https://doi.org/10.2352/ISSN.2470-1173.2017.16.CVAS-345

Pose Estimation for Deriving Kinematic Parameters of Competitive Swimmers

Dan Zecha, Christian Eggert, Rainer Lienhart; Multimedia Computing and Computer Vision Lab; Augsburg University, Germany

Abstract

In the field of competitive swimming a quantitative evaluation of kinematic parameters is a valuable tool for coaches but also a labor intensive task. We present a system which is able to automate the extraction of many kinematic parameters such as stroke frequency, kick rates and stroke-specific intra-cyclic parameters from video footage of an athlete.

While this task can in principle be solved by human pose estimation, the problem is exacerbated by permanently changing self-occlusion and severe noise caused by air bubbles, splashes, light reflection and light refraction. Current approaches for pose estimation are unable to provide the necessary localization precision under these conditions in order to enable accurate estimates of all desired kinematic parameters. In this paper we reduce the problem of kinematic parameter derivation to detecting key frames with a deep neural network human pose estimator. We show that we can correctly detect key frames with a precision which is on par with the human annotation performance. From the correctly located key frames, aforementioned parameters can be successfully inferred.

Motivation

In this paper, we consider a real-world computer vision application that assists a competitive athlete in assessing and improving his or her performance by taking advantage of the proposed pose estimation system. In the field of competitive swimming a quantitative evaluation of kinematic parameters is both a valuable tool for trainers as well as a labor intensive task.

The training scenario is limited to a competitive swimmer swimming in a special pool: a swimming channel (see Figure 1). The water in this pool can be artificially accelerated to constantly flow from one end of the pool to the other. The swimmer performs regular swimming motions while being filmed through a glass wall by a single camera. After the training session, the video footage is evaluated by trainers and athletes. A qualitative evaluation is usually supplemented by a quantitative analysis, where the video footage is assessed and annotated frame by frame to extract points of interest like joint positions, body part angles relative to the image plane, and other variables of interest. From theses annotations, kinematic parameters can easily be derived and used for rating the efficiency of an athlete's swimming style and giving recommendations for pose adjustments, which finally can lead to a significant performance improvement. The whole task of manually performing a quantitative analyses is extremely time consuming and therefore performed only for very few athletes nowadays.

We present a system which is able to automate the extraction of many kinematic parameters such as stroke frequency, kick rates and stroke-specific intra-cyclic parameters from video footage of an athlete. It reduces the problem of kinematic parameter derivation to detecting key frames. A key-frame depicts a key-pose, which is a pose defined by a human expert based on arbitrary features of the pose.

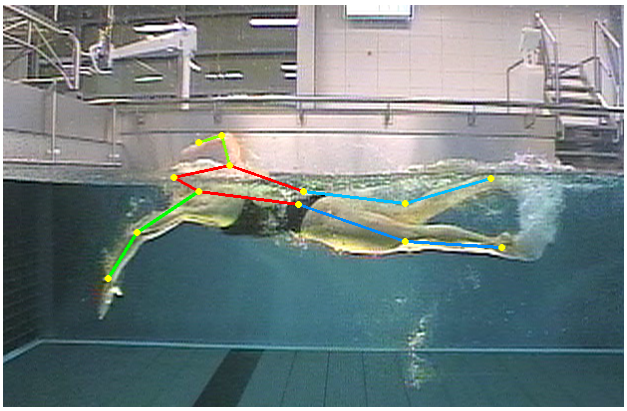

Figure 1. *A female swimmer in a swimming channel. The proposed system continuously and reliably detects poses from which kinematic parameters are extracted.*

While this task can in principle be solved by human pose estimation, the problem is exacerbated by permanently changing self-occlusion and severe noise caused by air bubbles, splashes, light reflection and light refraction. Current approaches for pose estimation are unable to provide the necessary localization precision under these conditions in order to accurately estimate all desired kinematic parameters. Our system is two-staged: Firstly, we develop a deep convolutional neural network (DCNN) architecture for efficiently computing pose configurations of swimmers in a swimming channel. Secondly, we classify the output using a neural network to distinguish between key-poses and all other poses in a swimming cycle. We show that we can correctly detect key frames with a precision which is on par with the human annotation performance. From the correctly located key frames, aforementioned parameters can be successfully inferred. Our contributions are two-fold:

- We propose a novel representation of pictorial structure models in terms of a deep neural network, taking advantage of existing infrastructure.

- Our algorithm for deriving kinematic parameters performs on par with a human annotator, allowing quantitative performance assessments not only for top-level athletes, but also for a larger spectrum of athletes.

Pictorial Structures for Human Pose Estimation. In the last decade, pictorial structure models, commonly denoted deformable part models (DPMs), have played a central role in hu-

man pose estimation and object detection in general. Some of best performing and most influential systems ([4], [29], [11]) are based on derivatives and modifications of pictorial structures.

The fundamental concept of DPMs is to represent an object, in our case a person, by a collection of parts arranged in a deformable configuration. A part is usually defined as a local area centered on the position of some joint of a human body, although parts might be added that explicitly cover locations that lie between two adjacent joints. Extensions have been proposed for mixtures of parts [29], where one part representation is split into multiple part representations that depict the part in different articulations.

With the recent success of deep convolutional neural networks (DCNNs), hand-crafted features such as HOG [5] recently have been replaced by features learned by a neural network. These features have considerably improved the performance of DPMs in the last years. Appearance features, hand-crafted or learned, are used to assign a probability or score to every possible part position in an image. A higher score usually indicates the presence of a part. The matching procedure of DPMs involves optimizing some function that usually reasons over the appearance scores of every part and a deformation term that assigns a deformation score to relative positions of neighboring parts in the model. In the context of pictorial structures, a deformation term describes an allowed derivation of one part relative to its neighboring part. If the derivation is too large, a penalty for a misplacement is introduced to the optimization problem, lowering the overall score of a part configuration.

While deformation terms have been modeled explicitly in the past, they have recently been replaced by neural networks [27], which - given a detected region of interest depicting a person - densely apply a part detection network and then use a some additional network structure to find the most probable configuration of joints implicitly in the underlying ground truths. While these networks tend to yield better scores on pose estimation benchmarks, using DPMs can still be beneficial in some scenarios. For example, DPMs make no prior assumption about the location of a person in an image, meaning that they do not depend on an initial detection of the object in question. Thus it has no problems detecting multiple instances with no prior information about their location. By combining DPMs with learned appearance features from a DCNN, we do not have to miss out on the superior appearance detection performance of neural networks.

