CSIT Conference 2023, Yerevan, Armenia, September 25 - 30

On Mobile Pose Estimation Design and Implementation

Minas Aslanyan
Institute for Informatics and
Automation Problems
Yerevan, Armenia
e-mail: likwifi@gmail.com

***Abstract*—Human Pose Estimation (PE, tracking body pose on-the-go) is a computer vision-based technology that identifies and controls specific points on the human body. These points represent our joints and special points over the body determining the sizes, distances, angle of flexion, and type of the motion. Knowing this in a specific exercise is the basis of work for rehabilitation and physiotherapy, fitness and self-coaching, augmented reality, animation and gaming, robot management, surveillance and human activity analysis. Implementing such capabilities may use special suits or sensor arrays to achieve the best result, but massive use of PE is related to devices that many users own – namely smartphones, smartwatches, and earbuds.**

The body pose estimation system starts with capturing the initial data. In dealing with motion detection, it is necessary to analyze a sequence of images rather than a still photo. Different software modules are responsible for tracking 2D key points, creating a body representation, and converting it into a 3D space.

Human pose estimation is a machine-learning technology, which means that we need data to train it. Since human pose estimation completes quite difficult tasks of detecting and recognizing multiple objects on the screen, they use neural networks as an engine of it. Human pose estimation projects can be quite complex and require expertise in a number of domains. They need compact tools of generative NN and transformers, the use of special Dynamic Time Warping, movement coding languages, recommenders and decision making.

***Keywords*—Pose estimation, deep learning, dynamic time warping.**

I. INTRODUCTION

Static and dynamic objects, together with the natural physical environment, constitute the infrastructure for the coexistence of the entire set of biological and mechanical systems [1]. The objects that move are mainly biological beings and systems controlled by them. In the process of evolution, they are equipped with systems of observation, coordination and safe movement in the environment. We will be interested in the concept of pose estimation, with a global meaning of determining the absolute or relative position of an object or its elements in the environment [2]. This ability is inherent in biological beings and much research and work have been conducted to adopt a similar technique for artificial mechanical devices and systems. One of the major areas of research belongs to the autonomous cars industry with the goal of creating self-driving cars. In principle, we are talking about automating the visual sense of the situation through a system of video surveillance and possibly, sound, and traffic control on this basis. Other tasks from our field of interest are the task of drone self-guidance on a map without GPS, the task of detecting and tracking objects observed by radar, and of course, our main task of controlling the human body as a compound of complex critical functions [1,3-5]. Specifically, this task is crucial in controlling robots for the work in natural and man-made disasters, work in outer space. The other major use cases include the tasks of organizing training and evaluating functions of specific exercises, which are critical at sports competitions, dance competitions, in the training of soldiers, rescuers, etc. [6,7].

The two main components of video interpretation and automation systems are surveillance systems and data analysis systems. Usually, observations are made by one or many cameras placed at varying angles. The observed scene can also be specially prepared or an arbitrary environment can be used [8,9]. According to the captured data, its analysis systems also differ, some start from standard image analyzes, which can use elements of image annotation, others can be trained without them, as conventional neural networks do [10,11]. Success also depends on the specification of the task and the structure of the recorded data.

In this paper, we will not consider arbitrarily, but only focus on those moving objects that have a skeletal structure with hinges and a motor mechanism [12]. They are people and animals, as well as mechanical robots and devices. Each class that we study must be described with the structure, components, their properties and validation, etc. Input data that we consider for this task, can be direct raw images or their sequences, [2, 8, 9, 13, 14] from one or several positions. Annotations [15] will also be used and will introduce additional information, developed both by man and by technical means. In the context of the previous annotated example, the utilization of machine learning systems can be explored alongside direct geometric or kinetic analysis of motion [16-18]. Typically, the optimal outcomes are achieved through the combined application of geometric analysis and artificial intelligence. The pivotal constituents of the scientific and technical systems under consideration encompass mathematical, computational, and instrumental components. These components consist of observation subsystems, interfaces, analysis modules, and decision-making processes.

https://doi.org/10.51408/csit2023_26

The precision of detection and interpretation is heavily reliant on the quality of the technical equipment in use. Moreover, the extent of the system's capabilities is also contingent on this equipment.

