Kinematic Pose Rectification for Performance Analysis and Retrieval in Sports

Dan Zecha, Moritz Einfalt, Christian Eggert and Rainer Lienhart
Multimedia Computing and Computer Vision Lab
University of Augsburg
{dan.zecha,moritz.einfalt,christian.eggert,rainer.lienhart}@informatik.uni-augsburg.de

Abstract

The automated extraction of kinematic parameters from athletes in video footage allows for direct training feedback and continuous quantitative assessment of an athlete's performance. Recent developments in the field of deep learning enable the measurement of kinematic coefficients directly from human pose estimates. However, the detection quality decreases while errors and noise increase with the complexity of the scene. In aquatic training scenarios, for instance, continuous pose estimation suffers from several orthogonal errors like switched joint predictions between the left and right sides of the body. In this paper, we analyze different error modes and present a rectification pipeline for improving the pose predictions using merely joint coordinates. We show experimentally that joint rectification equally improves the detection of key-poses, which are essential for a continuous qualitative performance assessment and pose retrieval, as well as posture visualization for quantitative training feedback.

1. Introduction

We study the problem of human pose retrieval in context of human motion analysis for top-tier athletes. During training sessions qualitative and quantitative video analysis is a common tool for giving feedback and thereby improving the performance of athletes. An athlete is filmed during a training session and the footage is evaluated afterwards by coaches and training scientists. Evaluation includes expert judgment about the execution of motion for a qualitative feedback as well as quantitative assessment. For swimmers, quantitative evaluation includes kinematic parameters like stroke rate, leg kick frequencies and inner-cyclic execution times of motion intervals. The boundaries of motion intervals are defined by human poses of specific interest - denoted as key-poses - that have to be determined manually through a highly time consuming and exhaustive search.

Recent advances in deep learning assisted human pose estimation have the potential to automate and thereby accelerate this process. The continuous estimation of the human pose in videos tags a persons joints and allows for retrieving specific key-poses automatically. While state-of-the-art pose estimation algorithms often perform well on benchmark video footage and less complex scenes, we found that in visually difficult training scenarios, the output of such algorithms can be inaccurate and partially incorrect. Human pose estimation in aquatic environments is exacerbated by noise from air bubbles, water splashes, constant self-occlusion and refraction, which often impede the prediction of the human pose and consequently the retrieval of key-poses. For automated performance analyses to be useful for training feedback, expectations of coaches have to be met with appropriately precise results.

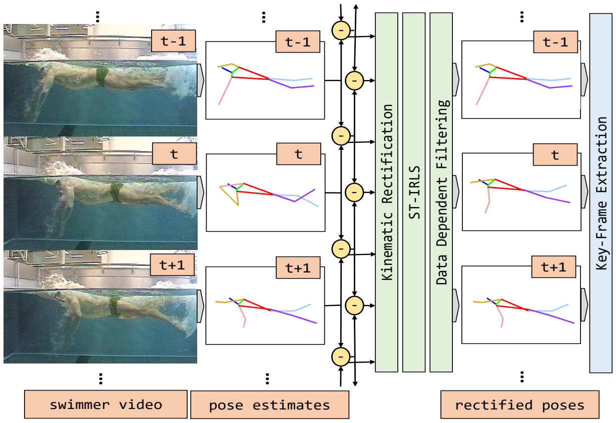

Figure 1: The proposed kinematic rectification pipeline (green) improves joint localization by enforcing temporal consistency between consecutive pose estimates.

We present a rectification pipeline for joint coordinate estimates of swimmers with the objective to improve on the prediction of performance indicators (Figure 1). Working solely on noisy joint coordinates obtained from a pose estimation system, an analysis of errors modes illustrates three typical detection errors: joint swaps, detection outliers and posture specific joint prediction noise. A three-stage pipeline is proposed to rectify aforementioned three

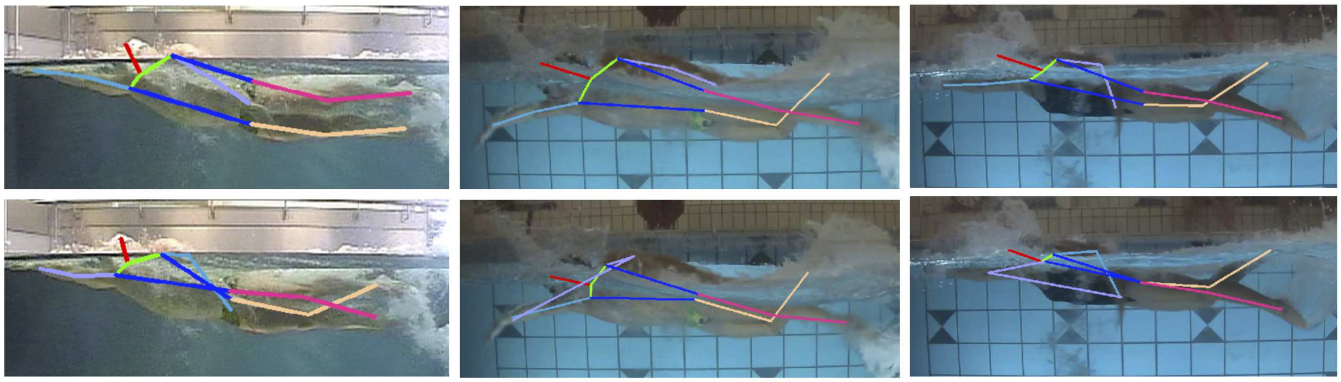

Figure 2: Qualitative examples of different errors. The first row depicts ground truth annotations, the second row pose estimates on a refined version of [19]. Left: arms and ankles are swapped. Middle: left wrist outlier and poor elbow prediction. Right: swaps and outliers occur together.

