Mining automatically estimated poses from video recordings of top athletes

Rainer Lienhart, Moritz Einfalt, Dan Zecha

Angaben zur Veröffentlichung / Publication details:

Lienhart, Rainer, Moritz Einfalt, and Dan Zecha. 2018. "Mining automatically estimated poses from video recordings of top athletes." *International Journal of Computer Science in Sport* 17 (2): 94–112. https://doi.org/10.2478/ijcss-2018-0005.

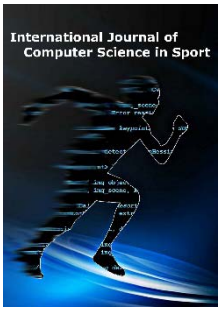
International Journal of Computer Science in Sport

Volume 17, Issue 2, 2018

Journal homepage: http://iacss.org/index.php?id=30

sciendo

DOI: 10.2478/ijcss-2018-0005



Mining Automatically Estimated Poses from Video Recordings of Top Athletes

Lienhart, R., Einfalt, M., Zecha, D.

Multimedia Computing and Computer Vision Lab, Computer Science Department, University of Augsburg, Germany

Abstract

Human pose detection systems based on state-of-the-art DNNs are about to be extended, adapted and re-trained to fit the application domain of specific sports. Therefore, plenty of noisy pose data will soon be available from videos recorded at a regular and frequent basis. This work is among the first to develop mining algorithms that can mine the expected abundance of noisy and annotation-free pose data from video recordings in individual sports. Using swimming as an example of a sport with dominant cyclic motion, we show how to determine unsupervised time-continuous cycle speeds and temporally striking poses as well as measure unsupervised cycle stability over time. The average error in cycle length estimation across all strokes is 0.43 frames at 50 fps compared to manual annotations. Additionally, we use long jump as an example of a sport with a rigid phase-based motion to present a technique to automatically partition the temporally estimated pose sequences into their respective phases with a mAP of 0.89. This enables the extraction of performance relevant, pose-based metrics currently used by national professional sports associations. Experimental results prove the effectiveness of our mining algorithms, which can also be applied to other cycle-based or phase-based types of sport.

KEYWORDS: HUMAN POSE ANALYSIS, HUMAN POSE MINING, POSE MINING IN SPORTS

Introduction

Since the arrival of deep neural networks (DNNs), state-of-the-art DNN-based human pose estimation systems have made huge progress in detection performance and precision on benchmark datasets (Wei, Ramakrishna, Kanade, & Sheikh, 2016; Andriluka, Pishchulin, Gehler, & Schiele, 2014; Chu et al., 2017; Yang, Li, Ouyang, Li, & Wang, 2017; Newell, Yang, & Deng, 2016). Recently, these research systems have been extended, adapted and re-trained to fit the application domain of specific sports (Zecha, Eggert, & Lienhart, 2017; Einfalt, Zecha, & Lienhart, 2018). Soon they will disrupt current performance analyses in all kinds of sport as the amount of available pose data will explode due to automation. Until today, pose determination and analysis of top-class athletes is very time-consuming manual work. Hence, it is scarcely performed by the national professional sports associations even for top-class athletes and almost never for athletes below that level. The forthcoming availability of automatic pose detection systems will make plenty of noisy[1] pose data available from videos recorded at a much more regular and frequent basis. Despite this imminent change in data quantity of noisy pose data in several orders of magnitude, very little research has been devoted to explore the opportunities of extracting informative and performance relevant information from these pose detection results through data mining.

This work is focusing on this task and presents a set of unsupervised pose mining algorithms that extract or enable extraction of important information about athletes and how they compare to their peers. We will use world-class swimmers in swimming channels as an example of a sport with dominant cyclical motion and long jumping as an example of a sport with clear chronologically sequential phases. Our pose data is created by the image-based pose detection systems presented in Einfalt et al. (2018) and Wei et al. (2016). Detected sample poses are depicted in Figure 1. Note, however, that our algorithms are supposed to work with the output of any state-of-the-art image or video-based pose detection system.

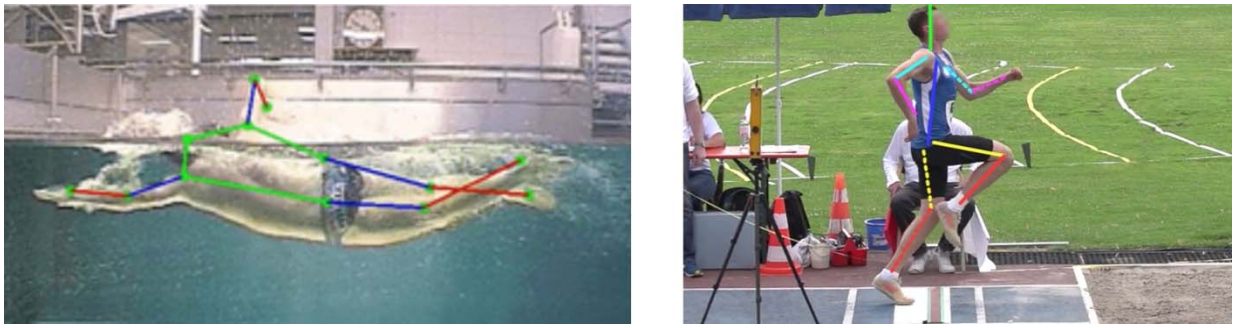


Figure 1: Detected poses of a swimmer and a long jumper.

Related Work

Both works in Ren, Lei, and Zhang (2011) and Vögele, Krüger, and Klein (2014) cluster 3D motion capture data and determine algorithmically similar motion sequences for database retrieval, while Sedmidubsky, Valcik, and Zezula (2013) develop a similarity algorithm for comparing key-poses, which is used to index motion features in human motion databases. For the task of action recognition, Lv and Nevatia (2007) and Baysal, Kurt, and Duygulu (2010)

[1] The term *noise* refers here to the residual error between the automatically determined poses and the correct but unobservable poses. As we have no means to determine the correct poses, we use temporally sparse manual annotations as our ground truth (GT) poses.

perform clustering on shape-based representations of 2d human poses and learn weights to favor distinctive key-poses. Both show that temporal context is superfluous if human poses with high discriminative power are used for action recognition. Data mining for action recognition based solely on joint location estimates is still scarce. Wang, Wang, and Yuille (2013) propose spatial-part-sets obtained from clustering parts of the human pose to obtain distinctive, co-occurring spatial configurations of body parts. They show that these sets improve the task of action recognition and additionally the initial pose estimates.

In the field of sport footage analysis, the task of action recognition often translates to the identification of specific motion sequences within a sport activity. De Souza Vicente et al. (2016) use latent-dynamic conditional random fields on RGB-d skeleton estimates of Taekwondo fighters to identify specific kicks and punches in a fight sequence. Long jump video indexing has been researched by Wu, Ma, Zhan, and Zhong (2002), who perform motion estimation and segmentation of camera and athlete motion velocity to extract and classify semantic sequences of long jump athletes. Li, Tang, Wu, Zhang, and Lin (2010) build a similar system for high diving athletes. They also derive human pose from shape and train a Hidden Markov Model to classify a partial motion of jumps.

The extraction of kinematic parameters of athletes from video footage, specifically stroke rates of swimmers, was recently researched by Victor, He, Morgan, and Miniutti (2017), who perform stroke frequency detection on athletes in a generic swimming pool. Zecha et al. (2017) derive additional kinematic parameters from swimmers in a swimming channel by determining inner-cyclic interval lengths and frequencies through key-pose retrieval. Compared to other approaches that rely on the concept of identifying key-poses, their approach lets a human expert define what a discriminative key-pose should be.

