Abstract—Tracking objects that move fast with respect to their size is challenging because it necessitates a large field of views often incompatible with the required spatial and temporal resolutions. Here, we present a novel computer vision system that overcomes this tradeoff by employing a camera with dynamic region-of-interest (ROI) capabilities, combined with an efficient predictive approach. We apply this method to extract the wing kinematics of tethered fruit flies in real time. At each frame, only the pixels immediately surrounding the wing are exposed, and the wing position is extracted. It is then fed to an extended Kalman filter that extracts four key parameters of the measurement time course and, therefore, provides real-time feedback of wing motion. Using this approach, we are able to sample the wing position of both wings at 7 kHz in a 2500 pixel ROI. Our methods promise new applications that can be implemented in general purpose digital hardware for high performance tracking and process control in a broad range of applications in technology and science.

Note to Practitioners—Real-time, high-speed tracking applications are subject to a bandwidth tradeoff between a large field-of-view, a high frame rate and a fine spatial resolution. Because relevant information in tracking problems is often localized in space and time, the bandwidth limitation can be partly overcome from a selective transfer and analysis of only those image data that are of momentary relevance for the tracking task at hand. Based on this approach, we present methods implemented in standard digital hardware that allow a small subset of pixels containing relevant information to be selectively exposed, transferred and analyzed in real time. A prediction model is used to perform this selection and, furthermore, to provide a parametrization of the periodic object motion to control external hardware in real time. Beyond the study of insect flight control, this paper demonstrates a novel approach to track complex and fast moving structures in real-time applications, a challenge often faced in micro and nanotechnologies.

Index Terms—Biomechanics, control systems, Kalman filtering, machine vision, tracking.

I. INTRODUCTION

Many processes in technology and science require systems to be controlled in real time using a contact-free measurement based on visual techniques. Production line processes may require faulty components to be identified visually and subsequently removed. Visual servoing requires manipulator and object positions to be extracted from live images to generate appropriate control commands [1]. External vision systems are also employed to locate micro-robots [2], which are typically not equipped with position sensors. Real-time, contact-free measurement techniques also play an increasingly important role in biology to track moving organisms, while controlling aspects of the experimental process [3], [4].

These processes share the requirement to sample consecutive images, process them on a frame-by-frame basis and use the result of the image analysis to take appropriate action for the process control. Real-time high-speed image analysis is technically demanding and underlies constraints with respect to the amount of information $BW_{\text{max}}$ [bytes/s] that can be transferred and processed between frames. The system bandwidth $BW$ is given by

$$BW = (f \cdot d \cdot \text{FOV})/\sigma^2 < BW_{\text{max}}$$

where $f$ [Hz] is the sampling frequency, $d$ [bytes] the depth of each pixel, FOV [m²] the field-of-view and $\sigma$ [m] the smallest resolvable object size. The amount of data generated at each frame is dictated by the application specific requirements for the spatial resolution $\sigma_{\text{max}}$ and field-of-view FOV_{min}, which are related to the vision system’s pixel array size and optics as follows:

$$\begin{cases} \sigma = (z \cdot p)/F < \sigma_{\text{max}} \\ \text{FOV} = N\sigma^2 > \text{FOV}_{\text{min}} \end{cases}$$

where $F$ [m] is the lens focal length, $z$ [m] is the lens working distance, $p$ [m] is the pixel side length on the camera sensor, and $N$ is the total number of pixels on the camera sensor.

Bandwidth limitations become particularly restrictive when tracking small objects, which tend to move at higher relative velocities and cover larger areas relative to their size than larger objects. Consequently, $f$ and FOV/$\sigma^2$ need to be increased, respectively, and the bandwidth, therefore, quickly becomes a limiting factor in practical applications.

1For these equations, we assume that the smallest detectable change of position is one pixel wide (no subpixel resolution). For simplicity, we also assume a field-of-view (FOV) and camera sensor that have the same width and height.
To avoid bandwidth limitations, a general strategy of transferring only relevant information is clearly beneficial. Generally speaking, relevant information in tracking applications is often localized in space and time, and, therefore, selective image sampling and analysis should provide a powerful strategy to avoid bandwidth limitations in real-time vision tracking applications. In this paper, we describe concepts and techniques applied to implement a high-performance real-time high-speed vision application based on a selective image sampling and analysis strategy. This concept is implemented using a commercially available camera and, thus, provides a flexible and easily duplicable solution to high-speed tracking.