orthogonal errors: (1) A joint-kinematic based optimization identifies and corrects joint swaps. (2) A robust regression scheme identifies outliers and replaces them with motion-consistent values. (3) An adaptive filtering targets specific kinematic states of an athlete's pose to reduce the variance of joint predictions. Alltogether our pipeline improves joint estimates up to 5% for individual swimming styles. Also, temporal prediction of key-frames directly benefits from enforcing temporal consistency, while at the same time discarding up to 7% false positive events.

Contributions. (1) While complex approaches are often built on image data, we present an error analysis and rectification pipeline merely based on joint coordinate predictions. The rectification pipeline improves joint estimates considerable with little training and negligible inference costs. (2) The reported orthogonal joint errors are not swimming specific, but can occur in many other complex scene, too. Thus, the proposed rectification pipeline is universally applicable despite our focus on swimming in this paper. (3) With the instrumentation of training grounds with cameras, sport specific databases of training footage grow each day. However, obtaining large quantities of high precision annotations to training sophisticated machine learning algorithms remains a time-consuming and tedious challenge. Especially in less popular sports, this effort is often not feasible. We show that the proposed pipeline for improving continuous pose predictions can be trained with a comparatively small set of annotated training data. (4) The rectified pose estimates lead to a better pose visualization, thus assisting coaches with a qualitative pose assessment by removing joint localization jitter between frames.

2. Related Work

Human pose estimation in 2D images has recently experienced a shift from deformable part models based on hand-crafted features[2, 22, 13] to using deep neural networks for learning visual appearance [21, 17] and structure [6, 19, 20] of the human pose. Deep architectures have also been proposed for pose estimation in videos, for example by stacking multiple contiguous frames [15] as a parallel input for a deep network or by means of a recurrent network structure [4]. In context of sports, pose estimation has been researched by [10], who combine global and local pose estimation to refine the location an athlete's joints. [8] propose a generative approach for estimating the human pose in TV footage. The extraction of kinematic parameters of athletes from video footage, specifically stroke rates of swimmers, was recently researched by Victor et al. [18], who perform stroke frequency detection for athletes in a generic swimming pool by means of a deep neural network architecture for key-pose regression. [24, 23] derive additional parameters from swimmers by determining interval lengths and frequencies from multiple keyframes.

The topic of cleaning pose data was recently addressed by [12], who propose a recurrent framework for finding outliers in motion capture data. Within this field, methods like Kalman filters [3], dimensionality reduction [1] or conditional Boltzmann machines [16] have been successfully applied to outlier detection. Data-dependent filtering was addressed in [7], who introduce the concept of filter forests for learning data-dependent filters for noise reduction in images.

3. Problem Description and Analysis

Given video footage of a performing athlete, a state-of-the-art pose estimation system estimates his/her pose for each frame. A pose estimate is commonly comprised of a set of joint coordinates for head, neck, shoulders, elbows, wrists, hips, knees and ankles. We use Convolutional Pose Machine (CPM, [19]) models throughout this work to estimate poses. They were initially pre-trained on the Leeds Sports Pose dataset [11] and refined on our sport

	μ_x	μ_y	σ_x	σ_y
left wrist	-3.99	0.06	17.65	12.82
right wrist	-6.65	0.64	20.32	12.92

Table 1: Mean μ and standard deviation σ of normalized error between predicted wrist location and ground truth in freestyle in our dataset. As the swimmer is swimming from right to left, the right arms is often occluded by his/her body. Thus, the error for the right wrist is higher.

% rand swaps	0%	1%	2%	3%	4%	5%
$RMS(\|p_i'(t)\|)$	4.4	5.1	6.5	7.3	8.7	9.5
$RMS(\|p_i''(t)\|)$	4.4	6.0	8.7	10.5	13.2	14.9

Table 2: RMS of joint velocity and acceleration for freestyle swimmers with varying fractions of simulated random joint swaps.

specific footage, one fine-tuned model for each of the four main swimming styles (freestyle, backstroke, butterfly and breaststroke).