Contributions

While our work is influenced by the related work above, it is new and unparalleled to existing works due to the (a) large-scale, (b) data-mining as well as (c) time-continuous aspect of the proposed mining algorithms. Previous work heavily relies either on very few, correctly annotated ground truth data to train models or recordings from motion capture systems. In detail, our contibutions are

1. Some sports are dominated by cyclical motion, some by clear chronologically sequential phases. For cyclical sports, we present novel mining algorithms to determine unsupervised performance parameters such as time-continuous cycle speeds, temporally striking poses and cycle stability in swimming as a representative sports. Additionally, we use long jump as an example of a sport with a rigid phase-based motion to present a technique to automatically partition the temporally estimated pose sequences into their respective phases. This enables the extraction of performance relevant, pose-based metrics currently used by national professional sports associations.
2. Manual pose annotations are typically confined to a few key poses during the relevant actions (i.e., annotations are temporally sparse), and so are the derived performance parameters. We, however, exploit that pose detection systems can process every frame. Our mining algorithm for extracting performance parameters produce a temporally dense output. They robustly estimate the performance parameters time-continuously for $t \in \mathbb{R}^+$ which in turn can be sampled, e. g., at frame-rate. This has not been done before.

3. Our mining algorithms are the first to focus on the massive processing of noisy outputs of DNN-based pose detection systems for large-scale analysis. Robustness to errors in pose estimates is key and implicitly handled.

Methods

Measuring Pose Similarity

In computer vision, the human pose at a given time is defined by a set of locations of important key points on a human, such as joint locations. The number of key points varies based on the application domain. In the analysis of top-level athletes, the pose is the basis of many performance indicators and may include points on the device(s) the athlete is using. Since the pose is so central to most sports-related performance indicators, we need to be able to reliably evaluate the similarity or distance between poses. This section develops our pose distance measure that is invariant to translation, scale and rotation in the image plane. It will be used by all our algorithms.

Throughout the paper, we assume that all video recordings have been processed by some pose detection system. In our case, we use the system from Einfalt et al. (2018) for swimming and Wei et al. (2016) for long jump. We do not expect to have a pose for all frames. Through some parts of a video, the athlete might not be completely in the picture, if present at all. Or the detection conditions are so difficult that the detection system does not detect any pose. Our mining algorithms have to deal with that. However, we assume that the athletes perform the desired action for more than half of the duration of each video clip. Also, we discard all poses that are only partially detected to make mining simpler.

Pose Definition

Mathematically, a 2D pose p is a sequence of N two-dimensional points, where each 2D point by convention specifies the coordinates of the center of a joint location or of some other reference location on the human or object(s) under investigation:

$$p = \{(x_k, y_k)\}_{k=1}^{N} \equiv \begin{pmatrix} x_1 & \cdots & x_N \\ y_1 & \cdots & y_N \end{pmatrix}^T \quad (1)$$

Our human pose model consists of $N = 14$ joints. Throughout the paper, a pose clip and pose sequence denote a temporal sequence $\boldsymbol{p}_{t1:t2}$ of poses $[p_{t1}, p_{t1+1}, \ldots, p_{t2-1}, p_{t2}]$. The term pose clip hints at a short temporal pose sequences (e.g. 0.5 to 2 seconds), while pose sequence often refers to much longer durations – up to the complete video duration (e.g., 30 seconds and longer). Video time and time intervals are usually expressed using sequential frame numbers as we assume recordings at a constant frame rate.

Aligning Two Poses

Before we can define our pose distance measure, we need to specify how we align a pose p to a given reference pose p_r by finding the scaling factor s, rotation angle θ and translation $t = (t_x, t_y)$, which applied to each joint of p results in p' and minimizes the mean square error (MSE) between the transformed pose p' and the reference pose p_r (Rowley, Baluja, & Kanade, 1998):

$$\text{MSE}\,(p_r, p) := \text{MSE}\,(p_r, p') = \frac{1}{2N} \|p_{r,reshaped} - p'_{reshaped}\|_2^2 \quad (2)$$

with

$$t_{trans} = (a, b, t_x, t_y)^T \tag{3}$$

and

$$p'_{reshaped} := \begin{pmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \end{pmatrix} = \begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} =: A \cdot t_{trans} \tag{4}$$

Note that the $N \times 2$ matrix p' is reshaped to a $2N \times 1$ vector $p'_{reshaped}$. The pseudo-inverse $t^{opt}_{trans} = (A^T A)^{-1} A^T p_{r,reshaped}$ gives us in closed form the transformation of pose p that minimizes the mean squared error between the joints of reference pose p_r and transformed pose p'. Each joint (x, y) of p is mapped to

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s\cos\theta & -s\sin\theta \\ s\sin\theta & s\cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} a & -b & t_x \\ b & a & t_y \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{5}$$

using the optimal transformation t^{opt}_{trans}. The associated MSE value indicates how well a pose fits a reference pose. Thus, given a set of poses, their associated MSE values can be used to rank these poses according to their fitness to the reference pose. However, two peculiarities about MSE (p_r, p) need to be emphasized:

1. It is not symmetric, i.e., generally MSE $(p_r, p) \neq$ MSE (p, p_r). The reason for this is that the pose is always scaled to the size of the reference pose. Thus, if their two scales are very different, so will be MSE (p_r, p) and MSE (p, p_r).
2. Its magnitude depends on the scale of the reference pose. Doubling the reference pose's scale will quadruple the MSE value. Thus, if a pose is compared against various reference poses, the scale of the references poses matters.

Both peculiarities of the MSE (p_r, p) value suggest that we need to normalize the poses we are comparing to get universally comparable MSE values and thus a universally applicable distance measure between two poses.

Pose Distance Measure

In pose detection evaluation it is common to scale a reference pose by assigning a fixed size either to the length of the distance between two characteristic points of the pose or to the head. While using two reference points or a single rectangle may be fine in case of ground truth annotations, it is statistically not advisable for noisy detection results. We need a normalization that is based on more joints to reduce noise. Hence the scale s_p of pose p is defined as the average distance of all joints of a pose to its center of mass $c_p = (c_{p,x}, c_{p,y})^T$:

$$s_p = \frac{1}{N} \sum_{k=1}^{N} \left\| \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \begin{pmatrix} c_{p,x} \\ c_{p,y} \end{pmatrix} \right\|_2 \tag{6}$$

with

$$\begin{pmatrix} c_{p,x} \\ c_{p,y} \end{pmatrix} = \frac{1}{N} \sum_{k=1}^{N} \begin{pmatrix} x_k \\ y_k \end{pmatrix} \tag{7}$$

Given an arbitrary reference scale s_{ref}, we define our symmetric translation, rotation and scale invariant distance measure between two poses as

$$MSE_{norm}(p_1, p_2) = \frac{s_{ref}^2}{2s_{p_1}^2}\text{MSE}\,(p_1, p_2) + \frac{s_{ref}^2}{2s_{p_2}^2}\text{MSE}\,(p_2, p_1) \quad (8)$$

It enables us to judge pose similarity between poses derived from videos recorded by different cameras, at different locations and distances to the athletes.

Mining Pose Data of Swimmers

Cyclical motions play a decisive and dominant role in numerous sports disciplines, e.g., in cycling, rowing, running, and swimming. In this section, we use swimming as an example to explore what kind of automated mining we can perform on the detected noisy poses. We use the pose data derived from world class swimmers recorded in a swimming channel. A single athlete jumps into the flowing water against the flow (from the right in Figure 1 left), swims to the middle in any manner (e.g., by an extended set of underwater kicks or by freestyle on the water surface) and then starts the cyclic stroke under test. The video recording can start any time between the dive and the action of interest (.i.e, swimming a specific stroke) and stops shortly after it has ended. During most of the recording time the athlete executes the cyclic motion under test.