Our solution relies on a process model that provides the spatial and temporal information required to select relevant pixel locations for analysis. We only expose and transfer the relevant portion of the image using a dynamic region-of-interest (ROI) that varies in both position and size, which avoids being limited by image transfer bandwidth. In quantitative terms, we can replace the global FOV in (1) with a smaller local FOV, allowing the BW budget gain $\text{FOV}_{\text{global}}/\text{FOV}_{\text{local}}$ to be used to achieve both a fine spatial resolution and a high frame rate, thus making efficient use of the available system bandwidth. The applied concepts are general and can be implemented in standard high-speed video hardware in applications where bandwidth is a limitation (see comparative diagram of Fig. 1).

II. EXISTING SOLUTIONS

Taking inspiration from early tracking work [5], [6], various approaches have emerged that take advantage of recent advances in imaging technology and computing power. One common approach is to circumvent the bandwidth problem by employing specific knowledge about the task to reduce the data generated at each image. For instance, Nagle Research (Cedar Park, TX, USA) inspects the 3-D surface of 250,000 pharmaceutical tablets per hour using a linear beam of structured laser light to only illuminate a profile of the tablet at each frame. The deviation from the reference line is used as a measure of the tablet’s height. The disadvantage of such highly specialized solutions is the difficulty in adapting them to different applications.

A second strategy is to increase the effective bandwidth from parallelization of image acquisition and processing. For example, multiple temporally and spatially synchronized image acquisition systems have been used to increase the effective frame rate [7], [8]. System bandwidth is increased only in proportion to the number of employed vision systems, however. The advantage of multiple camera systems is particularly problematic for small-scale applications, where space constraints are inherent.

Third, the bandwidth problem can be overcome by integrating image computations at the pixel level [9]–[12]. Massively parallelized pixel level computations, such as linear convolutions [9], have been implemented in very large scale integration (VLSI) vision chips. In this way, relevant information is already computed on chip and only the higher order information needs to be transferred for further processing. An interesting variant of this concept has been implemented in a temporal contrast vision sensor that transmits the addresses (i.e., pixel locations) only of pixels experiencing supra-threshold variation in relative intensity [13], [14]. By filtering relevant information content at the hardware level, only relevant image information is transferred with sub-ms latency to the image processing software, thus reducing the transfer requirements dramatically. VLSI techniques are bound to go beyond proof-of-concept experiments and will be increasingly integrated in complex applications as their pixel array sizes increase and as they become more available as packaged products.

III. TARGET APPLICATION AND IMPLEMENTATION

As part of a research project exploring biomechanical and neural mechanisms of flight control, fruit flies (Drosophila melanogaster) were tethered to a capacitive MEMS micro-force sensor to measure time resolved (10 kHz sampling) flight forces [15]. A high-speed vision system was required to obtain a time resolved and spatially detailed concurrent measurement of both wing positions. We also desired a real-time read-out of the stroke parameters to control external hardware. For example, to implement a visual flight “simulator,” we intended to use the measured kinematic parameters to provide the fly with realistic visual feedback in real time, as normally experienced in free flight. Similarly, we wanted to simulate realistic force feedback stimulation to specialized gyroscopic organs (halteres), which required phase-coupling the mechanical stimulus with the wing stroke [16]. The challenge, therefore, was to perform a real-time measurement of the complex movements of the fruit fly’s two tiny wings (each only about 3 mm long), which beat back and forth more than 200 times each second.

The conceptual approach presented in the introduction is well suited for this tracking problem. First, the relevant image information is concentrated around a small dynamic image region. Second, the wing movement is predictable, allowing a selective sampling of the image. The use of dynamic ROI provides a manifold BW budget gain, which in turn allows the wing kinematics to be reconstructed with a higher spatiotemporal resolution.

To faithfully reconstruct the wing motion and identify relevant kinematic control parameters, the spatial resolution had to be sufficient to distinguish the subtle changes in wing kinematics, but coarse enough to allow the camera to run at sufficiently high frame rates and for both wing paths to stay within the global FOV.
To determine the required spatial resolution, we turned to earlier studies of flight control. The wing kinematics of tethered and free flying fruit flies were captured by Fry et al. at a frame rate of 5 kHz (about 25 samples per wing stroke) [17], [18]. A semi-manual outline fitting procedure was applied to measure the time course of wing motion ([17], their Fig. 1.B). In free flight, the changes of stroke amplitude occurring during fast turning maneuvers (body saccades) were measured in the range of 10°. We therefore opted for an angular resolution of 1°, which is sufficient to distinguish even subtle steering commands. Given a wing span of about 3 mm, the spatial resolution corresponded to about 26 μm in the world coordinate frame.2

The global FOV needed to be large enough to cover the wing stroke path on each side, which required a FOV of at least 7 × 7 mm, or, given the spatial resolution above, 270 × 270 pixels. Within the constraints of the system bandwidth, which in our case were limited by the camera, we were able to obtain sampling rates between 4 and 7 kHz, corresponding to 10–17 samples per stroke for each wing. The latter value represents an eightfold increase to using a static ROI with the same hardware.