Frame-based pose estimators have the disadvantage of ignoring consistency between temporally contiguous poses. While CPMs can nevertheless perform surprisingly well on video footage of standard scenes, visually challenging footage of swimmers may still confuse the fine-tuned and adapted pose detectors due to heavy splashes, water bubbles or refraction, producing many false estimates. Figure 2 depicts some qualitative examples. The top row visualizes ground truth annotations obtained from a human expert, the bottom row depicts the CPMs' pose estimates. Typically errors include complete swaps of left and right body sides (left image), single joint outliers (middle) and a combination of both (right). Pose estimation errors are sometimes isolated phenomena within a long sequence, but more often than not occur in clusters. Figure 4 (left) depicts the x and y coordinates of elbows of a freestyle swimmer. In this example, a joint swap occurs for single frames in the sequence as well as for multiple contiguous frames. Additionally, three single joint outliers (left elbow, blue) blend in together with switched joints. Apart from obvious pose estimation errors, extremity joint predictions can be strongly influenced by the orientation of their respective limbs. We performed the following experiment: From all wrist predictions of freestyle swimmers in our dataset, we subtract the true (ground truth) wrist positions. Additionally, we normalized the orientations of all predictions with respect to the true orientation of wrist and elbow. Table 1 shows the means and standard deviations of the errors in predicting the x and y coordinates of both wrists. First, we observe that the error mean is not zero-centered, but biased towards the elbows joint. Second, the standard deviation in x direction is much larger than in y direction, i.e., the prediction error along the limb is larger than the prediction error orthogonal to that limb. We observe the same effect for all extremity joints. In this paper, we propose a pipeline to improve on the aforementioned problems. First, we present an optimization for untangling joint swaps. Second, we approach the problem of filtering coordinate outliers and signal noise with a novel method for robust regression. Third, we propose data-dependent filters for fine-tuning joint coordinates. All three stages work solely on the noisy raw output of a pose detection system, i.e., on joint coordinates.

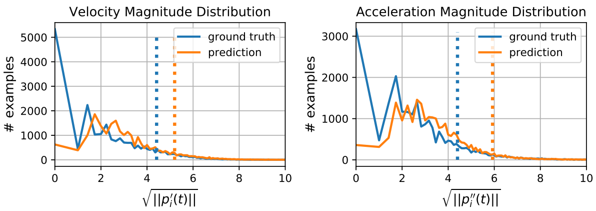

Figure 3: Distribution for velocity and acceleration of joints for the ground truth (blue) and predictions (orange) from finetuned [19]. Dashed lines indicate distribution means.

4. Joint Refinement

We define configurations of the human pose as a graph $\Gamma = (V, E)$ with vertices corresponding to $N = |V|$ human joints identified by an index $i \in \{1, ..., N\}$. V includes the head, neck, shoulders, elbows, wrists, hips, knees and ankles of a person. We denote two joints of the same type, e.g., the left and right wrists, as partner joints. The subset $\hat{V} = V \setminus \{head, neck\}$ contains all joints with a semantic partner. Joints i and j are connected via edges $(i, j) \in E$ in Γ according to the skeletal structure of the human body depicted in Figure 2.

We describe the temporal locations of a predicted joint i by a function $p_i(t) : \mathbb{N}_0 \rightarrow \mathbb{R}^2$ with $t \in \{0, ..., T\}$. Each function value at $p_i(t) = (x_i(t), y_i(t))^T$ describes the location of a joint i in frame t of a video. $P(t) = [p_1(t)^T, ..., p_N(t)^T]^T$ is used to describe a whole pose configuration at time t. A slicing operator ":" is used to extend this vector to a matrix $P(t_1 : t_2) = [P(t_1), P(t_1 + 1), \cdots, P(t_2 - 1), P(t_2)]$, combining all pose coordinates within a timeframe $[t_1, t_2]$. The ground truth related to $p_i(t)$ and $P(t)$ is denoted by the time series of ground truth joint coordinates $g_i(t)$ and ground truth configurations $G(t)$.

Preprocessing. All raw pose configurations $\hat{p}_i(i)$ are resized relative to a reference upper body size d_{ref}. In our experimental setup, the size of the athlete in a video is approximately constant. Hence, let d_{up} is estimate by the median length of an athlete's upper body size in a video, determined by the distance between right shoulder and left hip. Then, preprocessed $p(i)$ is defined as

$$p_i(t) = \frac{d_{ref}}{d_{up}} \hat{p}_i(t), \forall i \in V. \qquad (1)$$

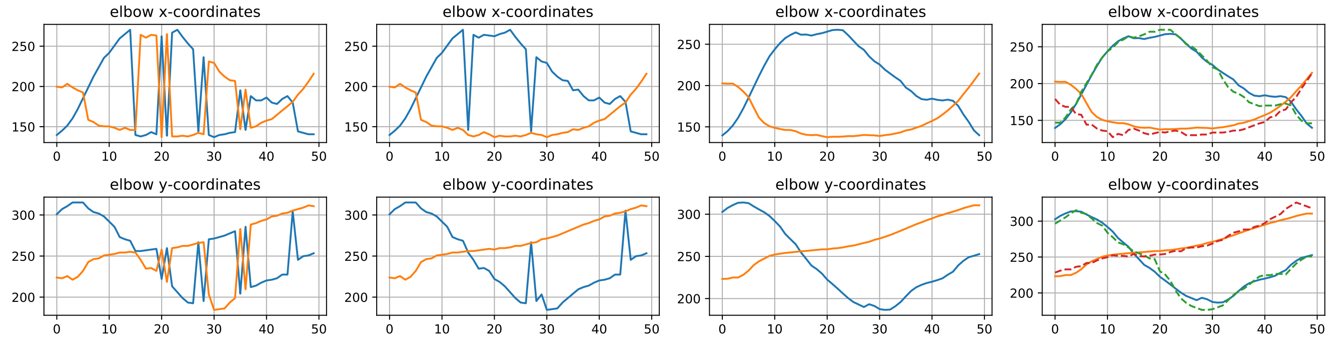

Figure 4: A sequence of joint coordinates of a freestyle swimmer's left (blue) and right (orange) elbow. Top row depicts x coordinates, bottom row y coordinates. From left to right: Original sequence predicted, after swap correction, after outlier and noise correction, after data-dependent filtering. Dashed lines in last column represent the ground truth.