Time-Continuous Cycle Speeds

For all types of sports with dominant cyclical motions, the change in cycle speed over time is a very indicative performance parameter. It can be derived through data mining without providing any knowledge to the system, but the automatically detected joint locations for each pose throughout a video sequence. Given a pose at time t, the *cycle speed* at time t is defined as 1 over the time needed to arrive at this pose from the same pose one cycle before. In the case of a swimmer, the desired cycle speed information is strokes per minutes, which can be derived from the stroke length in frames given the video sampling rate in frames per seconds by

$$\frac{\#\ strokes}{minute} = \left(\frac{\#\ frames}{stroke}\right)^{-1} \cdot \frac{\#\ frames}{seconds} \cdot \frac{60\ seconds}{minute} \quad (9)$$

The *stroke length* is measured by the number of frames passed from the same pose one cycle before to the current pose.

In the following, we describe the individual steps of our statistically robust algorithm to extract time-continuous cycle speeds by first stating the characteristic property of cyclic motion we exploit, followed by an explanation how we exploit it. The adjective *time-continuous* denotes that we will estimate the **cycle speed** for $t \in R^+$ which in turn can be sampled at every frame of a video in which the cyclic motion is performed:

1. **Input:** A sequence P of poses p for a video: $P = \{(f_p, p)\}_{f_p}$.

   The ordered set consists of pairs describing a detected pose p and a frame number f_p in which it was detected. The subscript f of set $\{(f, \ldots)\}$ indicates that the elements in the set are ordered and indexed by frame number f. Note that we might not have a pose for every video frame.

2. **Property:** Different phases of a cycle and their associated poses are run through regularly. As a consequence a pose p from a cycle should match periodically at cycle speed with poses in P. These matching poses p' to a given pose p identify themselves visually as minima in the graph plotting the frame number of poses p' against its

normalized distance to given pose p. Therefore, we compare every pose p in a video against every other pose p' and keep for each pose p a list L_p of matches:

$$L_p = \left\{\left(f_{p'}, p', \text{MSE}_{norm}(p, p')\right)\right\}_{f_{p'}} \quad \forall p \in P \tag{10}$$

Poses match if their normalized MSE value is below a given threshold. For a target scale of $s_{ref} = 100$ we use a threshold of 49 (on avg. 7 pixels in each direction for each joint).

3. **Property:** Not every pose is temporally striking.

   An athlete might stay for some time – even during a cycle – in a very similar pose, for instance, in streamline position after bringing the arms forward in breaststroke. However, at one point this specific pose will end in order to enter the next phase of the cycle. Thus, from step 2, we sometimes not only get correct matches, but also nearby close matches. We consolidate our raw matches in L_p by first temporally clustering poses p'. A new cluster is started if a gap of more than a few frames lies between two chronologically consecutive poses in L_p. Each temporal cluster is then consolidated to the pose p_c with minimal normalized MSE to the pose p. The cluster is also attributed with its *temporal spread*, i.e., the maximal temporal distance of a pose in the cluster from the frame with the consolidated pose p_c, leading us to the *reoccurrence sequences* L'_p with

$$L'_p = \left\{\left(f_{p_c}, p_c, spread\right)\right\}_{f_{p_c}} \quad \forall p \in P \tag{11}$$

   and for the complete video to $L_{video} = \left\{\left(f_p, p, L'_p\right)\right\}_{f_p}$.

4. **Property:** Temporally non-striking poses are unsuitable to identify cyclic motion. Therefore, all clusters with a temporal spread larger than a given threshold are deleted.

   In our experiments we set this value to 10 frames at 50 frames per seconds, resulting in

$$L''_p = \left\{\left(f_{p_c}, p_c, spread\right) | spread < 10\right\}_{f_{p_c}} \quad \forall p \in P. \tag{12}$$

5. **Property:** Most of the time the video shows the athlete executing the cyclical motion under test. Consequently, poses from the cyclic motion should most often be found.

   Hence, we create a histogram over the lengths of the reoccurrences sequences ($|L''_p|$) for the various poses p. We decided to keep only those reoccurrence sequences L''_p which belong to the 50% longest ones:

$$L'_{video} = \left\{\left(f_p, p, L''_p\right) | \left|L''_p\right| \geq \underset{p' \in P}{\text{median}}\left(\left|L'''_p\right|\right)\right\}_{f_p} \tag{13}$$

6. **Property:** The observed difference of the frame numbers in each reoccurrence sequence in L'_{video} between two chronologically consecutive matches should most frequently reflect the actual stroke length.

   Figure 2 shows two sample plots. On the x-axis, we have the minuend of the difference and the difference value on the y-axis. The blue and yellow dots display all observed difference values from L'_{video}. From them we derive our final robust estimate by local median filtering in two steps: (1) We take each frame number f with at least one difference value and determine the median of the observed stroke lengths (= difference values) in a window of ± 2 seconds (approx. 2 to 4 stroke cycles). We remove all difference values at frame number f, which deviate more than 10% from the median. E.g., @50 fps a median stroke length of 60 frames results in keeping only difference

values in [54,66]. The deleted difference values are shown in yellow in Figure 2, while the remaining ones are shown in blue. (2) We piecewise approximate the remaining data points with a polynomial of degree 5 over roughly 3 cycles while simultaneously enforcing a smoothness condition at the piecewise boundaries.