To elaborate the configuration of the system can vary widely [14,19]. It can range from an elaborate setup in a dedicated studio replete with multiple cameras and specialized recording devices, to the integration of additional sensors on observed objects. Conversely, it could involve nothing more than the commonplace pairing of a regular smartphone with a projector. In the former professional scenario, tasks necessitating exacting precision can be effectively tackled. In contrast, the latter case is tailored to address the needs of a broader user base.

The focus of our research resides in the latter category. It will be demonstrated that even with modest technical support, it is feasible to resolve these challenges with the requisite accuracy [15,20]. This is achievable due to the incorporation of modern smartphones, tablets, and similar devices, which come equipped with the requisite engineering for data capture and analysis [21].

II. Design of the System

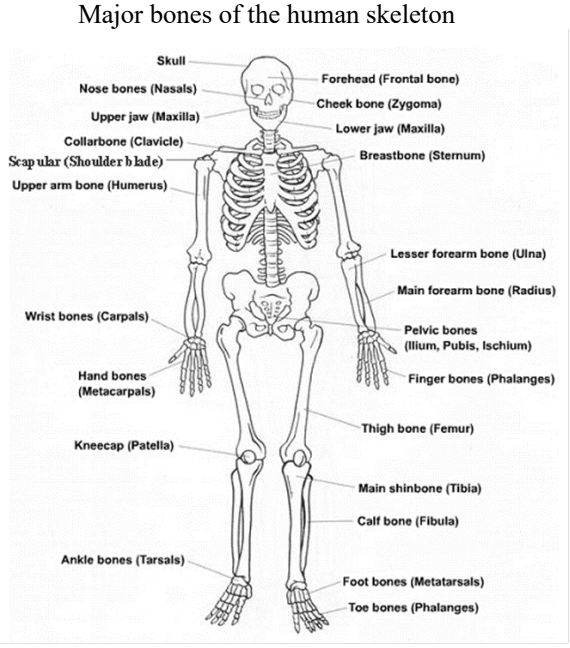
A. Requirements

Skeletal structure and its components:

The main component of the skeleton of biological beings is bones. The number and structure of bones in the same biological individuals are not constant. A person has 360 bones, but there is a difference associated with sex, and this number changes over the years when some components are fused into one. Along with this, there is a reference table indicating all shapes and sizes and their possible defects [12].

The second important component of the skeleton is the mechanism of connection and interaction of bones, which is called a joint. There is a developed system of joint compounds and, of course, a reference system for the types and joints and functionality they are responsible for. At the technical level, types are distinguished such as synovial, cartilaginous and fibrous joints. Together they represented the following structure:

Major bones of the human skeleton



B. What do joints do?

Joints connect bones. They provide stability to the skeleton, and allow movement. There are different types of joints:

a) Synovial joints

Joints in the arms and legs are synovial joints. The ends of the bones are covered with cartilage and separated by the joint cavity, which is filled with a thick gel called synovial fluid. Synovial fluid helps to lubricate the cartilage and provides nourishment to it. Ligaments stretch across the joint, connecting one bone to another and help to stabilize the joint so it can only move in certain directions.

b) Cartilaginous joints

Joints in the spine and pelvis and the joints between the ribs and the sternum are cartilaginous joints — they provide more stability but not as much movement. The bones are connected by cartilage in this type of joint.

c) Fibrous joints

Fibrous joints allow no movement — only stability. They are held together by fibrous connective tissue and located on the skull.

C. Environment of implementation

The project is under implementation in Python and Jupiter notebook. Further to run our program, it will be deployed on the cloud with a docker container. IIAP cloud would be used for the purpose of training, which hosts double Tesla GPUs and Xeon CPUs. Our task will require 2 CPU cores, 8gb of memory and 4gb of GPU. Since the approach suggested in the implantation section relies on DTW algorithm, which is considered as relatively simple for implementation and requires smaller machine resources for the runtime, it can be efficiently implemented for runtime on mobile devices with Swift of Java.