The estimate of d_{up} may be more complex if athletes are filmed in different training situations. For example, if pan shots are used and the size of the athlete in the footage changes over time, d_{up} needs to be continuously estimated, e.g., by a smoothing spline over predicted upper body lengths.

4.1. Untangling Joint Swaps

As discussed before, state-of-the-art pose detection systems sometimes erroneously swap joint assignments. For instance, the left knee is recognized as the right knee, while the right knee is identified as the left knee. This section describes our robust approach to correct most of these errors. Our approach is based on the following observation: Erroneous joint swaps increase the observed magnitudes of joint velocities and joint accelerations. Table 2 shows the average joint velocities and average joint accelerations of the training data set for varying percentages of noise, i.e., random joint swaps, ranging from 0% to 5%. The noise was generated by randomly picking joint pairs of $\hat{V}$ from the training data ground truth and swapping the ground truth joint positions until the desired noise level was reached. As can be seen in Table 2, the observed average joint velocities and average joint accelerations are increasing with the noise level. The same increase can be observed in the actual joint detections of our human pose detector with respect to the ground truth joint locations (see Figure 3). Some of this increase is due to the faulty joint swaps.

Based on this observation, our approach is to minimize a loss comprised of the squared magnitudes of joint velocities and joint accelerations by swapping the assignments of associated joints. Thus, we propose the following loss function f for finding swapped joint predictions of an athlete for a predicted pose sequence of length T:

$$\min f(T) = \sum_{t=2}^{T} \sum_{i \in \hat{V}} (\|p_i'(t)\|^2 + \lambda \|p_i''(t)\|^2) \qquad (2)$$

The numerical derivative of the function $p(t)$ is approximated by $p'(t) \approx p(t) - p(t-1)$. For the joint location functions $p_i(t)$, the first derivative $p_i'(t)$ equals the velocity in [pixels /framerate^{-1}], the second derivative $p_i''(t)$ corresponds to the joint acceleration with unit [pixels /framerate^{-2}]. The coefficient λ balances the magnitudes of both objective terms. As evident in our error simulation in Table 2, the magnitude ratio changes depending on the error bias of the pose detector, which has to be determined on a dataset specific validation set.

At each timestep t of the minimization of the loss f, we only consider joint placements that are contained in our original pose predictions. For example, the position of the right wrist can only be one of two possible solutions: either the position it already has or the position of the left wrist. Whatever solution is assigned to the right wrist, the second solution is consequently assigned to the left partner wrist. For each of the six partner joints, there are 2 possible solutions, leaving us with $n = 2^6$ possible swaps per frame that have to be evaluated with Equation 2. Minimizing Equation 2 directly would lie in $O(n^T)$ for a video of length T, and is therefore too expensive. This problem, however, structurally resembles the well studied optimization problem of deformable part models [9]. It is decomposable into a sequence of optimization stages, where each stage is only dependent on previous stages via the numerical derivatives. This can be solved by finding the Viterbi path via dynamic programming in $O(n^2T)$. For our problem, the Viterbi path contains the sequence of pose estimates that minimize Equation 2. It is computed as follows:

We start with the third pose estimate, the first for which joint velocities and accelerations can be computed, and add it to the empty Viterbi path. From this initial estimate, all possible joint swaps of partner joints for the sequentially next pose estimate are generated and evaluated by Equation 2, taking into account the last pose estimates already in the solution path. The configuration with the smallest error is choosen and added to the solution path. We only accept solutions with joint swaps if the joint velocities and accelerations for a joint are above a threshold, determined by a validation set of ground truth pose sequences. After the last pose estimate was added, the Viterbi path contains all poses minimizing Equation 2. One example of a resulting Viterbi path for two elbow joints is depicted in Figure 4 (middle). Joint swaps have been eliminated, only single outlier are left in the sequence.

4.2. Outlier interpolation and noise reduction

After untangling swaps by minimizing the loss in Equation 2, we are left with outliers. In order to detect and rectify them, we have to account for noisy joint detections over time. We propose a windowed least squares robust regression for finding and replacing outliers and increasing the signal to noise ratio for each $p_i(t)$. Hereto, the time series of 2D locations $p_i(t)$ is separated into their respective coordinate time series $x_i(t)$ and $y_i(t)$. Each time series is segmented into pieces of length m, overlapping at $\lfloor m/2 \rfloor$ values. A good value for m depends on the frequency of arm strokes or leg kicks. We found that a window size of $m = 21$ is a good value for both, independent of the swimming style.

For each segment, we fit a polynomial using iteratively reweighted least squares regression (IRLS, [5]). IRLS is an algorithm that iteratively performs polynomial regression on a set of weighted sample points. In each iteration, the samples are reweighted according to the inverse of their residuals. A new polynomial is then fitted with updated weights. Consequently, the influence of outliers is mitigated by lowering their weights. We apply IRLS with a polynomial of degree three to each window of size m. Each value in the final fit is weighted with a triangular Bartlett window of size m. All weighted polynomials are added up according to the position of the original segments in $p_i(t)$, leaving us with a robust regression $s_i(t)$ for each $p_i(t)$. We denote this process a Short-Time Iteratively Reweighted Least Squares (ST-IRLS) robust regression of a time series.