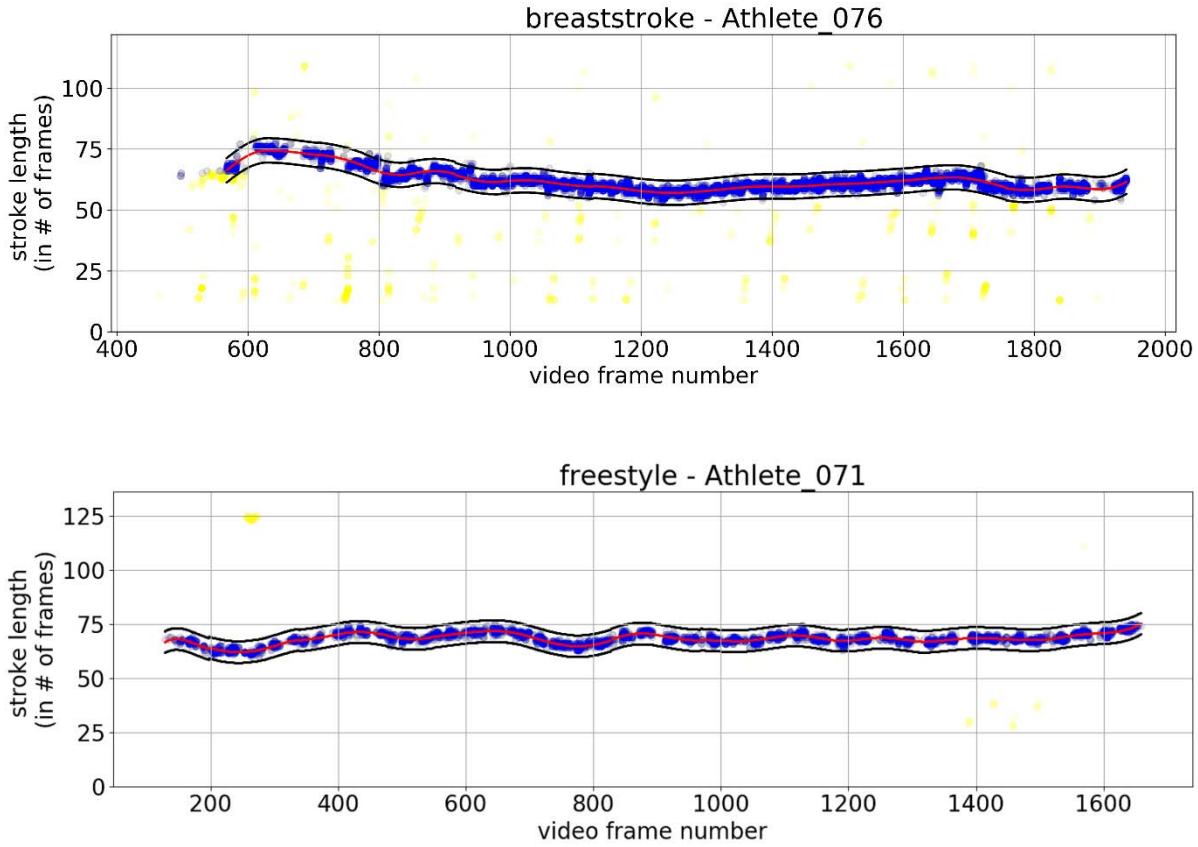


Figure 2: Two examples showing the frame differences between chronologically consecutive matches of all reoccurrence sequences in L'_{video} on the y-axis against the frame numbers of the minuends of the differences on the x-axis. The redline visualizes the time-continuous estimate of stroke cycle lengths, while black lines indicate the associate $\pm 10\%$ corridor.

This approximation gives us our time-continuous estimates of the stroke cycle length over the interval in the video throughout which the stroke was performed. As a side effect it also automatically identifies the temporal range in the video during which the stroke was performed by the frame number ranges for which we have cycle speeds. The same technique is applicable to determine the kicks per minutes for freestyle and backstroke by restricting the pose to joints from the hip downwards.

Temporally Striking Poses

During a cyclical motion some poses are more striking than others with respect to a given criterion. One such highly relevant criterion is how well a repeating pose can be localized temporally, i.e., how unique and salient it is with respect to its temporally nearby poses. The temporally most striking poses can be used, e.g., to align multiple cycles of the same swimmer for visual comparison.

Commonly, local salience is measured by comparing the local reference to its surrounding. In our case the local reference is a pose p_r at frame r or a short sequence of poses $p_{r-\Delta w_l}, \dots, p_r, \dots, p_{r+\Delta w_l}$ centered around that pose, and we compare the sequence to the

temporally nearby poses. Thus, we can compute saliency by:

$$saliency(p_r) = \sum_{\Delta w_s=-w_s}^{w_s} \sum_{\Delta w_l=-w_l}^{w_l} \frac{MSE\left(p_{r+\Delta w_l}, p_{r+\Delta w_l+\Delta w_s}\right)}{(2w_s+1)(2w_l+1)} \quad (14)$$

Experimentally, the saliency measure was insensitive with respect to the choices of w_l and w_s. Both were arbitrarily set to 4.

The salience values for each pose during the cyclic motion of a video can be exploited to extract the K most salient poses of a cycle. Hereto, we take the top N most salient poses ($N \gg K$) and cluster them with affinity propagation (AP) (Frey & Dueck, 2007). Salient poses due to pose errors will be in small clusters, while our most representative poses are the representative poses of the K largest clusters.

For determining the most salient pose of an athlete's stroke, it is sufficient to pick the top 20 most salient poses, cluster them with AP and retrieve the cluster representative with the most poses assigned. Figure 3 shows one example for each stroke. Note that the most salient pose is another mean to determine the cycle speed reliably cycle by cycle as this pose is most reliably localized in time. However, we only get one cycle speed value per cycle.

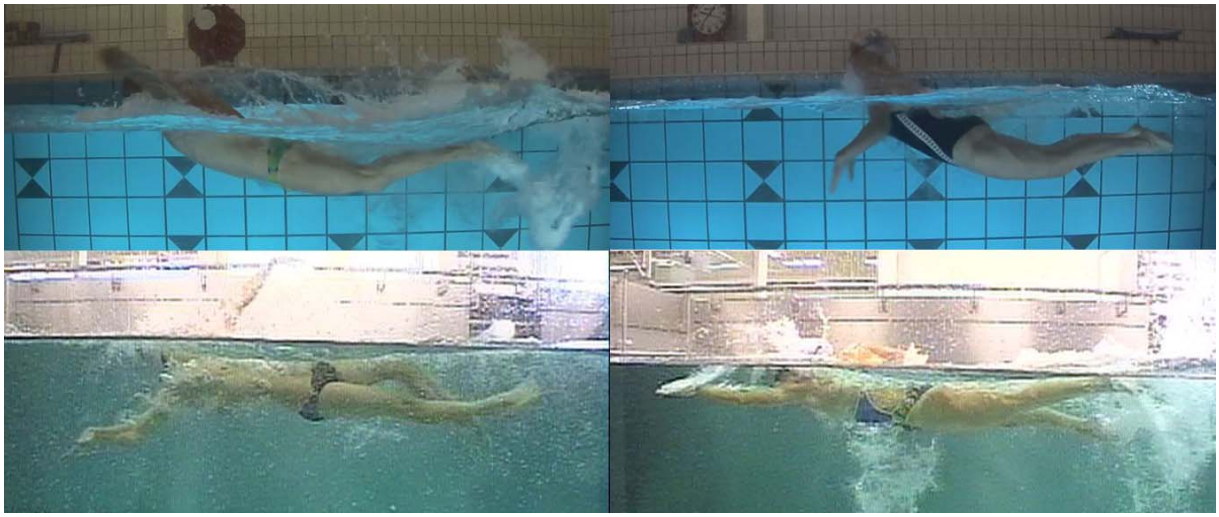


Figure 3: Examples of temporally striking poses; top left to bottom right: fly, breast, back and free.

Cycle Stability

A common and decisive feature among winning top athletes is their trait to show off a very stable stroke pattern over time, under increasing fatigue and at different pace. One way to measure stroke cycle stability is to select a reference pose clip of one complete cycle and match this reference pose clip repeatedly over the complete pose sequence of the same video. Alternatively, the reference pose clip is matched repeatedly over a set of pose sequences derived from a set of videos recordings of some performance test such as the 5×200m step test after Pansold (Pyne, Lee, & Swanwick, 2001; Pansold, Zinner, & Gabriel, 1985). Given all these clip matches and their associated matching scores, an average score of matching can be computed and taken as an indicator of stroke cycle stability: The better the average matching score, the more stable the stroke of the athlete. Alternatively, the matching score may be plotted versus time in order to analyze, how much the stroke changes from the desired one over (race) time. A reference pose cycle may automatically be chosen by selecting a clip between

two contiguous occurrences of a temporally striking pose or by specifying a desired/ideal stroke cycle.

Levenshtein distance: With regards to that goal, we first turn our attention to the task of how to match a pose clip to a longer pose sequence and compute matching scores. We phrase the task to solve in terms of the well-studied problem of approximate substring matching: The task of finding all matches of a substring pat in a longer document $text$, while allowing up to some specified level of discrepancies. In our application, a pose represents a character and a clip/sequence of poses our substring/document. The difference between 'characters' is measured by a [0,1]-bounded distance function derived from the normalized MSE between two poses:

$$dist\_fct(p_1, p_2) = \begin{cases} 0 & \text{if } MSE_n(p_1, p_2) \leq th_{same} \\ \dfrac{MSE_n(p_1, p_2) - th_{same}}{th_{diff} - th_{same}} & \text{if } MSE_n(p_1, p_2) \geq th_{diff} \\ 1 & \text{else} \end{cases} \quad (15)$$

The cost of transforming one pose into another is 0 for poses which are considered the same ($MSE_n(p_1, p_2) \leq th_{same}$) and 1 for poses which are considered different ($MSE_n(p_1, p_2) \geq th_{diff}$). Between these two extremes, the transformation cost is linearly scaled based on the MSE_n value.