A second problem of modeling spatial deformability with modern neural networks is that a model is usually trained for a fixed number of joints. An inaccurate bounding box prior for the initial position of a person usually confuses the detection system, yielding a false positive estimate for the part locations of the person. Additionally, pose estimation DCNNs are forced to detect the number of joints they are initially conditioned to detect, while one can simply extract a partial pose configuration from a DPM detection by resetting the detection threshold. Furthermore, as mentioned above, DPMs can be and have been extensively extended in the past with a wide variety of concepts, and are therefore frameworks which can easily be extended with application specific optimizations.

Pictorial Structures as Neural Networks. In this paper, we will focus on deriving a DCNN representation of DPMs. Expressing a pictorial structure as a neural network has has some advantages compared to classical DPM formulations.

Firstly, with recent advances in parallel computing technologies like GPU processing, DPMs can highly benefit from parallelism, improving execution time for time critical applications. With the emergence of field-programmable gate arrays (FPGAs) in the machine learning community, the block-wise formulation of DPMs as networks allows for transferring existing models to specialized hardware, enabling a real-time execution even on small devices.

Secondly, through the evolution of DPMs a lot of very well performing detectors for specific applications have been build. Our formulation allows for transferring well-tried models to new hardware without having to train them explicitly on the new hardware.

Thirdly, our formulation allows to eliminate one of the disadvantages of deformable parts, namely the restriction that classical DPMs can only learn deformation terms which approximate a Gaussian with a diagonal covariance matrix. However, depending on the application, it might be beneficial to learn arbitrary deformation matrices.

Related Work

Within the last ten years of computer vision research, part based models had a big impact in the fields of object detection [11] and (human) pose estimation ([29],[17], [9]). Based on the fundamental work Fischler and Elschlager [14], these models represent an object through multiple parts which are connected via deformation terms, allowing for matching them in a flexible configuration [13]. Different refinements have been proposed specifically for human pose estimation, e.g. by enriching a model with additional parts to compensate for the flexibility of the human body [29] or by allowing rotation of parts [1].

Improving general part appearance features has been actively researched in the context of DPMs in [20], [1] and [8]. With the resounding success of deep convolutional neural networks (DCNNs), hand-crafted features have been replaced by features learned by a neural network [26]. Chen and Yuille show in [4] that local appearance of parts learned by a DCNN can also be used to predict neighboring part locations and thereby improve the prediction performance of the whole model.

A joint training for pictorial structures and DCNNs has been picked up by Tompson et al in [24] who combine a DCNN with a Markov Random Field and successfully show that their model can successfully exploit geometric relationships between body joint locations. Yang et al [28] formulated DPMs in the context of a DCNN by introducing a message passing layer which recurrently performs inference on part detection maps. Compared to our system, their formulation only covers a basic DPM formulation excluding image dependent pairwise relations.

Most work researching the tracking of people in aquatic environments has focused on drowning detection [10], localization of athletes in swimming competitions [22], the automatic analysis of large databases of swimming videos [23] and motion analysis for video based swimming style recognition [25]. A Kalman filter framework is presented in [16] to explicitly model the kinematics of cyclic motions of humans in order to estimate the joint trajectories of backstroke swimmers. Ries et al [21] use Gaussian features for detecting a specific pose of a swimmer in a pool with the intention of initializing his/her pose.

The concept of key poses has been researched for different sport applications. Given perfectly detected poses, Dios et al [7] automatically determine key poses in cyclic human motion by performing a principal component analysis on pose estimates, followed by a supervised cluster analysis, which allows for classifying the quality of stretch motions with a high precision. Likewise, Vicente et al [6] automatically pick key poses from a query video with pose annotations obtained through motion capture in order to classify motion sequences for Taekwondo videos using a latent-dynamic conditional random field. Both approaches presume an almost perfect annotation of the human pose. Carson et al [3] includes a selection process for action specific postures by matching shape information from individual frames in order to recognize specific tennis strokes in game footage. In a previous publication [31] which builds on the findings of [30], we detected key frames in the same scenario as presented in this paper using armlet and leglet [15] classifiers for predicting cyclic swimming motion. Compared to [7] and [6], no external pose prior was used for inferring key poses. Instead, we showed that key poses can be detected from noisy and vague pose estimates.

Human Pose Model

We will introduce a deformable part model for human pose estimation in the following. The specified description will serve as the underlying model for our DCNN formulation in the next Section. Compared to the standard formulation of a DPM which usually only includes a term for part appearance and one for deformation scoring, we include two additional terms that model image dependent pairwise relations (IDPR terms, [4]). These visual terms model the basic concept that the visual context of a part can be used to make a prediction for the location of neighboring parts and therefore improve part location estimates in the framework. For example, if we look at a cropped part of a human wrist, it is quite simple to roughly extrapolate the position of the elbow and thereby restrict the search radius. We will show that an extension like IDPR terms can also easily be expressed as a deep convolutional network.

Model Description. We represent the human pose by means of a tree graph $G = (V,E)$, where vertices V specify the positions of body parts and edges $E \subseteq V \times V$ indicate which vertices are spatially related. The pixel locations of the parts are denoted $\mathbf{l} = \{\mathbf{l_i}\}_{i=1}^{|V|} = \{(x_i, y_i)\}_{i=1}^{|V|}$, where $i = \{1,...,|V|\}$ is the i-th part. We denote the part $i = 1$ the root part of G. Each vertex j relative to the root has a depth d_j which is the number of edges between it and the root node. While the edges in our model are not directed, we will say that i is a child of j and j is a parent of i iff $(i, j) \in E$ and $d_j < d_i$. The children $\mathscr{C}(i)$ of vertex i are the vertices connected to i with a depth of $d_i + 1$. Every vertex i other than the root has exactly one parent $\mathscr{P}(i)$ with depth $d_i - 1$. The neighboring vertices of vertex i are denoted $\mathscr{N}(i) = \mathscr{P}(i) \cup \mathscr{C}(i)$. Furthermore, a subtree $\mathscr{S}_i$ of G contains all vertices j that are directly or indirectly connected to a vertex i while $d_j > d_i$ as well as vertice i itself.