The ST-IRLS estimates $s_i(t)$ are now used for two purposes: Finding and correcting outliers and reducing the noise in the signal. Therefore, for each value in $p_i(t)$, we set

$$p_i(t) = \begin{cases} s_i(t), & \text{if } |s_i(t) - p_i(t)| > \tau \\ p_i(t) - \eta(p_i(t) - s_i(t)) & \text{otherwise} \end{cases} \tag{3}$$

The first line in Equation 3 replaces the original joint estimate with the value of the ST-IRLS estimate if $p_i(t)$ is more than τ away from $s_i(t)$. The second line reduces noise in the signals by pulling $p_i(t)$ closer to its regression result $s_i(t)$ by reducing the distance by a fraction η of the error $p_i(t) - s_i(t)$. The parameter τ is dataset specific and adaptively set through an established ratio of assumed outliers per window m. This ratio and η are determined via a parameter search on a validation set.

4.3. Data Dependent Joint Refinement

With larger joint estimation errors mostly out of the way, we propose *adaptive* temporal filtering for improving joint predictions on a finer scale. Adapting to the observed joint velocities in a pose seems to be a natural choice. Thus, our goal is to learn different temporal pose filters for different clusters of observed (i.e., predicted) joint velocities in predicted poses, allowing the filter to better fit to the current situation than a single static filter.

The velocity of a joint is physically defined as the first derivative of the location $p_i'(t)$ and is approximated here by $p_i'(t) \approx p(t) - p(t-1)$. Unfortunately, noise in joint locations is amplified by taking the derivative, leading to potentially unstable velocity estimates. Therefore, only in the case of the ground truth annotations we compute the velocities directly from the joint locations, while in the case of the predicted joint locations we compute a more stable velocity estimate from our smoothed trajectory estimates $s_i(t)$ and not from our refined estimate $p_i(t)$ from Subsection 4.2.

The velocity clusters are computed as follows: Given sequences of ground truth annotated poses $G(t)$ in our validation set, we compute the velocities for all N joints in $G(t)$, yielding a velocity feature $V_G(t) = [g_1'(t), \cdots, g_N'(t)]^T$. We cluster these velocity features into 11 clusters with k-means clustering function $\mathcal{K}(.)$, resulting in 11 different cluster labels. We now determine the cluster id c of each smoothed pose and assign this very cluster id c to the respective pose estimates $P(t)$. Each predicted pose estimate in the validation set is thereby augmented with a cluster id c, i.e., $\mathcal{K}(V_P(t)) = c$.

Our cluster id specific filters are now learned as follows: We sort all poses with the same cluster assignment c in separate training sets S_c:

$$S_c = \{(P(t-\rho : t+\rho), G(t))\,|\mathcal{K}(V_P(t)) = c\}\,. \tag{4}$$

For the sake of readability, we write $P_{[t]} \equiv P(t-\rho : t+\rho)$ to describe $P(t)$ in its temporal context of width $2\rho + 1$. Each S_c contains tuples of training data $P_{[t]} \in \mathbb{R}^{2N \times (2\rho+1)}$, i.e., coordinates of contiguous pose estimates around $P(t)$, together with a vector of ground truth annotations $G(t) \in \mathbb{R}^{2N}$ for the pose at time t.

For each training set S_c, we improve each $P(t)$ by finding a mapping W_c^* that minimized the distance between $P(t)$ and $G(t)$, i.e.,

$$W_c^* = \arg\min_{W_c} \left\| \sum_t diag([P_{[t]}|\mathbf{1}] \cdot W_c) - G(t) \right\|^2 \tag{5}$$

Note that this formulation extends matrices $P_{[t]}$ with bias terms to allow for a constant offset between pose estimates

	free	back	fly	breast
# sequences	24	28	26	26
# pose annotations	1883	2268	2281	2100
# videos	97	74	80	80
# key-poses	1504	1172	1920	1280

Table 3: Statistics of datasets and annotations.

and ground truth coordinates. It can be shown that minimizing this problem is equivalent to solving several subproblems, namely

$$w_i^* = \arg\min_{w_i} \|[D_i \cdot w_i - G_i\|^2 , \tag{6}$$

where D_i are the i-th rows from all $[P_{[t]}|\mathbf{1}]$ stacked into one data matrix, w_i is the i-th column of W_c and G_i are the corresponding target coordinates $g_i(t)$ stacked into one vector. Depending on the number and distribution of training samples in S_c, several solutions for Equation 6 are possible: (a) Equation 6 is overdetermined. Then there exists a closed form solution $w_i^* = (D_i^T D_i)^{-1} D_i^T G_i$. (b) The system is underdetermined, which may happen if an insufficient number of training examples are assigned to S_c. In this case, we set $w_i^* = (0, \cdots, 0, 1, 0, \cdots, 0, \mu)^T$, where the bias term is initialized with the mean deviation μ between the predicted coordinates and ground truth. (c) The training set S_c is empty. In this case, all w_i^* are are identity mappings, i.e., $w_i^* = (0, \cdots, 0, 1, 0, \cdots, 0)^T$.

5. Experiments

Competitive swimming covers four different swimming styles: breaststroke (breast), butterfly (fly), freestyle (free) and backstroke (back). The first two are termed *symmetrical* styles, because both halves of the body perform the same motion at all times. The later two are denoted *anti-symmetrical*: The left half of the body performs a motion that is mirrored approximately half a cycle later by the right half of the body and vice versa. If viewed from the side, both body halves in symmetric styles are mostly indistinguishable due to self occlusion, hence joint swaps appear to a lesser extend. On the other side, anti-symmetrical styles are affected by joint swaps and outliers. This may change with camera perspective. However, we only consider a side view on athletes in this work.