Any algorithm to compute the Levenshtein distance (Levenshtein, 1966; Meyers, 1994) and its generalization called edit distance is suitable to perform matching and compute a matching score between a search pattern pat and a longer document $text$ at every possible end point location of a match within $text$. It results in a matrix d of matching costs of size $len(pat) \times len(text)$, where $d[i,j]$ is the cost of matching the first i characters of pat up to end point j in $text$.

We use our custom distance function not only for transformations, but also for insertions and deletions. We deliberately made this chose as it better fits the characteristic of swimming: The absolute duration of a stroke cycle, i.e. the number of poses in a sequence, depends on the pace of the swimmer. However, the better the athlete, the more consistent he/she executes the succession of poses across different paces. We therefore do not want to see an additional cost if, e.g., a swimmer stays longer/shorter in a perfect streamline position or if he/she goes slower/ faster through the recovery phase of a stroke cycle than the reference clip. Pace is already captured by the cycle speed. Here we only want to focus on the stability of the stroke pattern, no matter how fast the stroke is executed. Note that swimmers with less than perfect swimming technique typically modify their poses when changing pace.

Match extraction: The matching distances $d[len(pat), j]$ of the complete search pattern pat computed by the edit distance at end point j in $text$ are normalized by the virtual matching length, i.e., by the number of transformations, deletions and insertions needed for that match. We call this $len(text)$-dimensional vector of normalized matching scores over all possible end points in $xt\ score_{match}(pat, text)$. All clear minima in it identify the end points of all matches of the pose clip to the sequence together with the associated matching distances. Since our pose clips are highly specific in matching, our minima search does not require any non-maximum suppression. The matching sequence is derived by backtracking from this end point to the beginning of the match by using $d[i,j]$. Figure 4 shows one example of matched poses of two different stroke cycles.

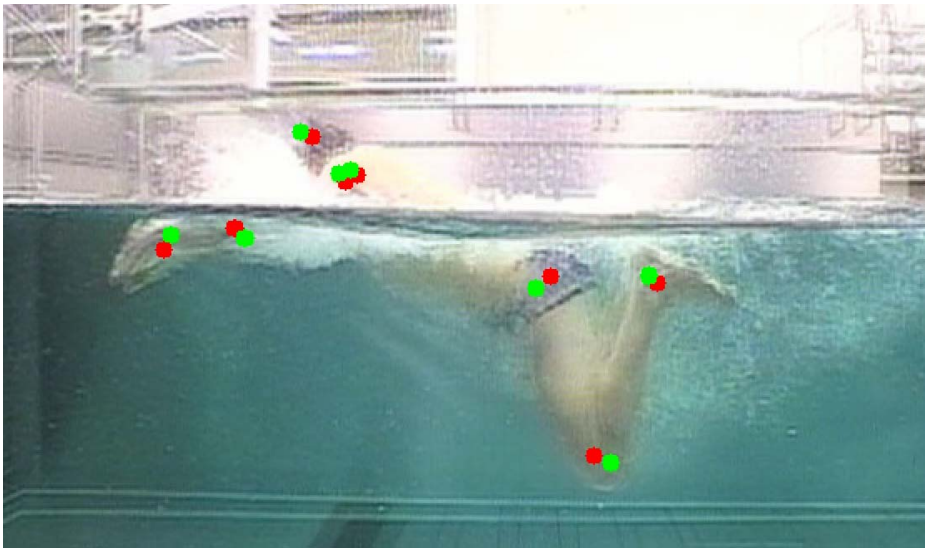


Figure 4: Alignment example of the same swimmer at different stroke cycles. Joints of the reference/matching pose are in shown in red/ green.

Athlete Recognition: While we were matching a given pose clip to all videos in our video database, we accidentally discovered that $score_{match}$ is also a perfect tool to automatically recognize a specific athlete. Usually, when matching a pose clip to the pose sequence of a different male or female swimmer, $score_{match}$ is 4 to 8 times higher in comparison to the score computed against the video the pose clip was taken from. However, in this case the matching score was as low as a match against the same video despite being recorded at a different time in a different swimming channel. Thus, $score_{match}$ can be used to identify a swimmer.

Figure 5 summarizes the overall processing chain of our mining algorithms for cyclic motions.

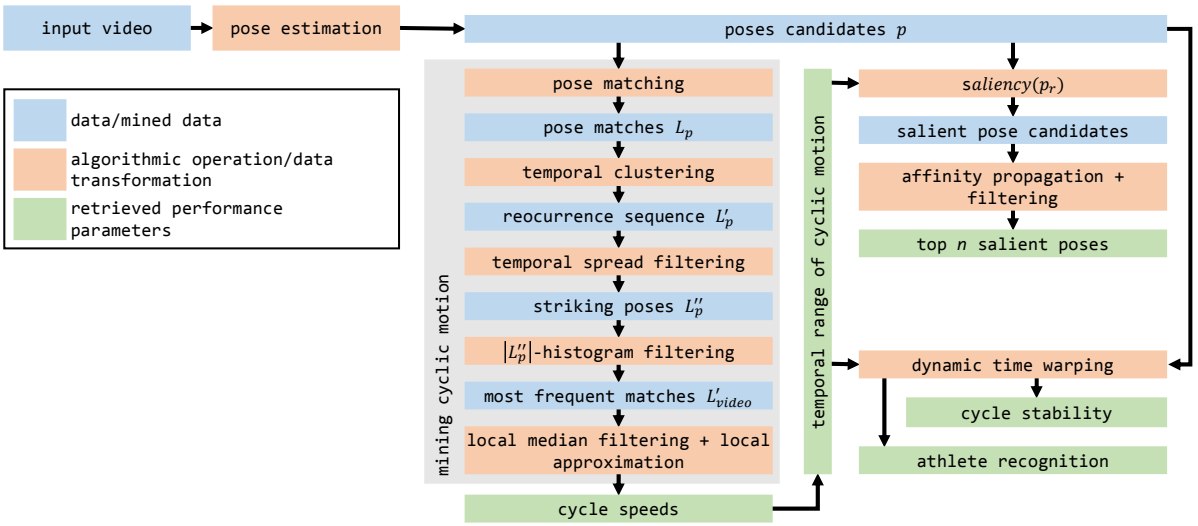


Figure 5: Block diagram that summarizes the proposed processing chain. It shows the different processing components along with their corresponding input and output data.

Mining Long Jump Pose Data

As a second example for pose data mining, we look at data of long jump athletes recorded at athletics championships and training events. Long jumping is different from swimming in many respects: Firstly, long jump features only semi-cyclic movement patterns. While the run-

up is composed of repetitive running motion, the final jump itself is strikingly different and only performed once per trial. Secondly, the action is performed over a complete running track and recorded by a movable camera from varying angles. Thirdly, spectators and other objects in the background along the track are likely to cause regular false detections of body joints. Our data consists of 65 videos recorded at 200Hz, where each video shows one athlete during a long jump trial from the side. The camera is mounted on a tripod and panned from left to right to track the athlete. The videos cover various athletes and six different long jump tracks. Figure 6 shows exemplary video frames from one trial. The long jump pose database consists of 45,436 frames with full-body pose estimates.