Part Appearance Detectors. Parts usually cover the region around joints of the human body, although a model might explicitly include parts that lie between two joints. In the context of human pose estimation, the term spatial relationship translates to the offset vector between two neighboring parts.

As the depiction of the human pose in an image can be highly deformable, we allow for more than just one offset between two parts and model a set of spatial relationships $\mathbf{t}_{i,j} = \{k\}_{k=1}^{K}$ and $\mathbf{r}_{i,j} = \{\mathbf{r}_{i,j}^{t_{i,j}}\}$, where the offsets between two neighboring parts i and j are clustered into K clusters with identifiers $\mathbf{t}_{i,j}$ and cluster centers $\mathbf{r}_{i,j}^{t_{i,j}}$.

Given a dataset of swimmer images with a ground truth joint annotations $\mathbf{l}_i$, we extract image patches $I(\mathbf{l}_i)$ centered around each joint i. A class label out of a set of labels $\mathscr{L}$ is assigned to each patch $I(\mathbf{l}_i)$, consisting of its joint class $c = i$ and identifiers $\mathbf{t}_{i,\mathscr{N}(i)}$ obtained from clustering each offset between a joint i and its neighbor $j \in \mathscr{N}(i)$ in the training data. Intuitively, all part patches sharing a part label depict the same joint while all the neighboring joints have the same spatial configuration. We train an AlexNet [18] with soft-max loss on all extracted patches and their respective class labels.

This leads to a rather large set of $|\mathscr{L}| = \sum_{i \in V} \prod_{j \in \mathscr{N}(i)} |\mathbf{t}_{i,j}| + 1$ specialized classes for detecting very specific configurations of joints and one class for background detection. After training, we modify the last two fully connected layers to behave like convolutional layers. This allows for feeding an image of arbitrary size into the network instead of single image patches, while retrieving a score map covering each patch sized input window in the original image. From the rather specific output of our part detection network, we retrieve general joint detectors for joint i by summing over all network outputs where the class label incorporated the joint class $c = i$. We denote the appearance detector for joint i at location $\mathbf{l}_i$ with $\Phi(c = i, \mathbf{l}_i)$.

Image Dependent Pairwise Relation Terms. Our model incorporates image dependent pairwise relations (IDPR) terms, originally introduced in [4]. The key idea behind idpr-terms is that an image region around a joint often gives strong evidence of the positions of neighboring joints and can therefore be used to improve said neighbor joint locations. We define an idpr-term $\Gamma(c = i, \mathbf{t}_{i,j} = k, \mathbf{l_i})$ by summing over all class outputs for joint i where the class label includes a specific cluster id $\mathbf{t}_{i,j} = k$ for the edge $(i, j) \in E$.

Deformation Terms. The model description is completed by a second stage, connecting different visual estimates form the detectors and providing a holistic pose estimate. The classical deformation term assesses the fit of two detections for neighboring parts and allows for some flexibility in the relative position between both. It is defined by

$$\Psi(\mathbf{l_i}, \mathbf{l_j}, t_{j,i}, \mathbf{w}_{j,i}^{t_{j,i}}) = \left\langle \mathbf{w}_{j,i}^{t_{j,i}}, \delta\left(\mathbf{l_i} - \left(\mathbf{l_j} + \mathbf{r}_{j,i}^{t_{j,i}}\right)\right)\right\rangle \quad (1)$$

where $\delta(\Delta\mathbf{l} = (\Delta x, \Delta y)) = [\Delta x, \Delta x^2, \Delta y, \Delta y^2]^T$ is a deformation feature and $\mathbf{w}_{j,i}^{t_{ij}}$ are learned deformation weights penalizing larger magnitudes of $\Delta\mathbf{l}$. The notion behind deformation terms is to allow for a part j to deviate slightly from its ideal placement $\mathbf{r}_{j,i}^{(k)}$ relative to part i. While small deviations from the ideal offset of two parts lower the score only by a small margin a part that is located further away from the ideal offset also leads to a larger score decay. The steepness of this decay is defined by the deformation weights $\mathbf{w}_{j,i}^{t_{j,i}}$, where smaller deformation weights allow for a larger deviation from the optimal displacement.

Model Energy. The goal of articulated human pose estimation is to find a placement for all joints so that the model energy is maximized. The goodness of a placement $\mathbf{l} = \{l_1, ..., l_{|V|}\}$ of all

model parts c for an image I can be evaluated by the following energy function:

$$\begin{aligned} F(\mathbf{l},\mathbf{t},I;\Theta,\mathbf{w}) &= \sum_{i\in V} w_i\Phi(c=i,\mathbf{l}_i) + \sum_{(i,j)\in E} \Psi(\mathbf{l_i},\mathbf{l_j},t_{j,i},\mathbf{w^{t_{j,i}}_{j,i}}) \\ &+ \sum_{(i,j)\in E} w_{i,j}\Gamma(c=i,t_{i,j},\mathbf{l_i}) \\ &+ \sum_{(i,j)\in E} w_{j,i}\Gamma(c=j,t_{j,i},\mathbf{l_j}) \end{aligned} \tag{2}$$

where Θ are the parameters of the DCNN and $\mathbf{w}$ are learned weights. The energy function is composed of three semantically distinguishable terms. The first sum in equation 2 evaluates the appearance of a specific placement of the parts. The second sum rates the goodness of placements of neighboring parts. Both terms describe the traditional energy formulation for matching a traditional deformable part model. The last two sums assess the local belief that a neighboring part is placed at a certain location based on image evidence.

Weight Parameter Training. The energy formulation in equation 2 includes weight terms for appearance and image dependent pairwise relations as well as weights influencing the allowed deformation between two neighboring parts. We learn all weight parameters by defining $\phi(\mathbf{I^n},\mathbf{l^n},\mathbf{t^n})$ as sparse feature vectors representing the concatenation of image dependent terms, idpr terms, deformation features and a constant 1 representing a bias term for a positive pose example in image $\mathbf{I^n}$ with annotated part locations $\mathbf{l^n}$ and derived type labels $\mathbf{t^n}$. Note that all dimension in this feature that do not correspond to type labels and deformation terms included in $\mathbf{t^n}$ are set to zero. Negative examples are obtained by hard-negative mining detections on images depicting no athletes.