### A. Implementation

We captured images of the tethered flying flies using a MVD-1024-Trackcam digital high-speed video camera (Photonfocus AG, Pfäffikon, Switzerland), equipped with a 1024 by 1024 pixels CMOS image sensor (see Fig. 2). At full resolution, the camera achieves only 75 fps, however, the frame rate increases in proportion to a decrease in the chosen image size as predicted by (1) for a constant BW. The ROI and exposure settings could be updated on a frame by frame basis. The camera was connected to a Silicon Software (Mannheim, Germany) Micro Enable III Framegrabber via a Camera Link interface. We used a standard commercial PC with an Intel Pentium IV Dualcore 2.8 GHz processor and 1024 MB of RAM. The operating system was Windows XP. To avoid system interrupts from causing delays (typically on the order of 1 ms), the program’s thread was assigned real-time priority and ran separately from the operating system’s threads on one of the dual processors, thus achieving real-time performance. The software was programmed using Visual C++ 6.0 and made use of OpenCV and Intel Performance Primitives (IPP).

The experiments were performed on an optical table. The camera was equipped with an Edmund Optics (New Jersey, USA) VZM 300i zoom lens with a primary magnification set to 0.75:1. The aperture was fully closed to 1.5 mm to maximize focal depth. The camera’s exposure time was set to 50 μs to minimize wing blurring [17]. Due to the small aperture and short exposure time, extremely bright lighting conditions were required. We achieved this by backlighting with a randomized bundle fiber light guide attached to a 150 W halogen light source (Schott ACE, Mainz, Germany). Diffusive tracing paper was placed 15 mm in front of the light source to provide a homogenous light distribution. A 650 nm high-pass filter (RG 715 Longpass, Edmund Optics) eliminated light in the range visible to the flies [19] to avoid behavioral artifacts.2

Flies were immobilized at 5°C on a custom built Peltier cooled stage and glued to a tungsten probe with UV cured glue (Loctite, Duro Clear Glass Adhesive) using standard techniques [20]. The probe with the attached fly was positioned along six degrees of freedom with a micromanipulator (Sutter MP285), such that the stroke plane roughly coincided with the camera image plane and the wing stroke path was not occluded by the tether.

### IV. Calibration

ROI-based wing tracking and an optimized analysis of the sampled images requires detailed knowledge of the wings’ expected paths at the time of tracking. Because the tethering and lighting conditions vary considerably between consecutive preparations with different flies, detailed image information needs to be acquired between consecutive measurements. For this, we employ an automatic calibration procedure that: a) segments the FOV into functionally relevant areas containing the fly body, the tether and the wing stroke envelope; b) determines the optimal threshold to distinguish the wings from the background; and c) determines the pixel paths along which the wing will be followed during the tracking.

### A. Image Segmentation

The algorithms for image segmentation are based on an analysis of the varying image characteristics between functionally distinct regions (body, tether and stroke envelope). First, a set of 100 full-frame (1024 × 1024 pixels) images are acquired at 75 Hz, whereas the wing phase in each image is arbitrary [see Fig. 3(A)]. Next, a statistical analysis is performed of the pixel values throughout the sequence. This results in two images [see Fig. 3(B)]. The first defines a mask for the stroke envelope based on a binarized image of the variance. The second defines a background image based on the median pixel values.

The dark halo caused by the background lighting is then removed using a vignetting filter and the tether is extracted using a Hough transform. Finally, the body blob is extracted from size and position criteria and its orientation and heading used to separate the left and right sides of the fly [Fig. 3(C) Body].
B. Wing Segmentation Characteristics

The appropriate choice of the threshold used to track the wing edges during tracking is critical. To accommodate for the variation between individual fly preparations, the wing edge detection threshold also needs to be calculated during the calibration phase.

The threshold is determined in two steps. First, the background image is subtracted to increase the signal-to-noise ratio by rendering the system robust to spatial variations in backlighting. Second, the Lloyd-Max quantization algorithm [21] is applied to the pixels belonging to the wing envelope [white pixels in the variance image Fig. 3(B)]. This results in an optimal threshold in the least squares sense [see Fig. 3(C) Wing].

C. Wing Path Extraction

The wing tracking paths are the key information extracted during calibration. For each wing, a circular arc with an optimal radius and centered on the wing hinge is chosen. These arcs correspond to a list of pixel locations used to locate the wing edges and position the ROI in subsequent frames.