III. Implementation

A. General Description

a) We use the popular HAR (Human activity recognition) dataset from UCI [21], which contains labeled time series. Specifically, an instance of this dataset is a person wearing a smartphone, which captures the linear acceleration and angle velocity while performing one of the following activities (WALKING, WALKING_UPSTAIRS, SITTING, WALKING_DOWNSTAIRS, STANDING, LAYING). Therefore, each observation is a 561-feature vector with time/frequency domain variables and a label describing the person's activity, and the goal is to build a model that accurately predicts the activity using the transformed feed from the smartphone.

b) Time Series Similarity Measures

One of the simplest similarity measures for time series is the Euclidean distance measure. Assume that both time sequences are of the same length n, we can view each sequence as a point in n-dimensional Euclidean space, and define the dissimilarity

between sequences as the familiar Euclidean distance. This measure is simple to understand and easy to compute being the most widely used distance measure for similarity.

c) Longest Common Subsequence Similarity

The longest common subsequence similarity measure is a variation of edit distance used in speech recognition and text pattern matching.

The basic idea is to match two sequences by allowing some elements to be unmatched. The advantage of the LCSS method is that some elements may be unmatched or left out (e.g., outliers), whereas in Euclidean all elements from both sequences must be used, even the outliers.

d) Dynamic Time Warping

More flexible for some applications is the Dynamic Time Warping distance measure, that we use in [22,23]. This is preferable when two sequences have approximately the same overall shapes, but these shapes do not line up in the X-axis. In order to find a similarity between such sequences we must "warp" the time axis of one sequence to achieve a better alignment.

Consider two gesture sequences (of possibly different lengths) to be compared against each other as two time series: $X = (x_1, x_2, \dots, x_{t_1}, \dots x_{T_1})$ and $Y = (y_1, y_2, \dots, y_{t_2}, \dots y_{T_2})$ Using multivariate series, these two sequences form a much larger feature vector for comparison. Evidently, it is impossible to compute a distance metric between two vectors of unequal dimensions. A local cost measure is defined: $d: \mathcal{F} \times \mathcal{F} \to \mathbb{R} > 0$ where $x_{t_1}, y_{t_2} \in \mathcal{F}$ for $t_1 \in [1, T_1], t_2 \in [1, T_2]$. By evaluating the cost matrix for all elements in X and Y, we obtain the matrix $C_{T_1 \times T_2}$. From this local cost matrix, we wish to obtain a correspondence mapping element in X to the elements in Y that will result in the lowest distance measure. We can define this mapping correspondence as f= $c(1), c(2), \dots, c(k), \dots, c(K)$ where $c(k) = c(x_k, y_k)$. The mapping function should follow the time sequence order of the respective gestures. Hence, we impose several conditions on the mapping function:

1. Boundary conditions: the starting and ending observation symbols are aligned to each other for both gestures, $c(1) = c(x_1, y_1)$ and $c(K) = c(x_{T_1}, y_{T_2})$.
2. Monotonic condition: the observation symbols are aligned in the order of time. This is intuitive as the order of observation signals in a gesture signal should not be reversed, $k_1 \leq k_2 \leq \cdots \leq K$.
3. Step size condition: No observation symbols are to be skipped, $k_{i+1} - k_i \leq 1$.

In this way, we arrive at an overall cost function defined as $C(X, Y) = \sum_{k=1}^{K} c(k)$ which gives an overall cost/distance between two gestures according to a warping path, as defined by the function f. Since the function $C(X, Y)$ denotes all possible warping paths between two gesture observation sequences X and Y, the dynamic time warping algorithm is to find the warping path, which gives the lowest cost/distance measure between the two gestures.