We learn all weights $\mathbf{w}$ by training a support vector classifier by optimizing

$$\min_{\mathbf{w}} \frac{1}{2}\langle \mathbf{w}^T\mathbf{w}\rangle + C\sum_{i=1}^{n}\xi_i \tag{3}$$

subject to

$$\begin{aligned} y_i(\mathbf{w}^T\phi(\mathbf{I^n},\mathbf{l^n},\mathbf{t^n})) &\geq 1-\xi_i \\ \xi_i &\geq 0, i=1,...,n \end{aligned} \tag{4}$$

Here, $y_i \in \{1,-1\}$ denotes the label of example i, where $y_i = 1$ for positive examples and $y_i = -1$ for a negative feature. Note that the bias term for this optimization problem is omitted as we already included it in the feature vector.

Inference and Backtracking. Inference aims at maximizing equation 2, i.e., we wish to find the best placements $\mathbf{l}$ of the parts and their pairwise relations $\mathbf{t}$, respectively:

$$\mathbf{l}^*,\mathbf{t}^* = \text{argmax}_{\mathbf{l},\mathbf{t}} F(\mathbf{l},\mathbf{t},I;\Theta,\mathbf{w}) \tag{5}$$

Optimizing this joint distribution function over a tree-shaped graph is usually done by means of dynamic programming using the max-sum algorithm, which allows for recursively splitting the problem into subproblems. Let $S_i(\mathbf{l_i})$ denote the score of a model for a subtree $\mathscr{S}_i$ of G. We can recursively compute the score of each subtree of G by computing $m_{i,n}$ for each child n of node i through

$$\begin{aligned} m_{i,n} = \max_{l_n,t_{n,i}} (&\Psi(\mathbf{l_i},\mathbf{l_n},t_{n,i},\mathbf{w^{t_{n,i}}_{n,i}}) + w_{i,j}\Gamma(c=i,t_{i,n},\mathbf{l_i}) \\ &+ w_{j,i}\Gamma(c=n,t_{n,i},\mathbf{l_n}) + S_n(\mathbf{l_n})). \end{aligned} \tag{6}$$

Note that the deformation term in 6 can be efficiently implemented by means of a generalized distance transform [12]. All partial subtree scores are summed up for the parent node by

$$S_i(\mathbf{l_i}) = w_i\Phi(c=i,\mathbf{l_i}) + \sum_{n\in\mathscr{C}(i)} m_{i,n}, \tag{7}$$

yielding a partial score for subtree $\mathscr{S}_i$.

Given the set of all partial scores, we can infer the joints locations of a high scoring detection through backtracking. In order to perform backtracking efficiently, it is beneficial to trace certain variables of the optimization procedure. For each location $\mathbf{l_i}$ of part i, the generalized distance transform gives us the location of the highest score from nearby locations.

We save these locations in an array L_i of the same size as S_i. If a score at a location $\mathbf{l_i}$ was produced by a nearby high score, we can look up the location of this high score through $L_i(\mathbf{l_i})$. Additionally, as we like to track more than one person in an image, we also have different variables $t_{i,j}$ for different locations. We save these in an array T_i of the same size as S_i. For a certain position $\mathbf{l_i}$ of a parent part i, the respective value for $T_i(\mathbf{l_i}) = t_{i,j}$ tells us the offset $\mathbf{r}_{i,j}^{t_{i,j}}$ of the child part j relative to i.

Once all part solutions S_i are computed, we can find the optimal solution $\mathbf{l_i}$ through backtracking. A global maximum of the energy function is found by picking the root location $\mathbf{l_1} = \mathbf{l_1^*} = \text{argmax}_{\mathbf{l}}(S_1(\mathbf{l}) > \tau)$ given a detection threshold τ. The location of all other parts are traced back in order of increasing depth in our pose tree by a two step process. Firstly, starting from the position of the parent $\mathbf{l_j}$, we invert the translation performed during inference by the deformation term equation 1 by computing

$$\mathbf{l_i^*} = \mathbf{l_j} + \left(-\mathbf{r}_{j,i}^{T_i(t_{j,i})}\right) = \mathbf{l_j} + \mathbf{r}_{i,j}^{T_i(t_{i,j})}. \tag{8}$$

Note that we perform a lookup of the child offset in T_i. This allows for tracking multiple persons in one image for different local root part maxims. Secondly, we need to invert the score spreading performed by the generalized distance transform. This is simply done by a lookup:

$$\mathbf{l_i} = L_i(\mathbf{l_i^*}) \tag{9}$$

Note that our system has no prior information about the position of a person, but instead is able to find all high scoring pose configurations in one image. We perform non-maximum suppression on bounding rectangles of part locations on order to filter multiple detections of a pose configuration for one person.

Pictorial Structures as a Neural Network

In this section, we will show that the model described in the previous can be expressed as a computational graph and therefore implemented as a deep neural network. Firstly, we will focus on a single pairwise optimization step of a subtree as described by Equations 6 and 7 and explicitly characterize all used calculations.