To obtain the wing hinge position of each wing, the wing envelope is binarized using the wing edge detection threshold (see above), which provides a representation of the isolated wing. A morphological opening is performed to eliminate possible holes within the wing blob caused by the semitransparent sections of the wing. Next, a Canny edge detector is applied to isolate the wing edges [see Fig. 3(C) Wing edge]. A Hough transform is applied to extract the strongest line. In most of the cases, this line corresponds to the leading edge of the wing. The intersection between each strongest line of the sequence is calculated. The median intersection is taken as an estimate of the wing hinge position [marked with full circles in Fig. 3(D)].

Combining the knowledge of the wing hinge positions and the stroke envelopes, the optimal circular path segments for subsequent wing tracking are chosen by applying an angular range criterium [see the circular arcs in Fig. 3(D)]. The sequence of pixel locations of the path segments are stored together with the corresponding value describing the angle subtended with respect to the body, providing a lookup table for wing position in angular coordinates of the body.

V. Real-Time Data Acquisition

To track the wings robustly, the ROI needs to be placed in appropriate positions along the wing path and the wing position in the ROI determined using efficient image analysis. To initiate and perform wing tracking, as well as to recover from loss of tracking (e.g., because of flight interruption), we implemented the tracking state machine shown in Fig. 4. The process is described in detail below.

A. Wing Edge Detection

The tracking process begins with a detection mode, in which the ROI is placed at one of the two limits of the wing path (maximum or minimum point). This point has two advantages; first, because it is bijective, it corresponds to a unique phase of the wing beat kinematics, creating an unambiguous starting point to the tracking system. Second, the low wing velocity at these points allows the parametric model to adjust itself without losing track. Images are continuously acquired and analyzed using the static ROI until both the leading and trailing edge of the wing are detected. The parametric model is then initialized and the state machine switches to the wing edge tracking mode.

B. Wing Edge Tracking

In the wing edge tracking mode, the ROI is dynamically positioned according to the prediction of a parametric model. The ROI is placed at the position of the expected wing traversal, which can be several wing chords distant from the previous measurement. The current ROI is grabbed and the image processed (see below) to obtain a measure of the wing position, which is then used to update the parametric model of the wing kinematics. The ROI is then placed according to the prediction of the updated model and the process is repeated. If the process fails to detect the wing edges, it continues to estimate the ROI position according to its last measurement. After about 500 μs, the system reverses to the detection mode and sets the ROI back to
its initial position. The grace period gives the system a chance to recover from faulty predictions, which typically happen during the first wing beat after the tracking initialization. The longer interruptions were usually the result of an interruption of flight.

C. Image Analysis: Wing Extraction

The time budget allotted to the processing of the individual images does not allow CPU intensive image analysis, such as blob analysis and pattern matching, to be applied. Using instead only the circular paths extracted during calibration (see Fig. 3), it is possible to reduce the number of pixels to be processed from 2500 to about 60 and perform the necessary image analysis within a time frame of about 50 µs (see Section V-F and Fig. 8 for details).

After transferring the image from the camera to the frame grabber, the following procedure is used to detect the wing edges: The pixel values of the wing path are copied from the frame grabber’s onboard memory to an array in the computer’s RAM (Fig. 5.1). The background image captured during calibration is then subtracted from the sequence (Fig. 5.2) and the resulting sequence binarized using the wing detection threshold (Fig. 5.3). A morphological opening is applied to fill in gaps caused by the semitransparent sections of the wing and remove thin dark stripes that appear if the fly extends its legs in the ROI (Fig. 5.4). Finally, an edge detector extracts the leading and trailing edge of the wing (Fig. 5.5).

D. Process Model: Implementation of an Extended Kalman Filter

The parametric model used to predict wing motion was implemented using a Kalman filter, by virtue of its computational efficiency (due to its recursiveness) and optimality with respect to most statistical criteria [22].

The Kalman filter is initialized prior to entering the wing tracking mode. Its initial state estimate is based on parameters extracted during calibration and on previous knowledge of wing kinematics, e.g., typical wing beat frequency. During tracking, four Kalman filters are used simultaneously to track the positive and negative edges of both left and right wings. The measured wing edge positions are used to update the state estimate and thus increase the accuracy of the prediction for the subsequent measurements. To position the ROI in the subsequent frame, the current state estimates are used to predict the future positions of the positive and negative edges of the wing. The ROI is then centered around the mean value. As a further kinematic parameter, the difference of the wing edge coordinates provide a measure of the wings’ angle of attack.

Because the Kalman filter state estimate represents an online parametrization of the stroke kinematics, it can be used to control external processes in real time. An example for its use as a behavioral metric for real time feedback control is given in Section VI-A.