A *key-frame* depicts an athlete in a *key-pose*. A key-pose in turn is defined by means of a key-pose feature. Features can be angles between certain body parts, angles of body parts relative to the camera, contact points of body part and water surface or moments where a specific motion starts or ends. Key-poses are commonly defined by human experts based on training requirements.

Dataset. Our datasets are comprised of swimming channel footage. All sequences and videos depict swimmers of different age, stature, gender and body size in two different swimming channels. The athletes are filmed from a side view through a glass wall, hence the whole body - above and below the water surface - is always visible. We distinguish between two overlap-free datasets in our experiments. The first set is contains 104 fully annotated *sequences of human poses* with 50 to 100 consecutive images each, giving us a total of 8532 annotated video frames. The second dataset covers 331 swimmer *videos* for key-pose retrieval. Each video is between 20 and 30 seconds long and has a framerate of 50 fps. Instead of ground truth pose annotations, we obtained annotations for key-poses from human experts. These annotations are spread over two to four full swimming cycles per video, summing up to a total of 5876 key-poses. Key-poses cover the angle of the upper arm for freestyle, backstroke and butterfly and the angle between upper and lower arm and leg for both breaststroke and butterfly. We give a visual overview over evaluated key-poses in Figure 7. Table 3 summarizes the dataset size and the number of annotations available for our experiments.

Metrics. For the quantitative evaluation of pose estimates, we apply the Percentage of Correct Keypoints (PCK, [14]) measure. PCK counts a joint as correctly localized if the euclidean distance to the ground truth annotation does not exceed a fixed fraction α of the upper body size, which is defined by the euclidean distance between right shoulder and left hip. Commonly, thresholds $\alpha = 0.1$ ($PCK@0.1$) and $\alpha = 0.2$ ($PCK@0.2$) are evaluated for comparing the performance of pose estimation systems.

We evaluate a correct detection of a key-pose using a related metric, denoted as Percentage of Correct Key-Poses (PCKP, [24]). Similar to PCK, a key-pose is counted as correctly identified if the absolute temporal deviation of its key-frame from the annotated ground truth frame is smaller than a fraction β of one swimming cycle length. This metric takes into account that key-poses are often difficult to identify, even by experts. Multiple human annotations of the same key-pose can derive up to ± 2 frames from each other, which translates to $\beta \approx 0.03$. Some annotations in our dataset miss a second key-pose as a closing boundary of the interval that defines a complete cycle, hence we will specify the derivation from the ground truth not by $beta$, but w.l.o.g. in full frames in this paper.

5.1. Joint Localization

The initial pose estimates are predicted by a 3-staged CPM as described in [19]. The CPM model is initially pretrained on the Leeds-Sports-Pose dataset [11]. As we wish to compare the PCK of our approach with the CPM baseline but only have a small, fully annotated dataset of motion sequences available, we have to evaluate our pipeline on our training data. Therefore, we refine a CPM on each swimming style by fine-tuning it on our fully annotated se-

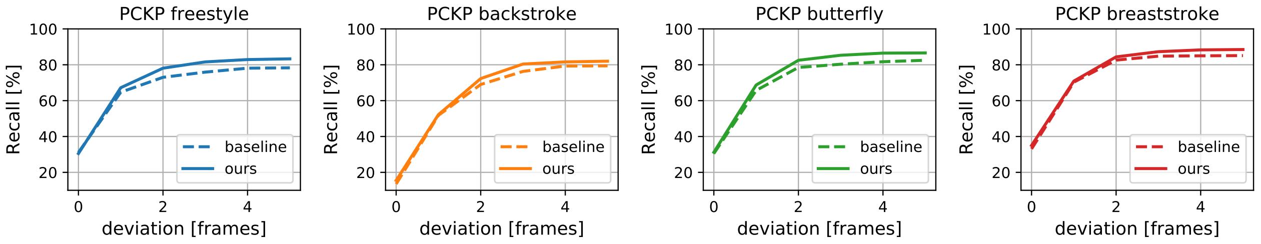

Figure 5: Percentage of correct key frames. From left to right: freestyle, backstroke, butterfly, breaststroke.

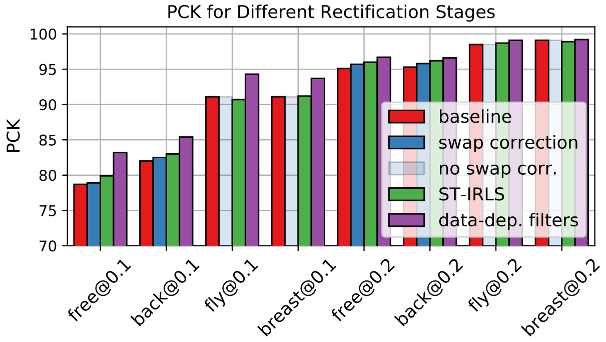

Figure 6: PCK@0.1 and PCK@0.2 for all four major swimming styles.