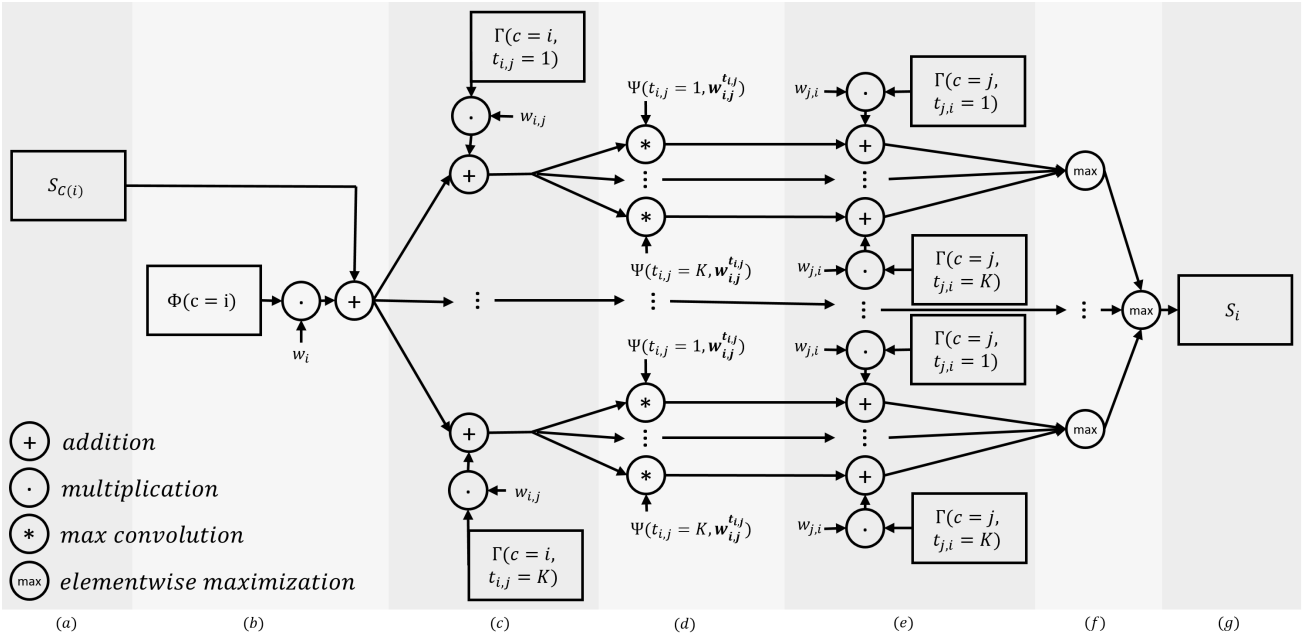

Figure 2. *One step of the dynamic program. Partial solutions of all children in (a) are added to the weighted appearance of the current vertex in (b). The sum is furthermore added to c) weighted image dependent pairwise relations for the child node in (c). Different distance transformations are performed in (d) for different pairwise relations. In (e), the transformed results are added to parent idpr term maps. The final partial optimal solution for S_i is obtained in (g) by taking the element-wise maximum over all partial results in (f). A high scoring joint configuration can be obtained by backtracking the optimal path in this trellis.*

Then, we will present a network structure for detecting an athlete in an image.

Pairwise Optimization in a DCNN. We propose the computational graph depicted in Figure 2 to solve Equations 6 and 7. In this figure, rectangles correspond to probability maps f appearance and idpr terms, circles denote an operation on said maps and arrows visualize the data flow of maps and scalars through the optimization procedure. The whole problem can be broken down into four basic operations: element-wise addition between two maps, multiplication of a map with a scalar, a max-convolution between a deformation kernel and a probability map and an element-wise maximization for different probability maps of the same size. All these operation can simply be expressed as a DCNN. For clarity, we assume that the edge between a part i and its parent j has been clustered K times. As a result, there are K different idpr maps describing the belief for the position of parent j seen from part i and another K idpr maps describing the belief for the position of child i seen from parent part j. Also, K clusters implicitly define K different deformation terms together with their offsets $\mathbf{r}_{j,i}$ between part j and i.

The computational graph depicted in Figure 2 performs pairwise optimization for a vertex i, yielding a solution for the subtree $\mathscr{S}_i$ in G as follows. The appearance of part i is weighted with its respective weight in (a) and added to all subtree scores obtained from children of part i in (b). If i itself is a leaf vertex in G, then this sum is omitted. The sum of appearance and partial solutions is then added to the weighted image dependent terms (c) of vertex i with respect to its parent node j. This step is executed K times for all idpr maps. For each sum of appearance, partial subtree solutions and idpr terms, a max-convolution is performed in (d). The number of max-convolutions to be computed corresponds to K, yielding overall K^2 differently transformed terms in stage (d). Each set of differently transformed probability maps obtained from just one child-to-parent idpr map is now added to all parent-to-child idpr terms in (e). Note that stages (c) to (e) simply compute the Cartesian combinations of all idpr terms between two parts. Also, the last element-wise sum in step (e) does not alter the number of K^2 maps from stage (d) as every map in that step get added to exactly one of the K idpr terms. In step (f), the element wise maximum of all intermediate results are computed, finally yielding the subtree-score $\mathscr{S}_i$.

A DCNN Layer for Max-Convolutions. While the operations of addition, multiplication and element-wise maximization are straight forward to implement, the formulation of a max-convolution has to be defined more precisely. Felzenszwalb et al. proposed in [12] a general distance transform algorithm for computing the deformation cost efficiently on 2D grids. While their algorithm in its proposed form is not suited for a direct implementation on a GPU, we can find an equivalent formulation that can easily be implemented for parallelization on a GPU.

Let F be a regular grid and $f : F \to \mathbb{R}$ be a function defined on that grid. The generalized distance transform at a point $\mathbf{p}$ is defined over neighboring points $\mathbf{q}$ given a convolution mask $h(.)$ as

$$D_f(p) = (f \otimes h)(p) = \max_{\mathbf{q}}(f(\mathbf{q}) - h(\mathbf{p} - \mathbf{q})) \tag{10}$$

Note that this formulation somewhat resembles a discrete convolution where the sum is replaced by the max operator and the product is replaced by a summation. In contrast to the standard convolution a max convolution is highly non-linear. Classical

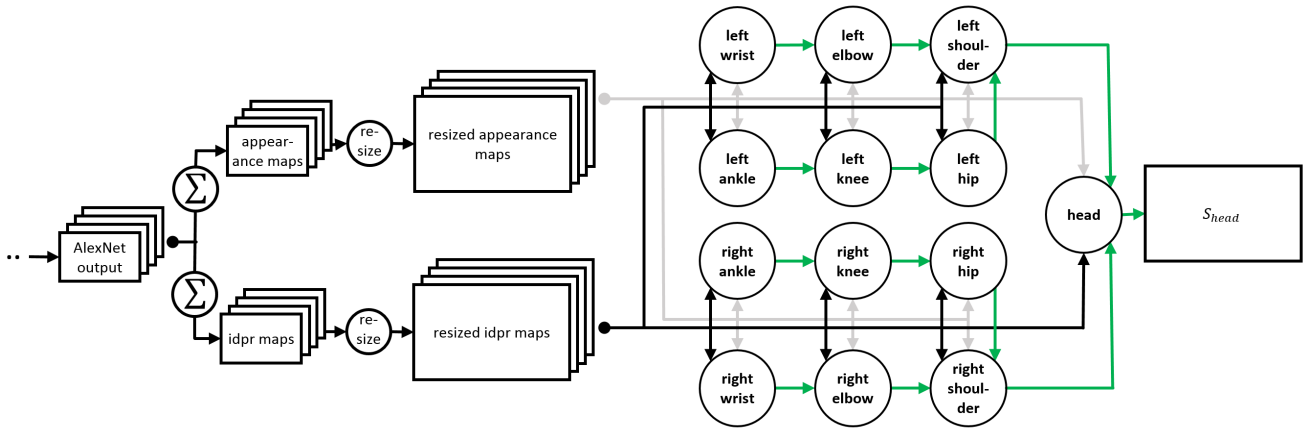

Figure 3. *A DCNN structure for pose estimation. The output maps of the AlexNet are summed up to form a representation for appearance and image dependent pairwise relation terms. After resizing all terms to the original image size, both are passed to a network (black and grey arrows) of part-to-part layers (circles, see also Figure 2) representing the structure of the human pose graph G. The part to part layers compute partial solutions for each joint ($\mathscr{S}_i$, green arrows).*