The periodic motion of each wing edge is modeled using a Fourier series

$$\theta = a_0 + \sum_{i=1}^{N} a_i \cos(\omega t) + b_i \sin(\omega t)$$

(3)

where \(\theta\) represents the 1-D angular position of the edge with respect to the hinge and \(N\) is the number of Fourier terms.

By constraining all parameters except amplitude, offset, frequency and phase, we maintained the original shape of the function, while scaling it in time and space. Equation (3) is rewritten to incorporate these state variables (also see Fig. 6)

$$\begin{cases} \theta = a_0 + \sum_{i=1}^{N} a_i M \sin(\xi + \varphi_i) \\ \xi = \omega t \end{cases}$$

(4)

where \(a_i\) is the relative amplitude of the Fourier coefficient \(i\), \(M\) is the amplitude of the function, \(\xi\) contains the phase information of the function with respect to the frequency \(2\pi \omega\)
TABLE I

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Inter-fly S.D.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td># flies</td>
<td>n = 10</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total # of frames analyzed</td>
<td>28,557,764</td>
<td>2 · 10⁶ frames</td>
<td></td>
</tr>
<tr>
<td>Total tracking time</td>
<td>9142 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Successfully extracted both wing edges</td>
<td>80.0%</td>
<td>9.31%</td>
<td></td>
</tr>
<tr>
<td>Extracted one wing edge</td>
<td>15.9%</td>
<td>6.2%</td>
<td></td>
</tr>
<tr>
<td>No wing edge found</td>
<td>4.1%</td>
<td>4.8%</td>
<td></td>
</tr>
<tr>
<td>Time spent in &quot;Wing edge tracking&quot; mode</td>
<td>&gt;99.9%</td>
<td>&lt;0.1%</td>
<td></td>
</tr>
<tr>
<td>Mean prediction error of Kalman Filter</td>
<td>6.07°</td>
<td>1.77°</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 6. Significance of the state vector: the shape of the model is defined by the relative amplitudes \( a_i \) and relative phases \( \varphi_i \) of each Fourier component \( (N = 3) \). This shape is scaled and offset, both in time \( \omega \) and space \( \xi \), and \( M \) and \( a_0 \) by the EKF to fit the measurements.

(phase is dependent on frequency), and \( \varphi_i \) is the relative phase of the Fourier coefficient \( i \). The \( a_i \) and \( \varphi_i \) coefficients are static values describing the shape of the model. The state vector is \( x_k = [a_0 \ M \ \xi \ \omega ]^T \), and the derivation of the discrete state dynamics from (4) yields

\[
\begin{align*}
    x_k &= \begin{bmatrix}
        1 & 0 & 0 & 0 \\
        0 & 1 & 0 & 0 \\
        0 & 0 & 1 & 0 \\
        a_0 & 0 & 0 & 1
    \end{bmatrix}
    x_{k-1} + W
\end{align*}
\]

where \( T_s = 1/(\text{camera frame rate}) \) is the discrete time step, \( A \) relates the state at the previous time step to the state at the current step, \( W \) is the process noise, assumed to be white, independent and with normal probability distribution \( W \sim N(0, Q) \) where \( Q \) is the process noise covariance. This noise is introduced to take into account the changes in kinematics performed by the fly and the imprecision inherent to any model.

The measurement equation is nonlinear. Therefore, an Extended Kalman Filter (EKF) is used, which linearizes the measurement equation at each time step

\[
\theta = H x_k + v_k
\]

\[
H = \left( \frac{\partial \theta}{\partial x_k}, \frac{\partial \theta}{\partial M}, \frac{\partial \theta}{\partial \xi}, \frac{\partial \theta}{\partial \omega} \right)
\]

where \( v_k \) is the measurement noise, assumed to be white, independent and with normal probability distribution \( v_k \sim N(0, R) \), where \( R \) is the measurement noise covariance.

E. Parameter Selection and Filter Tuning

In the experiments, the static parameters \( \alpha_i \) and \( \varphi_i \) with a total of three Fourier terms \( (N = 3) \) were chosen to fit the kinematic measurements of the wing beat presented in [18]. The two (static) parameters of the Kalman filter, \( R \) and \( Q \) were tuned offline. \( R \) represents the measurement noise and is low since we can extract the position of the wing with high confidence. \( Q \) represents the process noise and is in general more difficult to estimate because the process model is indirectly connected with the measurements. Together, \( R \) and \( Q \) determine how the state estimate is affected by an incoming measurement: a new state vector is estimated at each new measurement by solving the Ricatti equations [22]. We tuned \( R \) and \( Q \) through a nonlinear minimization of the filter’s mean prediction error, applied to a set of measurements from different flies made with a previous estimate of \( R \) and \( Q \). A detailed discussion of tuning techniques for Kalman filters is beyond the scope of this paper, and the interested reader is referred to one of the available books [23].

