acoust Speech*, volume 6, pages 3461–3464, 2001. 2, 3

[23] T. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput Vis Image Und*, 104:90–126, 2006. 2

[24] U. M. Norberg. *Vertebrate Flight - Mechanics, Physiology, Morphology, Ecology, and Evolution*. Springer, 1990. 1

[25] R. Poppe. Vision-based human motion capture analysis: An overview. *Comput Vis Image Und*, 108:4–18, 2007. 2

[26] D. Riskin, J. Bahlman, T. Hubel, J. Ratcliffe, T. Kunz, and S. Swartz. Bats go head-under-heels: the biomechanics of landing on a ceiling. *J Exp Biol*, 212:945, 2009. 2

[27] D. Riskin, J. Iriarte-Diaz, K. Middleton, K. Breuer, and S. Swartz. The effect of body size on the wing movements of pteropodid bats, with insights into thrust and lift production. *J Exp Biol*, 213:4110–4122, 2010. 1, 2, 4

[28] D. Simon. *Optimal State Estimation*. Wiley, 2006. 3, 4, 5

[29] B. Stenger, P. Mendonça, and R. Cipolla. Model-based 3D tracking of an articulated hand. In *Computer Vision and Pattern Recognition*, 2001. 2

[30] S. Swartz, J. Iriarte-Diaz, D. Riskin, A. Song, X. Tian, D. Willis, and K. Breuer. Wing structure and the aerodynamic basis of flight in bats. *AIAA J*, 2007. 1

[31] X. Tian, J. Iriarte-Diaz, K. Middleton, R. Galvao, E. Israeli, A. Roemer, A. Sullivan, A. Song, S. Swartz, and K. Breuer. Direct measurements of the kinematics and dynamics of bat flight. *Bioinspir Biomim*, 1:S10–S18, 2006. 1

[32] J. Wang, D. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE T Pattern Anal*, 30:283–298, 2008. 2