Automatic Temporal Classification of Long Jump Pose Sequences

Video based performance analysis for long jump athletes involves various time dependent measures like the number of steps until the final jump, the relative joint angles during the run-up, the vertical velocity during the final jump, and the flight phase duration. To obtain such measures automatically, pose information alone does not suffice. Instead it requires to pick the poses from the right phase of a long jump. Therefore, we present here how to mine the pose data to temporally identify the different phases of a long jump such that the phase specific performance measures can be computed from the detected poses. We partition a long jump action during one trial into a periodic and an aperiodic part. The periodic run-up consists of repeated *jumps* (the rear leg pushes the body upwards), *airtimes* (no contact with the ground) and *landings* (from first contact with the ground till the jump phase). The aperiodic part consists of the *flight phase* and the *final landing* in the sandpit. We annotated the long jump videos with respect to these five phases. Given a long jump video of length T and the extracted pose sequence $p_{1:T}$, our mining task is to predict the phase class $c_t \in C = \{\text{jump}, \text{airtime}, \ldots, \text{final landing}\}$ the athlete is in at each time step $t \in [1, T]$. Figure 6 depicts exemplary frames for each phase.

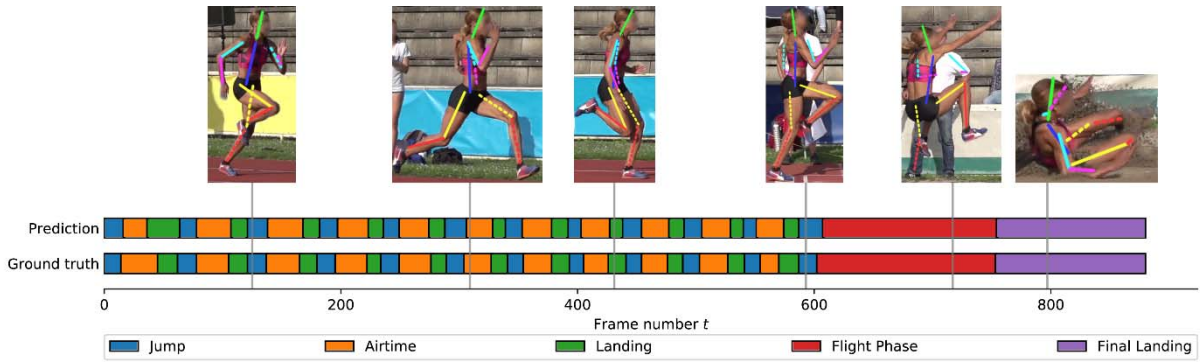


Figure 6: Qualitative comparison of predicted and ground truth long jump phases in one test video. Exemplary video frames and their estimated poses are depicted for each phase.

Pose Clustering: Similar to the cyclic strokes in swimming, we expect poses in identical long jump phases to be similar to each other. We expect this to be true even across videos of different athletes and slightly varying camera viewpoints. This leads to assumption 1: *Similar poses often belong to the same phase* (Asm. 1). Instead of learning a direct mapping from poses to the possible long jump phases C, we first partition the space of poses into a fixed number of subspaces. Henceforth, each pose is described by the discrete index of its subspace. As long as the subspace partition preserves similarity, we expect that the distribution of phases in one pose subspace is informative, i.e. non-uniform with respect to phase class c_t. Let S be the set of poses in our database. We perform unsupervised k-Medoids clustering on S with our normalized pose similarity measure from Equation (8) to create our subspace partition. The

clustering defines a function $h(p) \rightarrow [1, k]$ that maps a pose p to the index of its nearest cluster centroid. With Asm. 1 we define the probability $P(c|h(p))$ as the fraction of poses in cluster $h(p)$ labeled with phase c:

$$P(c|h(p)) = \frac{|\{p_i \in S|h(p_i) = h(p) \wedge c_i = c\}|}{|\{p_i \in S|h(p_i) = h(p)\}|} \tag{16}$$

Markov Representation of Long Jump Sequence: With Equation (16) we could already predict the phase for each pose in a video individually. However, noisy predictions and phase-unspecific poses may render Asm. 1 in a fraction of the poses as incorrect. We have to incorporate the complete pose sequence to obtain correct phase predictions even for frames with wrongly estimated or ambiguous poses. With the rigid long jump movement pattern and the chosen phase definitions, we can make two more assumptions: *An athlete stays in a phase for some time before entering a different phase. Subsequent poses are likely to belong to the same phase* (Asm. 2). *Also, the possible transitions between long jump phases are limited by a fixed sequential pattern* (Asm. 3).

We can model these assumptions by stating the temporal succession of long jump phases as the state transition graph in Figure 6. Each state corresponds to one possible phase. Asm. 2 and 3 are reflected by self-loops and a small number of outgoing edges at each state, respectively. At each time step t the athlete is in a phase which we cannot directly observe. However, the estimated and thus noisy pose at time t is observable . Combining the graph with emission probabilities $P(h(p)|c)$ and transition probabilities $P(c_{t+1}|c_t)$ we obtain a classical Hidden Markov Model. The emission probabilities $P(h(p)|c)$ can be computed as

$$P(h(p)|c) = \alpha \cdot P(c|h(p)) \cdot P(h(p)), \tag{17}$$

where α is a normalization constant. The transition probabilities are obtained similarly by counting the number of observed transitions in the dataset.

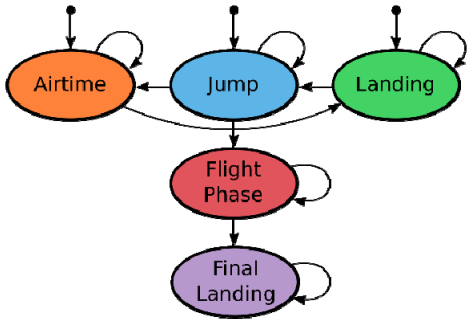


Figure 7: State transition graph modeling the possible transitions between the five long-jump phases. Edges with a black circle are possible entries into the graph.

Given a new long jump video and the corresponding pose sequence $p_{1:T}$ we first transform the sequence to the clustering-based discrete pose description $h(p_{1:T})$. We then use the Viterbi algorithm (Rabiner, 1989) for the most likely phase sequence $c^*_{1:T}$ with

$$c^*_{1:T} = \arg \max_{c_{1:T}} P\,(c_{1:T}|h(p)_{1:T}). \tag{18}$$

Results

Mining Pose Data of Swimmers

We tested our mining algorithms on a video database of 233 videos (see Table 1, 770237 video frames in total), showing over 130 different athletes swimming in two structurally different swimming channels. Videos were recorded either at 720×576@50i or at 1280×720@50p. The videos cover different swimmers in age, gender, physique, body size and posture, swimming in a swimming channel at different velocities between $1ms^{-1}$ and $1.75ms^{-1}$ and very different stroke rates. All mining was performed before any ground truth annotations were created.

Time-Continuous Cycle Speeds

The precision of the time-continuous cycle speed estimates expressed by the number of frames per cycle was evaluted by randomly picking one frame[2] from each video and annotating it manually with the actual stroke length. In 2 of our 233 video sequences, the mining system did not determine a cycle speed at the frame of the ground truth. For another 6 sequences the error in frames was larger than 2, while for the remaining 225 video sequences the average deviation in frames from the ground truth was 0.43 frames at 50 fps and 0.53, 0.32, 0.39 and 0.39 frames for breast, fly, back, and freestyle (see Table 1). This exceptional quantitative performance can intuitively be grasped by a human observer from the stroke length graphs in Figure 2. In these graphs it is also visually striking if something has gone wrong, which was the case for 6 videos. Figure 8 depicts one of the few videos where the stroke length was incorrectly estimated twice as high as it actually was due to difficulties in detecting the joints reliably.