Inter-fly differences usually did not affect the overall performance of the tracking (see Table I). In rare cases, the Fourier and EKF parameters had to be retuned for a specific fly that showed a particular waveform, such as for example a double maxima at stroke reversal. However, in the majority of cases, the filter did not have to be retuned and the ROI was large enough to avoid tracking losses caused by prediction errors.

F. Process Time Line

The different processes must work together efficiently in a synchronized way to achieve the required frame rates given the hardware constraints. During wing tracking, three pieces of hardware are running in parallel: the camera, the frame-grabber and the computer’s CPU. These three components must be able to determine where the ROI must be placed before the next frame is transferred. The camera (Fig. 8, top row) initially exposes the image during 50 \( \mu \)s. The camera then transfers the image to the frame-grabber (Fig. 8, middle row). The computer receives a signal once the transfer is complete (Fig. 8, bottom row) and starts the image processing. The wing position is extracted as described in Section V-C. The Kalman filter uses the measured position to update its state vector and the following ROI position is calculated (see previous section). This ROI is sent to the frame-grabber which updates the camera once the current frame has been transferred.

Three Fourier terms were sufficient to model the kinematics within their inter-individual variations. A higher number of terms increases the model fit, but also increases the computational cost of the filter.
Fig. 7. Sample from a tracking sequence, the squares and circles represent the extracted positive and negative edges, respectively. The rectangles show the borders of the ROI inside of which pixels were exposed and transferred during a given frame. The median image extracted during calibration is shown in the background for ease of interpretation.

Although the process for a single frame is serial, up to three of these processes are running simultaneously on the different hardware components to maximize the speed. For instance, the next frame is exposed and transferred, while the current frame is being processed. This does not affect the tracking performance because the left and right sides are exposed sequentially, giving an extra frame period to update the ROI on each side.

G. Data Logging and Hardware Control

During tracking, the Kalman filter state vector, the position of the ROI and the position of the leading and trailing edge of the wings are stored in the computer’s heap for each captured frame. The camera data (ROI pixel values) are left on the frame grabber where they are overwritten during the subsequent buffer cycle. While this procedure does not retain the majority of the image data due to bandwidth requirements, the last 1 GB of data remain available on the frame grabber and were retrieved and saved to disk for postprocessing (Fig. 7, see also Discussion).

VI. RESULTS

Various combinations of ROI sizes and frame rates were used to explore optimal parametrization for robust and detailed tracking. We obtained robust tracking performance in a range of FOVs and sampling rates between 4000 and 7000 Hz, which was sufficient for our application. The accuracy of the wing position extraction was under 1° in our experiments.

In our system, the current limitation was the bandwidth of image acquisition and not image processing. At 7000 Hz, the size of the ROI is barely larger than the full width of the wing. Therefore, a further increase in temporal resolution could only be achieved at the cost of spatial resolution by reducing the magnification of the lens.

As shown with a representative example in Fig. 9, we were able to measure the positions of the two wing edges reliably. Our data closely resemble earlier published data on stroke angle ([18], their Fig. 4A).

We reconstructed the sampling procedure of ten consecutive frames based on data from the image buffer saved at the end of a measurement (See Fig. 7). The sequence shows a half a stroke cycle starting with an early down stroke (image 1) to ventral stroke reversal (image 9).

Measurements up to 120 s, or 840,000 samples, were routinely performed. A representative example of such a measurement is shown in Fig. 10. The fly was measured as it was
optically stimulated with a vertically oscillating pattern. The sinusoidal response of the fly is clearly visible.

Table I shows the tracking statistics collected with \( n = 10 \) flies. The robustness of tracking is high with over 80% of frames in which a full wing was extracted and over 99% of time spent in “wing edge tracking” mode.\(^4\) Occasionally, tracking fails during midstroke, where the wings move the fastest, but it typically recovers within the same stroke as the wing slowed down toward stroke reversal. The mistracking is caused by a mismatch between the fly’s kinematics and the EKF’s \( a \text{ priori} \) model. To address this issue, we developed an offline algorithm that can adjust the EKF parameters \( \alpha \) and \( \varphi \) given an initial measurement set. Incomplete wing extraction occurs when both wings temporarily overlap during dorsal flip, and only one of the wing edges can be extracted. Furthermore, flies sometimes extend their legs into the wing path, which led to false wing position readings. These effects are all visible in the visual data we collect at the end of the measurement (see Fig. 7) and can be avoided by a better calibration and an appropriate choice of the opening width in Fig. 5.