	free	back	fly	breast
CPM baseline	14.81	17.90	7.07	6.14
swap correction	12.50	15.66	6.91	6.14
outlier corr.	11.60	14.40	6.67	6.28
noise reduc.	11.39	14.20	6.64	6.18
data-dep. filt.	**10.74**	**11.34**	**5.78**	**5.56**

Table 4: RMS values for squared euclidean distance between prediction and ground truth for all partner joints on arms and legs.

quences using a *3-fold cross validation*. The data is split into 3 partitions. One partition is kept as the test set while the model is refined using the remaining partitions for training and validation. Hereby, we obtain pose estimates for each frame in our fully annotated pose sequences and can compare our approach directly without having to worry about overfitting. The PCK on the inital estimates serves as our baseline and is depicted in Figure 6 (red). We found that the symmetrical swimming styles already work very well: the PCK@0.1 is beyond 90%; almost all joints (>98.5%) are predicted within 0.2 of the upper body size. The anti-symmetrical swimming styles are more difficult to predict precisely, with a PCK@0.1 of 78.7% for freestyle and 82% for backstroke. Both hardly exceed 95% PCK@0.2.

We now evaluate each step in our pose rectification pipeline separately. Again, due to the small database of pose sequences, we apply a leave-one-out training scheme. One sequence is kept for testing, while the pipeline is trained on the remaining sequences. The joint swap optimization problem, outlier rectification, noise reduction and data-dependent filtering is applied to all swimming styles, although model specific parameters are optimized for each style individually. Figure 6 depicts the PCK for all stages. We observe an increase in PCK for almost all stages in our rectification pipeline. Anti-symmetrical swimming style benefit from untangling swaps (up to 0.8 % PCK). We choose the parameter τ for ST-IRLS depending on the swimming style to adaptively account for outliers per ST-IRLS window. Block size m, polynomial degree and smoothing parameter η are determined by a grid search on a separate validation set per swimming style. With this setup, we obtain a PCK increase of up to 1%.

The largest increase of up to $+3.5\%$ is gained from data-dependent filtering. As for ST-IRLS, an optimal number of clusters is determined via parameter search on a validation set. We found that this number is larger than 9 clusters and does not exceed 13. Overall, we obtain a considerable increase in PCK of up to 5% per swimming style for PCK@0.1 and +2% for PCK@0.2 for freestyle and backstroke. As butterfly and breaststroke were already performing very well at this range, performance increase for both is only marginal.

We additionally evaluate the Root Mean Square (RMS) of euclidean distance between joint prediction and ground truth in Table 4. This table should give the reader a better intuition of the decreasing distance between prediction and ground truth. Equivalent to 6, we observe a decrease of the error for each rectification step, although this decrease is much smaller for symmetrical swimming styles.

5.2. Key-Pose Retrieval

The initial motivation for improving the joint predictions of swimmers was to improve the identification of specific key-poses. Key-poses serve as interval boundaries to inner-cyclic intervals that are timed and give important insight into the quality of a swimming motion. The evaluation of inner-cyclic intervals is a valuable tool for improving the performance of top level swimmers. We approach the prob-

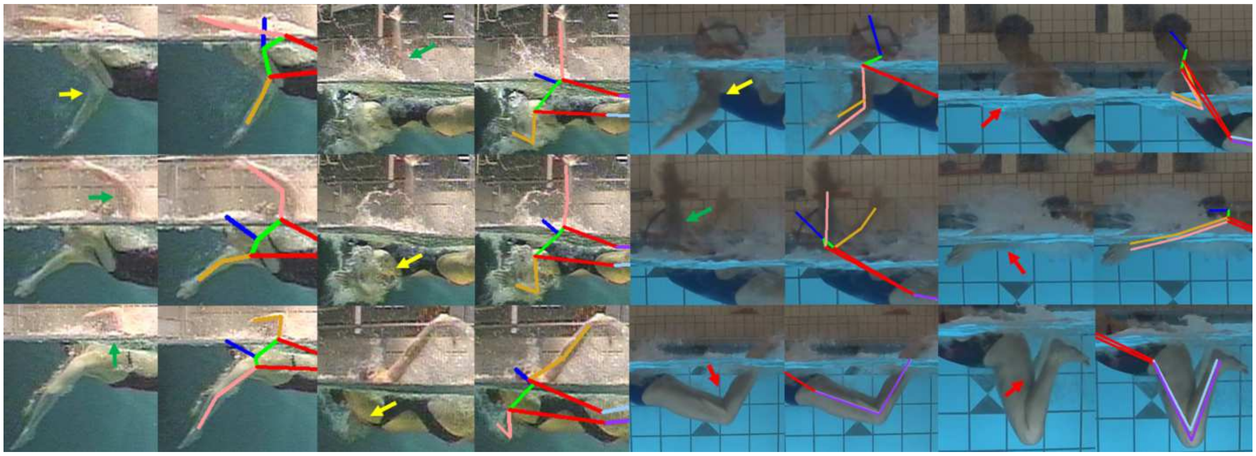

Figure 7: Example of key-poses defined by human experts. Raw images and rectified pose estimates are depicted. Arrows indicate the key-features: upper arm angle $\pi/2$ (green), upper arm angle $3/2\pi$ (yellow), angle between two parts (red).