graph-based pose estimation approaches often assume that the deformation costs are modeled by quadratic terms, i.e., $h(\mathbf{p}-\mathbf{q}) = \langle \mathbf{w}, (\mathbf{p}-\mathbf{q}) \odot (\mathbf{p}-\mathbf{q}) \rangle$, where $\odot$ denotes the Hadamard product. The parameter $\mathbf{w}$ controls the width of a 2D parabola centered at $\mathbf{p}$, penalizing scores $f(\mathbf{q})$ while taking into account the distance between $\mathbf{p}$ and $\mathbf{q}$. The greater the distance between $\mathbf{p}$ and $\mathbf{q}$, the greater the penalty on $f(\mathbf{q})$. We can interpret Equation 10 as a filtering operation, where $f(.)$ is a function defined on a 2D-grid and $h(\mathbf{p}-\mathbf{q})$ is a max-convolution filter. The filter kernel has a side length of $2s+1$ and is initialized to the function

$$h(\mathbf{x}) = \left\langle \mathbf{w}, \left(\mathbf{x} - (s,s)^T\right) \odot \left(\mathbf{x} - (s,s)^T\right) \right\rangle, \tag{11}$$

which is a weighted parabola rooted at the center of the kernel. Note that the weights $\mathbf{w}$ are initialized with the weights learned from our original SVM formulation. Using this kernel, the distance transform at any point $\mathbf{p}$ can then be obtained by placing the filter at position $\mathbf{p}$, performing an element-wise subtraction from the function $f(.)$ and maximizing over all resulting values.

DCNN inference. The network depicted in Figure 2 presents only one element in the matching pipeline of a DPM, namely the optimization step for finding the optimal placement of one part relative to its parent part. For convenience, we will denote the whole step a part-to-part-layer. This very high level of abstraction allows for describing the pose estimation optimization procedure as a neural network. This is depicted in Figure 3 discussed in the following.

As described in the previous section, we train an AlexNet to classify all parts in their different configurations to neighboring parts. We then reconfigure the last two fully connected layers in the detection network such that the connections are interpreted as convolution kernels. This modification allows for inherently efficient processing of a complete image in one pass through the network by sharing network weights for different input windows. For each part, we obtain a set ob probability maps representing the belief that the part is present in different configurations of neighboring parts.

From this rather large collection of specific belief maps, we extract two stacks of probability maps, one for the appearance of the parts and one for image dependent terms. Both collections are obtained by summing over distinct sets of output maps which we memorized during clustering. In the context of deep neural networks, this element-wise sum can easily be expressed by a 1×1 convolution, one for each appearance/idpr map. We found that reasoning over the network output, which is much smaller than the input image due to the pooing operations in the AlexNet is disadvantageous because we loose valuable spatial resolution. Therefore, we resize all appearance and idpr maps to the size of the original image using a deconvolution layer that implements a bilinear deconvolution kernel.

Both resized stacks then are fed into a tree-shaped network of part-to-part layers (black and grey arrows in Figure 3, each circle corresponds to one part-to-part layer), which are connected in a tree-shaped graph so that the partial solution $\mathscr{S}_{child}$ serves as an input for optimizing the next parent node (green arrows in Figure 3). Note that each of the part-to-part elements represents exactly the tree graph G.

Training Pose Classifiers for Kinematic Parameter Extraction

Given the pose estimates of our DCNN, we finally aim at predicting key-frames in the motion sequence of an athlete. We implement the following training procedure. Given a database of training videos of athletes swimming in a swimming channel with additional expert annotations of key-frames, we use the DCNN described in previous section and extract the top scoring pose estimate in each video frame.

We transform the absolute joint locations to relative coordinates by passing all pose estimates through a centering layer, which centers the joint locations around the x-coordinate of the head and the y-coordinate of the water surface. For each key pose, the centered pose configuration is passed into a small, pose specific fully connected neural network with one hidden layer of size 128 with rectified linear activations and one output neuron. While keeping all parameters of the part detection and the DPM network

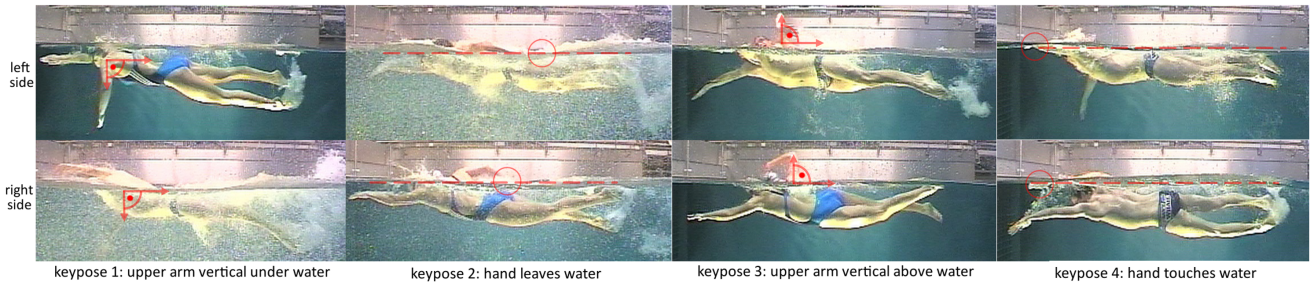

Figure 4. *Key poses for freestyle swimmers. Due to the anti-symmetric nature of freestyle swimmers, a key pose occurs two times within one cycle, once on the right body side and once on the left. Note: The image for key pose 4 (left side) actually also depicts a key pose 1 for the right arm. In our system, this is not a problem as we always detect a key pose independently from all all key poses.*

fixed, we train these small networks using a cross-entropy loss.