In our scenario, we apply dynamic programming principles to calculate the distance to each $c(k)$. We define D as an accumulated cost matrix:

1. Initialize $\mathrm{D}(1,1) = \mathrm{d}(\mathrm{x}_1, \mathrm{y}_1)$,
2. Initialize $\mathrm{D}(\mathrm{T}_1, \mathrm{T}_2) =$ an anarbitrary large number,
3. Calculate $\mathrm{D}(\mathrm{t}_1, \mathrm{t}_2) = \{\min \mathrm{D}(\mathrm{t}_1 - 1, \mathrm{t}_2 - 1), \mathrm{D}(\mathrm{t}_1 - 1, \mathrm{t}_2), \mathrm{D}(\mathrm{t}_1, \mathrm{t}_2 - 1)\}$.

We present orientation using quaternion. Quaternions are a compact and complete representation of rotations in 3D space compared with Euler angles. Quaternions are built from 4 dimension tuples (*W, X, Y, Z*). In a quaternion representation of rotation, singularities are avoided, giving a more efficient and accurate representation of rotational transformations. A quaternion, which is of 4 dimensions, has a norm of 1, and is typically represented by one real dimension and three imaginary dimensions. The three imaginary dimensions, which are i, j, and k, are unit length and orthogonal to one another. $q = xi + yj + zk + w, \quad w^2 + x^2 + y^2 + z^2 = 1, ij = k, ji = -k, ij = -ji$. Quaternions (w, x, y, z) typically represent a rotation about the (x, y, z) axis by an angle of $\alpha = 2cos^{-1}w = 2sin^{-1}\sqrt{x^2 + y^2 + z^2}$.

Although each series consisted of n (number of joints in the model) serialized quaternions, it will be split up into its individual quaternions for metric calculation. The final distance will be the sum of the distance between the n pairs of quaternions.

In simple case, we envision the use of a typical NN algorithm, where there is no specific learning phase. The system stores a list of multivariate time series of known activities and their corresponding labels in a database. When an unknown action is presented to the system, the system takes the unknown time series, and performs a sequential search with lower bounding DTW.

For the sake of implementation of dynamic time wrapping in Python, we will use dtw package code, shown below. In this example, we will use 2 sine signals. One of the phases of sine wave signals is shifted and dtw is used to assign a specific group from signal1 to the second signal.

```
from dtaidistance import dtw
from dtaidistance import dtw_visualisation as
dtwvis
import random
import numpy as npx = np.arange(0, 20, .5)
s1 = np.sin(x)
s2 = np.sin(x - 1)
path = dtw.warping_path(s1, s2)
dtwvis.plot_warping(s1, s2, path)
distance = dtw.distance(s1, s2)
```

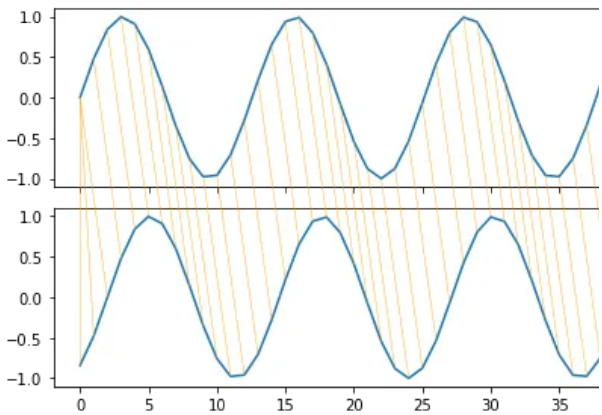


Figure 1: Optimal warping distances between the 2 series

Figure 1 shows the optimal distances between all points of the 2 sine waves. We can also plot the dynamic programming matrix (or accumulated cost matrix), which shows all the warping paths. This is shown in **Figure 2**:

Each cell in **Figure 2** is actually a number, representing the distance between the 2 respective data points being compared, one for each sequence. The darker the color, the lower the distance. After constructing the matrix, the optimal warping path is extracted (red line).