lem of correctly identifying key-poses from a retrieval point of view. Initial pose estimates for our video database are predicted by CPMs refined on the respective sequences for the corresponding swimming styles. In the following, the raw CPM output serves as the baseline and is compared with the rectified poses from the proposed rectification pipeline. We retrieve key-poses or rather their associated key-frame numbers as follows. For each key-pose, a mathematical function of the pose feature is defined. It returns a maximal value if a key-pose is present returns smaller values otherwise. For example, the key-pose ”upper arm angle γ equals 90 degrees relative to camera plane” translates to $sin(\gamma)$. The sine equals 1 if a pose configuration depicts an upper arm angle of 90 degrees and < 1 otherwise. By flipping the sign of the function to $-sin(\gamma)$, we can detect an angle of 270 degrees. By this, we compute a timeseries for each key-pose in a sequence of predicted poses, denoted as a key-pose timeseries. A local maxima search on each key-pose timeseries then identifies possible key-frames. In order to account for double detections within temporally close frames, we apply non-maximum suppression. Within a temporal window of size ±6 frames $\approx 0.2\cdot$ cycle duration, the peak with the largest function value is kept while all other maxima in this windows are discarded. If multiple peaks within this window have the same peak value, we keep only the first occurrence. A key-pose prediction is counted as a True Positive (TP) if a ground truth annotation lies within ±6 frames. All detections outside ±6 frames that can’t be matched against a ground truth are considered false positive detections (FP). If a ground truth is not matched against a key-pose prediction, we count it as a False Negative (FN).

PCKP values for all swimming styles are depicted in Figure 5. Within the human error of ±2 frames, we report a considerable PCKP increases between +2% to +5% for all swimming styles. We additionally report the precision and recall for all retrieved key-poses at the largest allowed deviation of ±6 frames in Table 5. An increase of up to 7% for the precision means that we not only are able to improve on correctly predicted key-poses within the human error, but also retrieve far less false positive predictions for key-pose occurrences.

	free	back	fly	breast
recall CPM	0.79	0.90	0.82	0.85
precision CPM	0.78	0.77	0.87	0.84
recall refinement	0.83	0.92	0.85	0.90
precision refinement	0.85	0.82	0.89	0.88

Table 5: Precision and Recall of key-pose retrieval for CPM baseline and the proposed pipeline.

6. Conclusion

We studied the problem of key-pose retrieval on training footage of world-class swimmers in a swimming channel. The difficult visual environment leads to erroneous human pose detections and consequently to false key-pose predictions. We demonstrated that even with few training data a rectification pipeline can improve the prediction of an athlete’s joints and thereby considerably increase recall and precision of retrieved key-poses. Experiments also show that there still is room to improve on both key-point prediction and key-frame extraction. For example, the optimization problem presented in Section 4.1 is often challenged by joint swaps for spatially close joints like hips and knees. Improving on peculiarities of our approach remains future work.

Acknowledgements: We would like to thank the Institute of Applied Training Science (IAT) Leipzig and the Olympic Training Center (OSP) Hamburg for providing the video data.

References

[1] I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Sheikh. Bilinear spatiotemporal basis models. *ACM Transactions on Graphics*, 31(2):17:1–17:12, Apr. 2012. 2

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1014–1021, June 2009. 2

[3] A. Aristidou and J. Lasenby. Real-time marker prediction and CoR estimation in optical motion capture. *The Visual Computer*, 29(1):7–26, Jan 2013. 2

[4] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 468–475, May 2017. 2

[5] C. S. Burrus, J. A. Barreto, and I. W. Selesnick. Iterative reweighted least-squares design of fir filters. *IEEE Transactions on Signal Processing*, 42(11):2926–2936, Nov 1994. 5

[6] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1221–1230, 2017. 2

[7] S. R. Fanello, C. Keskin, P. Kohli, S. Izadi, J. Shotton, A. Criminisi, U. Pattacini, and T. Paek. Filter forests for learning data-dependent convolutional kernels. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 1709–1716, June 2014. 2

[8] M. Fastovets, J.-Y. Guillemaut, and A. Hilton. Athlete pose estimation from monocular tv sports footage. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013. 2

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, Sept. 2010. 4

[10] J. Hwang, S. Park, and N. Kwak. Athlete pose estimation by a global-local network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 2

[11] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2, 6

[12] U. Mall, G. R. Lal, S. Chaudhuri, and P. Chaudhuri. A deep recurrent framework for cleaning motion capture data. *CoRR*, abs/1712.03380, 2017. 2

[13] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision*, pages 33–47. Springer, 2014. 2

[14] B. Sapp and B. Taskar. MODEC: multimodal decomposable models for human pose estimation. In *CVPR*, pages 3674–3681. IEEE Computer Society, 2013. 6

[15] J. Song, L. Wang, L. Van Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[16] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1345–1352. MIT Press, 2007. 2

[17] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014. 2

[18] B. Victor, Z. He, S. Morgan, and D. Miniutti. Continuous video to simple signals for swimming stroke detection with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 2

[19] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016. 2, 3, 6

[20] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang. Learning feature pyramids for human pose estimation. In *arXiv preprint arXiv:1708.01101*, 2017. 2

[21] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[22] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, 2013. 2

[23] D. Zecha, C. Eggert, and R. Lienhart. Pose estimation for deriving kinematic parameters of competitive swimmers. In *Electronic Imaging:Computer Vision Applications in Sports*, volume 2017, pages 21–29, 01 2017. 2

[24] D. Zecha and R. Lienhart. Key-pose prediction in cyclic human motion. In *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, January 2015. 2, 6