A. Proof-of-Concept Experiments

As a proof-of-concept for the general applicability of our process model for real-time hardware control, we used the stroke phase estimate provided by the process model to trigger a stroboscopic flash at defined times during the stroke cycle on a stroke by stroke basis. By linearly varying the phase position between successive strokes, we generated a “slow motion” impression of wing motion that could easily be verified by observation. An example of a stroboscopic video can be found in the supplementary material.

To control the strobe lighting, we estimated the wing phase after each Kalman update and calculated the remaining time to the subsequent strobe flash. If this time was calculated to be less than the camera’s frame time, a digital message was sent out that contained information about the precise timing of the flash.\(^5\) The timing information was decoded on dedicated hardware and used to trigger a flash of 50 \( \mu \)s duration. This application provided a live view of apparent slow motion that revealed induced kinematic changes from stimulation, but also potential problems occurring during tracking.

VII. DISCUSSION AND CONCLUSION

A. Comparison With Existing Technologies

In previous research, several solutions have been proposed to overcome bandwidth limitations to perform real-time high-speed tracking. Analog or hybrid analog/digital vision chips provide particularly interesting solutions as they allow processing power to be integrated at the pixel level. Not only does this allow a massive parallelization of image pre-processing, but it also offers the opportunity of transferring image information selectively to increase the information transfer rate. While these technologies promise highly effective and affordable tracking solutions, implementations using standard digital and software development technology nevertheless offer several advantages.

First, standardized digital solutions have a high degree of flexibility to adapt to specific measurement or control requirements. In our solution, the parameters can readily be tuned to different experimental situations. Such flexibility is difficult to achieve in

\(^4\)The initial tests required to get a proper calibration were removed from the data set. With an improper calibration, the tracking performance is poor.

\(^5\)The number of \( \mu \)s until the actual flash was encoded by seven bits and sent through the computer’s parallel port with a total delay of about 20 \( \mu \)s
vision chips, in which a specific task is typically built into the hardware.

Furthermore, a graphical representation of the tracking process is maintained in digital systems, in our case in the form of the final 1 GB of acquired image data. The availability of raw visual data allows additional off-line analysis to be performed, e.g., as a control for meaningful position extraction. The analysis of the tracking process in analog systems, on the other hand, can be challenging and hard to interpret.

B. Concept Analysis

Our contribution can be summarized as the combination of three key concepts that together form a novel solution to high-speed tracking: 1) Dynamic region-of-interest sampling; 2) Real-time image processing; and 3) Online process model.

1) Dynamic Region-of-Interest: Our implementation in digital hardware makes use of the dynamic ROI capabilities offered by recent CMOS-based digital video cameras. The benefit from transferring only a small part of a FOV is of general relevance to tracking applications in which the bandwidth gain is worth the extra processing time involved with the prediction and update of the ROI. This is the case if the object of interest is small compared to the global FOV.

As sensors increase in pixel array size and frame rates, the bandwidth problem will certainly become more pronounced, making dynamic ROI approaches more appealing. The concept of dynamic ROI need not be limited to standard CMOS cameras. A scanning electron microscope (SEM), for example, can easily be configured to scan a specific area, making it ideal for a high-speed tracking application.

2) Real-Time Image Processing: The real-time image processing is inherent to our solution because the region of interest must be updated as a function of the current wing’s position. A given image had to be analyzed before the following one was completely transferred. By analyzing each image online, only the useful information has to be stored. As a consequence, the system consumes very little memory and can therefore run almost indefinitely. This is useful for applications that must test large parameters spaces, or that are waiting for a single event to occur.

To analyze each image within a few microseconds, we benefited from some simplifying assumptions inherent to our application: the fly was tethered, allowing us to extract the paths of the wings and only analyze relevant pixels and, therefore, drastically reducing the computation costs. Similar simplifications can be found in many tracking applications: higher sampling rates decrease the distance an object has moved between two frames, such that the search of an object can first concentrate on the pixels most likely to contain it, and then extend to pixels with lower likelihoods. Limiting the analysis to a small amount of pixels allows even complex signal processing computations to be performed fast.

For future applications that employ higher computational power or that have less stringent spatiotemporal requirements, online pattern recognition could be used to extract more information. The large body of work that has been done in identifying the pose and kinematics of periodic motions in humans could be suitable [24].