Naturally, the number of occurrences of a key pose is limited to one frame per complete swimming cycle, while all other frames in the cycle do not depict a key-pose and are therefore considered negative training examples. To counteract this rather imbalanced set of positive versus negative training examples, we increase the number of positive training examples by labeling the poses in the adjacent frames of a key pose frame also as key poses. We decrease the number of negative non-key-poses by randomly sampling from the set of all negatives so that we have approximately 5 times more negatives than positives.

Given a test video, we apply the whole pose detection pipeline to each frame and build pose features for all frames. The features are then fed into all key-pose networks, yielding one key pose time series for each key pose. We aggregate all predictions into a time-series, which is filtered with a low pass filter to account for outliers. Finally, local peaks in the time series correspond to key-frames in the video.

Experiments

We evaluate the performance of our system on a set of 30 swimmer videos depicting the athletes from a side view through a glass wall. Each video frame is fed into the network at full resolution of 720×576 at 50 frames per second. The videos cover different freestyle swimmers (ages 15-25, 8 female, 6 male, different body size and posture) swimming in a swimming channel at different velocities between $1ms^{-1}$ and $1.75ms^{-1}$ with a maximal increasing flow velocity of $0.3ms^{-1}$ in one video. A human expert annotated all frames that depict one of four key-poses which are visualized in Figure 4.

We train the part detection network on a separate training dataset of 1200 images. Each edge in the model is clustered with k-means setting k=11. We crop patches from overall 13 joints (head, shoulders, hips, knees, ankles, elbows, wrists) with a patch-size of 100×100. The training set is extended by adding random variations where we rescale images with a factor of $[0.8, 1.2]$ and random rotations within $+/-15$ degrees. We add negative patches from images showing empty swimming channels with different water flow velocities to adapt to noise due to water bubbles. Overall, we collect 44.500 training patches, which are resized to a net input size of 256×256. We train the network for 40.000 eiterations with an initial learning rate of 0.001, which is reduced by a factor of 0.1 each 10.000 iterations. The weight parameters of the deformable part model are optimized on the training images using three bootstrapping rounds, holding the weights of the DCNN fixed. We hold all model parameters fixed and apply it to all training videos in a 30-fold leave-one-out cross-validation, where we train a neural network model on 29 videos and evaluate the performance on the remaining video.

Key pose	no. 1	no. 2	no. 3	no. 4	mean
[31]@0.02	0.76	0.62	0.59	0.52	0.62
ours@0.02	**0.81**	**0.81**	**0.79**	**0.62**	**0.75**
[31]@0.03	0.89	0.71	0.72	0.66	0.75
ours@0.03	**0.93**	**0.95**	**0.93**	**0.88**	**0.92**

Table 1. Percentage of correct key-frames at deviation thresholds 0.02 and 0.03 for all poses compared to the system presented in [31].

Key-Pose performance measures. As an error measure, we use a *percentage of correct key-frames (PCKF)* measure, which compares the stokelength-normalized deviation of a predicted key-frame to its ground truth annotation on the x-axis versus the percentage of correctly detected key-frames on the y-axis and is conceptually similar to the *percentage of correct joints (PCJ)* measure widely used in the field of human pose estimation. We will evaluate the PCKF measure at a threshold deviation of 3%, which we empirically found to be the error threshold at which human experts usually operate. A deviation of 3% reflects a discrepancy of $+/-2$ frames.

Joint localization. To get a feeling for the performance of the joint detection capabilities of the first part of our network, we evaluate the joint localization performance of our part detection network in Figure 6. The percentage of correct joints metric was used, where the distance between a ground truth joint annotation and a detected joint instance is normalized with the upper body size, defined by the distance between left shoulder and right hip, to take different body sizes into account.

Figure 6 (top right) shows that we achieve a pose estimation performance mean of 93.3% at the common threshold of 0.2. For the same threshold, Chen and Yuille [4] report a PDJ value for elbows and wrists, which in their implementation equals 94.9% and 92.0% respectively. Our implementation achieves scores of 0.943% and 0.911% respectively, which is slightly below their best scoring system trained for poses in non-aquatic environments. We assume that the difference between detection scores

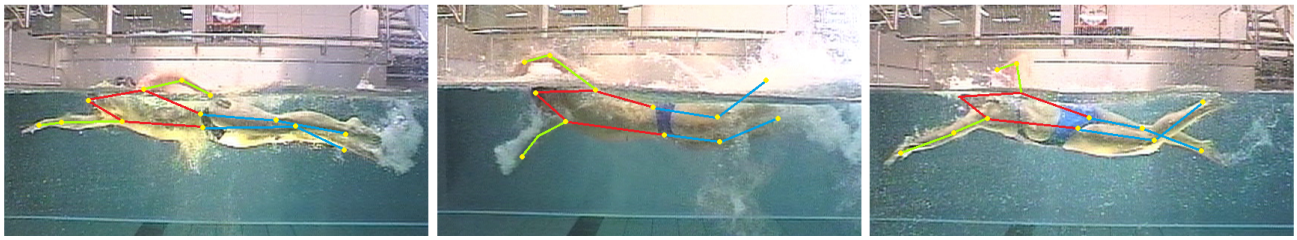

Figure 5. *Three different swimmer poses estimated by our system. The left and the right image are also classified as key poses 2 and 3, respectively.*