Identify Cyclic Motion

We annotated all 233 videos roughly with the start and end time of the stroke. This sounds like an unambiguous task, but it was not: When the swimmer was starting the stroke out of the break-out from the dive, the starting point is temporally fluent over some range. We decided to be more inclusive and marked the point early. However, it was extremely difficult to specify when the athlete stopped the stroke. Many athletes were drifting partially out of the image when getting tired due to fast water velocities while still swimming. This violates the assumption of our pose detection system that the simmer has to be completely visible. We decided to mark the end of the stroke range when a swimmer was knees downwards out of the picture. This choice, however, did not fit breast stroke well: During a cycle the swimmer pulls the heels towards the buttock, bringing the feet back into the image, providing the system suddenly with a complete pose. We can see this effect in Table 1, there our algorithm over-detects up to 6% of the breast stroke range according to our early cut-off ground truth. This over-detection is primarily an artifact of how we determined the ground truth range of the stroke, but no real error. Our mining algorithm detected overall 89.5% of all ground truth stroke ranges, while only detecting 3.1% additionally outside. This performance is more than sufficient in practice.

Moreover, the length of the detected cyclic motion range(s) per video was an excellent indicator to identify unstable and/or erroneous pose detection results. A cyclic motion range of less than 10 seconds indicated that our automatic pose detection system had difficulties to detect the human joints due to strong reflections, water splashes, spray and/or air bubbles in the water. For these sequences determining the stroke cycle stability based on the identified temporally striking poses of the athlete does not make sense. Hence, in the subsequent

[2] For interlaced videos, the term *frame* always refers to half-frames

experiments, only cyclic motion sequences of 10 seconds or longer were used. This reduced the number of videos from 233 down to 213.

Table 1: Swimming test video database with mining results. *Video length* reports the minimum, median and maximum duration of the video clips in seconds. *Stroke length* reports the distribution of the 233 manually annotated stroke lengths, one annotation per video. The annotated frames were randomly picked within a video in order to introduce no bias. *CMRs* stands for *Cyclic Motion Ranges* and denotes those time periods of the video clips that shows the cyclic motion.

Stroke		Fly	Back	Breast	Free
# videos		80	28	79	46
	min	18.3	15.8	19.3	17.2
video length [s]	median	35.0	31.2	35.5	33.9
	max	72.7	49.7	85.7	83.8
	min	51	58	48	52
stroke length [# frames]	median	67	69	69	67
	max	101	85	119	108
	avg	0.53	0.32	0.39	0.39
stroke length error [# frames]	# >2 frames	6	0	1	0
	# not det.	0	1	0	1
% of detected cyclic motion ranges (CMRs)		96.0%	84.5%	91.1%	82.8%
% of erroneously detected CMRs in non-CMRs		1.8%	3.2%	6.0%	0.3%

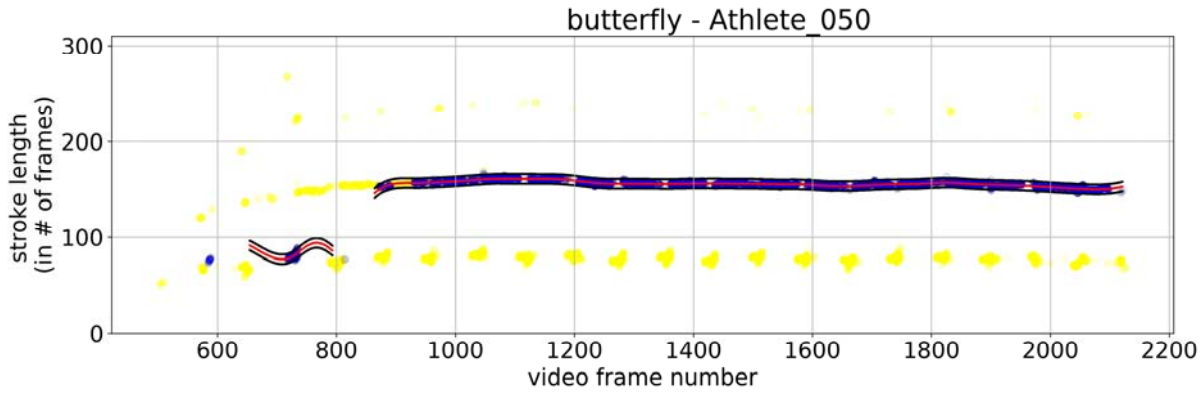


Figure 8: One of the 6 videos where the stroke length was incorrectly estimated twice as high as it actually was. The correct stroke length is below 100 frames.

Temporally Striking Poses

Poses which are temporally salient and unambiguously easy to determine by humans typically focus on one or two characteristic angles. An example is when the upper arm is vertical in freestyle (in the water) or backstroke (outside the water). Everything else of the pose is ignored. This is not how our temporally striking pose is defined: a pose which is easy to localize temporally by our system. Due to this mismatch between what the human is good at and our system, we only evaluate the temporally striking poses indirectly via their use to capture cycle stability.

Cycle stability

For each of the 213 videos we computed the stroke stability indicator value based on a single reference stroke clip. The reference stroke clip was selected by using the ground truth frame from the time-continuous cycle speed evaluation as the end point and by subtracting our estimated stroke length from that to compute the start frame. For each stroke we sorted the videos based on its stroke cycle stability indicator value and picked randomly one video from the top 20%, one from the middle 20% and one from the bottom 20%. We then asked a swim coach to sort these three videos based on his assessed stroke cycle stability. We compared the result to the automatically computed ordering:

Breast: There was an agreement in the ordering of the videos ranked 1st and 2nd. The athlete of the first video showed off an exceptionally stable stroke pattern. However, the video ranked 3rd was judged by the coach as being equivalent to the one ranked 2nd. The 3rd video is one of the instances there the swimmer is getting tired, drifting regularly with his lower legs out of the picture during the stretching phase in breast stroke. This explains the discrepancy between the judgement of the coach and our system.

Fly: The coach and the system agreed on the ordering. We also notice that our system was picking up those athlete, who were breathing every other stroke and exhibit a strong difference between the cycle with and without the breath. With respect to a two-cycle pattern their stroke was stable. Typically, coaches emphasize that there should be as little difference as possible between a breathing cycle and a non-breathing cycle.

Back: The coach and the system agreed on the ordering.

Free: The coach was ranking the second video as having a slightly better stroke stability than the first video. They agreed on the video ranked 3rd as the athlete was showing an unsteady and irregular flutter flick. The discrepancy between the first two videos can be explained by peculiarities of the video ranked 2nd: the water flow speed was higher than normal, leading to a slightly higher error frequency in the automatically detected poses.

Very similar results were obtained with the temporally striking poses as end points of the reference stroke clip.