3) Online Process Model: To be able to position the ROI, the position of the object must be predictable from past measurements. The prediction model is therefore a key element of our tracking approach. Here, a tradeoff must be found on the level of complexity of the model. A more complex model will predict the position of the object more accurately, therefore allowing a smaller region of interest to be chosen and increasing the frame rate of the camera. A more complex model also involves more computations, however, and will add time to the processing. The level of complexity must therefore be chosen in an iterative way to find an optimal tradeoff between transfer and processing constraints.

In our application, we opted for an EKF mainly due to its recursive implementation and its ability to precisely extract a state vector at each measurement instance, therefore providing real-time analysis of the measurement time-course. Simpler techniques, such as alpha-beta-gamma filters, do not provide a real-time analysis of the data. On the other hand, more complex techniques, such as unscented Kalman filters or particle filters, have the potential to be more precise but also more computationally expensive. For our application, the EKF provided the best tradeoff.

C. Concept Synthesis and Generalization

We demonstrated the feasibility and robustness of our techniques by tracking the wing motions of a tethered insect. Similar applications in biology may involve the requirement to track exceedingly fast movements of appendages or the body (walking/flying). Furthermore, the ca. 1 GB of image data left in the ring buffer at the end of a measurement allows more complex image processing algorithms to be applied offline.

The real-time functionality is nonetheless crucial to applications that require the control of an external piece of hardware based on its current process state. This is of general relevance to behavioral research paradigms in biology, because it provides controllable sensory conditions that nevertheless depend on the animal’s (intended) behavior. As robotics become more autonomous and integrated into processes, the need for rapid and robust process estimations based on non-contact sensors will also increase. For instance, production lines are likely to increase in speed and complexity. Likewise, visual servoing will be applied to faster manipulators. In these cases, the complexity and structure of the process model is completely dictated by the required feedback.

In our application, two of the Kalman filter’s state variables were directly employed for the control of external hardware. Because the difference in stroke amplitude between the right and the left wings are associated with intended turning maneuvers of the tethered fly [25], we could use them to control the fly’s visual panorama in a “flight simulator.” We also used the Kalman filter’s phase variable to stroboscopically illuminate the wings or excite the mechanosensors at precisely defined wing phases.

The concept presented in this paper can be applied to other periodic motion by adjusting the shape of the EKF’s fitted curve.

6The two videos in the supplementary material demonstrate the strobing application.
For nonperiodic motions, the EKF can easily be adapted to, e.g., a constant acceleration model that predicts the future location of the object based on its present position, velocity and acceleration. In general, the increase in sampling rate made possible using the ROI approach allows the complexity of the prediction model to be reduced. In the extreme case, sufficient tracking speed would allow even arbitrary trajectories to be tracked successfully with a minimal model.

As autonomous robots get smaller, their size and speed approach that of the biological counterparts from which they are often inspired. The technique developed in this paper could therefore be relevant to the tracking of micro and nano robots, where high relative velocities make them hard to follow and where robust visual position feedback is crucial for sensing and control.

In general, the combination of dynamic ROI, real-time image processing and online process model promises broad applications in experimental research and process control, whenever the status of a system must be rapidly evaluated as part of a control loop.

REFERENCES
in 2003 and 2008 and won Best Paper Awards at major robotics conferences and journals in 2004, 2005, 2006, 2007, 2008, and 2009. He was named to the 2005 “Scientific American 50,” Scientific American magazine’s annual list recognizing 50 outstanding acts of leadership in science and technology from the past year for his efforts in nanotube manufacturing. His laboratory won the 2007 and 2009 RoboCup Nanogram Competition, both times the event has been held. He serves on or has been a member of the editorial boards of the IEEE TRANSACTIONS ON ROBOTICS, the IEEE TRANSACTIONS ON NANOTECHNOLOGY, the Journal of Micromechatronics, the Journal of Optomechatronics, the International Journal of Biomechatronics and Biomedical Robotics, and the IEEE Robotics and Automation Magazine. He has chaired several international workshops and conferences, has served as the head of the Department of Mechanical and Process Engineering from 2005 to 2007, and is currently the Chairman of the Board of Directors of the ETH Electron Microscopy Center (EMEZ).

Steven N. Fry received the Diploma degree and the Ph.D. degree in natural sciences from the University of Zürich, Zürich, Switzerland, in 1994 and 1999, respectively.

He continued as a Postdoctoral Associate with D. Robert at the University of Zürich, Zürich, Switzerland (1999-2000) and M. H. Dickinson at the University of California, Berkeley (2000-2002). He is currently working as a Senior Researcher at the Institute of Neuroinformatics (INI) at the Swiss Federal Institute of Technology (ETH) and the University of Zürich, where he leads a work group researching the biomechanics and sensory processing in fruit flies.