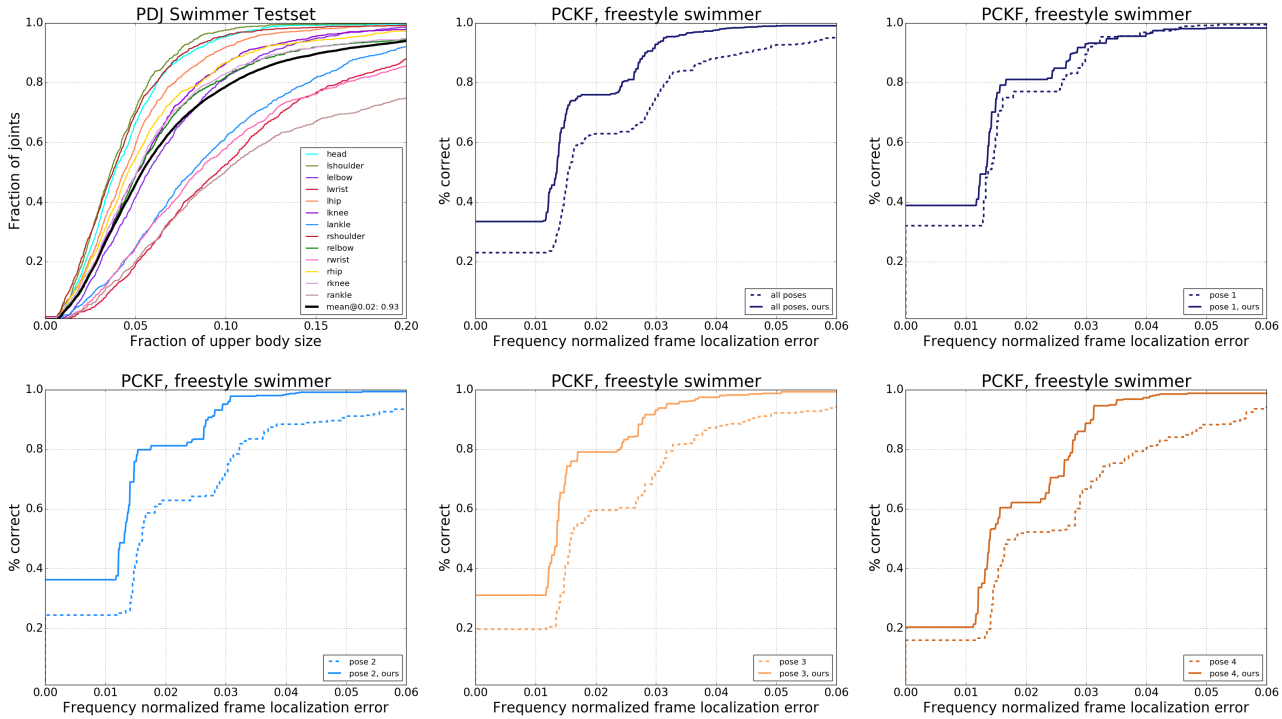

Figure 6. *Top-left: Percentage of correct joints for our part detection network. Top-middle: Mean percentage of correct key-poses, for all four keyposes. Rest: Percentage of correct key-poses for each keypose.*

are causes by the large amount of self occlusion of a swimmer's extremities. Figure 5 depicts three swimmer poses estimated by our system.

Key-Frame localization error. We evaluate the PCKF of our system against the best performance reported in [31] in Figure 6 (middle and right) and summarize the results in Table 1. On average, our system outperforms [31] by 17%. Note that [31] uses two cameras to predict a key pose while our approach only uses one camera perspective.

For a deviation threshold of 0.03 we can find 93% of all key-frames correctly while our worst detection rate is only 88%. We observe a significant key-frame identification improvement for all single key poses over all thresholds, although we find that the improvements for smaller thresholds are likewise smaller. If a key pose was difficult to detect in [31], we also achieve smaller score improvements.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009. Best Paper Award Honorable Mention by IGD.

[2] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision (ICCV)*, 2009.

[3] S. Carlsson and J. Sullivan. Action recognition by shape matching to key frames. In *IEEE Computer Society Workshop on Models versus Exemplars in Computer Vision*, 2001.

[4] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In C. Schmid, S. Soatto, and C. Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*,

volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

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

[7] P. F. d. Dios, Q. Meng, and P. W. H. Chung. A machine learning method for identification of key body poses in cyclic physical exercises. In *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics*, SMC '13, pages 1605–1610, Washington, DC, USA, 2013. IEEE Computer Society.

[8] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *British Machine Vision Conference*, September 2009.

[9] M. Eichner and V. Ferrari. Appearance sharing for collective human pose estimation. In *Computer Vision - ACCV 2012 - 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I*, pages 138–151, 2012.

[10] H.-L. Eng, K.-A. Toh, W.-Y. Yau, and J. Wang. Dews: A live visual surveillance system for early drowning detection at pool. *IEEE Trans. Circuits Syst. Video Techn.*, 18(2):196–210, 2008.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[12] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004.

[13] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vision*, 61(1):55–79, Jan. 2005.

[14] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.*, 22(1):67–92, Jan. 1973.

[15] G. Gkioxari, P. Arbelaez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:3342–3349, 2013.

[16] T. Greif and R. Lienhart. A kinematic model for Bayesian tracking of cyclic human motion. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7543 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Jan. 2010.

[17] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, pages 12.1–12.11. BMVA Press, 2010. doi:10.5244/C.24.12.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[19] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[20] D. Ramanan. Learning to parse images of articulated bodies. In P. B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1129–1136. MIT Press, 2007.

[21] C. X. Ries and R. Lienhart. Automatic pose initialization of swimmers in videos. volume 7543, page 75430J. SPIE, 2010.

[22] L. Sha, P. Lucey, S. Morgan, D. Pease, and S. Sridharan. Swimmer localization from a moving camera. In *DICTA*, pages 1–8. IEEE, 2013.

[23] L. Sha, P. Lucey, S. Sridharan, S. Morgan, and D. Pease. Understanding and analyzing a large collection of archived swimming videos. In *IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, March 24-26, 2014*, pages 674–681, 2014.

[24] J. J. Tompson, A. Jain, Y. Lecun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1799–1807. Curran Associates, Inc., 2014.

[25] X. Tong, L.-Y. Duan, C. Xu, Q. Tian, and H. Lu. Local motion analysis and its application in video based swimming style recognition. In *ICPR (2)*, pages 1258–1261. IEEE Computer Society, 2006.

[26] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013.

[27] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016.

[28] 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 *CVPR*, 2016.

[29] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1385–1392. IEEE, 2011.

[30] D. Zecha, T. Greif, and R. Lienhart. Swimmer detection and pose estimation for continuous stroke-rate determination. In *Proc. SPIE*, volume 8304, pages 830410–830410–13, 2012.

[31] D. Zecha and R. Lienhart. Key-pose prediction in cyclic human motion. In *Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision*, WACV '15, pages 86–93, Washington, DC, USA, 2015. IEEE Computer Society.

Author Biography

Dan Zecha is a PhD student at the Multimedia Computing and Computer Vision Lab at Augsburg University, Germany, where he also received his BSc and MSc in computer science. His research interests lie in applications of machine learning and computer vision for fully automatic human pose analysis. More specifically, his work examines how kinematic and dynamic parameters of competitive athletes can be extracted continuously from video footage and be used for improving training procedures.