Mining Long Jump Pose Data

Although we formulated our problem as a per-frame classification task, the predictions should reflect the sequential phase transitions as well as the length of each annotated phase. Therefore, we evaluate our phase detection mining by the standard protocol of average precision (AP) and mean average precision (mAP) for temporal event detection in videos (Gorban et al., 2015; Heilbron, Escorcia, Ghanem, & Niebles, 2015). For each video we combine sequential timestamps belonging to the same long jump phase c into one event $e_j = (t_{j,1}, t_{j,2}, c_j)$ with $t_{j,1}$ and $t_{j,2}$ being the start and stop time of the event. Let $E = \{e_j\}_{j=1}^{J}$ be the set of sequential events in one video. In the same manner we split the predicted phase sequence $c_{1:T}^*$ into disjoint predicted events e_j^*. Two events match temporally if their intersection over union (IoU) surpasses a fixed threshold τ. A predicted event e_j^* is correct if there exists a matching ground truth event $e_j \in E$ in the same video with

$$(c_j == c_j^*) \wedge \left(\frac{[t_{j,1}, t_{j,2}] \cap [t_{j,1}^*, t_{j,2}^*]}{[t_{j,1}, t_{j,2}] \cup [t_{j,1}^*, t_{j,2}^*]} > \tau\right). \quad (19)$$

For the evaluation protocol we now use AP to measure the precision in detecting events of one specific phase (i.e. discriminating one specific phase from all the other phases) and mAP as the

average AP over all defined phases. We optimize clustering parameters on a held-out validation set of six videos and use the remaining 60 videos to evaluate our approach using six-fold cross evaluation. We found the results to be rather insensitive w.r.t. the choice of clustering parameters, however. Table 2 depicts the results at a fixed $\tau = 0.5$ IoU threshold. We achieve a mAP of 0.89 for long jump phase detection. Due to their length and the unique poses observed during the flight and landing in the sandpit, these two phases are recognized very reliably with 0.94 and 0.97 AP, respectively. The phases of the periodic part show more uncertainty since each phase is considerably shorter and poses of the jump-airtime-landing cycle are more similar to each other. Figure 1 depicts qualitative results on one test video. Our method is able to reliably divide the cyclic run-up and the final flight phase and landing. Few predictions for the periodic phases are slightly misaligned, but the overall cyclic pattern is preserved. The phase predictions can directly be used to derive further kinematic parameters like the duration of the run-up and the number of steps. The results in Table 2 show that the run-up duration can be derived very accurately with an average deviation of 60ms. The correct number of steps is recovered in the majority of videos.

Table 2: Results of long jump phase detection (AP) with IoU threshold $\tau = 0.5$ (upper part) and the derived length and step count during the long jump run-up (lower part). The detection quality of the individual phases is measured by average precision(AP) and averaged for the computation of the mean average prevision (mAP).

Jump	0.84	Flight Phase	0.94
Airtime	0.91	Final Landing	0.97
Landing	0.80		
mAP			0.89
# videos with given abs. error in step count		$\lvert error_{steps} \rvert = 0$	53
		$\lvert error_{steps} \rvert = 1$	7
		$\lvert error_{steps} \rvert > 1$	0
Average abs. error in derived run-up length [s]			0.06

Discussions and Conclusion

Noisy pose data of individual sport recordings will soon be available in abundance due to DNN-based pose detections systems. This work has presented unsupervised mining algorithms that can extract time-continuous cycle speeds, cycle stability scores and temporal cyclic motion durations from pose sequences of sports dominated by cyclic motion patterns such as swimming. We also showed how to match pose clips across videos and identify temporally striking poses. As it has become apparent from the experimental analysis, results from our mining algorithms can be further improved if automatic pose detection systems focus on dealing with athletes that are not fully visible in the video.

We additionally apply our concept of pose similarity to pose estimates in long jump recordings. We model the rigid sequential progression of movement phases as a Markov sequence and combine it with an unsupervised clustering-based pose discretization to automatically divide each video into its characteristic parts. We are even able to identify short intra-cyclic phases reliably. The derived kinematic parameters show a direct application of this approach.

Acknowledgement

This research was partially supported by FXPAL during Rainer Lienhart's sabbatical. He thanks the many colleagues from FXPAL (Lynn Wilcox, Mitesh Patel, Andreas Girgensohn, Yan-Ying Chen, Tony Dunnigan, Chidansh Bhatt, Qiong Liu, Matthew Lee and many more) who greatly assisted the research by providing an open-minded and inspiring research environment.

References

Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2D human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (cvpr)*, 3686–3693.

Baysal, S., Kurt, M. C., & Duygulu, P. (2010). Recognizing human actions using key poses. In *20th International Conference on Pattern Recognition (ICPR)*, 1727–1730.

Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A. L., & Wang, X. (2017). Multicontext attention for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1831–1840.

de Souza Vicente, C. M., Nascimento, E. R., Emery, L. E. C., Flor, C. A. G., Vieira, T., & Oliveira, L. B. (2016). High performance moves recognition and sequence segmentation based on key poses filtering. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1–8.

Einfalt, M., Zecha, D., & Lienhart, R. (2018). Activity-conditioned continuous human pose estimation for performance analysis of athletes using the example of swimming. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 446–455.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science, 315* (5814), 972–976.

Gorban, A., Idrees, H., Jiang, Y.-G., Roshan Zamir, A., Laptev, I., Shah, M., & Sukthankar, R. (2015). *THUMOS challenge: Action recognition with a large number of classes.* http://www.thumos.info/.

Heilbron, F. C., Escorcia, V., Ghanem, B., & Niebles, J. C. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 961–970.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, *10*, 707–710.

Li, H., Tang, J., Wu, S., Zhang, Y., & Lin, S. (2010). Automatic detection and analysis of player action in moving background sports video sequences. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 20* (3), 351–364.

Lv, F., & Nevatia, R. (2007). Single view human action recognition using key pose matching and viterbi path searching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.

Meyers, E. W. (1994). A sublinear algorithm for approximate keyword matching. *Algorithmica, 12* (4-5), 345–374.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *European Conference on Computer Vision (ECCV)* (pp. 483–499). Cham: Springer International Publishing.

Pansold, B., Zinner, J., & Gabriel, B. (1985). Zum einsatz und zur interpretation von laktatbestimmungen in der leistungsdiagnostik. *Theorie und Praxis des Leistungssports*, *23* , 98–195.

Pyne, D. B., Lee, H., & Swanwick, K. M. (2001). Monitoring the lactate threshold in world-ranked swimmers. *Medicine and Science in Sports and Exercise, 33* (2), 291–297.

Rabiner, L. R. (1989, Feb). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77* (2), 257-286. doi: 10.1109/5.18626

Ren, C., Lei, X., & Zhang, G. (2011). Motion data retrieval from very large motion databases. In *International Conference on Virtual Reality and Visualization (ICVRV)*, 70–77.

Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20* (1), 23–38.

Sedmidubsky, J., Valcik, J., & Zezula, P. (2013). A key-pose similarity algorithm for motion data retrieval. In *15th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 669–681.

Victor, B., He, Z., Morgan, S., & Miniutti, D. (2017). Continuous video to simple signals for swimming stroke detection with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 122–131.

Vögele, A., Krüger, B., & Klein, R. (2014). Efficient unsupervised temporal segmentation of human motion. In *Proceedings of the ACM Siggraph/Eurographics Symposium on Computer Animation*, 167–176.

Wang, C., Wang, Y., & Yuille, A. L. (2013). An approach to pose-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 915–922.

Wei, S.-E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4724–4732.

Wu, C., Ma, Y.-F., Zhan, H.-J., & Zhong, Y.-Z. (2002). Events recognition by semantic inference for sports video. In *IEEE International Conference on Multimedia and Expo (ICME), 1*, 805–808.

Yang, W., Li, S., Ouyang, W., Li, H., & Wang, X. (2017, Oct). Learning feature pyramids for human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*.

Zecha, D., Eggert, C., & Lienhart, R. (2017). Pose estimation for deriving kinematic parameters of competitive swimmers. In *Computer Vision Applications in Sports, part of IS&T Electronic Imaging* (pp. 21–29). Society for Imaging Science and Technology.