On the other hand, the time complexity is *O(M,N)* where *M, N* are the lengths of the respective sequences - a quadratic cost. Considering that the sequences may be large (not uncommon in real-world examples) as well as the fact that KNN would still have to run afterwards, it is very likely that the model may take too long to get trained.

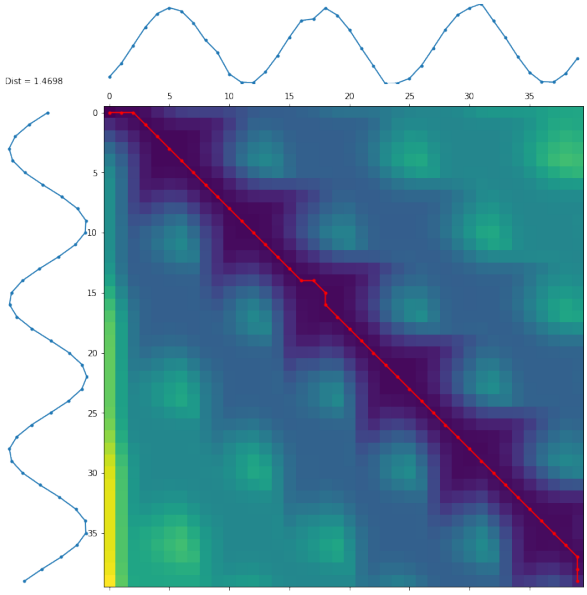


Figure 2. Cost matrix. Red line indicates the optimal warping path

B. *Use Cases and Results*

Dynamic Time Warping emerges as a valuable tool for ***action recognition*** not only in the fitness industry but also across a spectrum of applications. Its scientific foundation and unique ability to align and compare time-series data make it a versatile technique to address the nuanced and dynamic nature of human movements [24-34].

In the fitness and virtual-coaching realm, DTW plays a pivotal role in exercise classification, repetition counting, form assessment, and customized training. This enables the trainers and individuals to personalize workout routines, monitor progress, and enhance overall fitness outcomes. (Figure 3).

In the animation and gaming industry, DTW facilitates the real-time recognition of human gestures and movements, enhancing user experiences through immersive interactions. Gamification of fitness routines becomes more engaging and effective when DTW is employed for action recognition and can be used for interaction with virtual environments.

In the realm of surveillance and human activity analysis, DTW contributes to identifying and classifying human movements in real-time video streams. This has applications in security and safety, where detecting anomalous activities or identifying specific actions in crowded environments is crucial.

While DTW offers numerous benefits, it does face computational challenges, particularly in real-time applications and large-scale datasets [10,28,35]. Future directions include the integration of machine learning techniques to enhance its accuracy and efficiency, and the development of hybrid approaches that combine DTW with other alignment techniques to mitigate computational demands.

In conclusion, Dynamic Time Warping's applicability spans across the fitness industry, self-coaching, animation and gaming, robot management, surveillance, and human activity analysis. Its scientific underpinnings make it a powerful tool for recognizing and analyzing complex time-series data, thereby contributing to advancements in various domains and ultimately enriching human experiences.

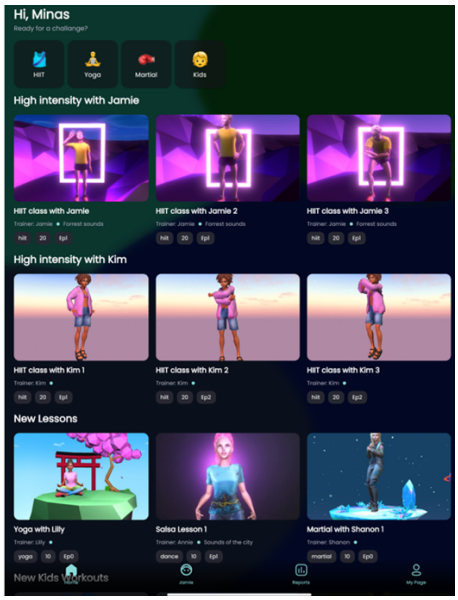


Figure 3. The general menu of the system with basic use cases

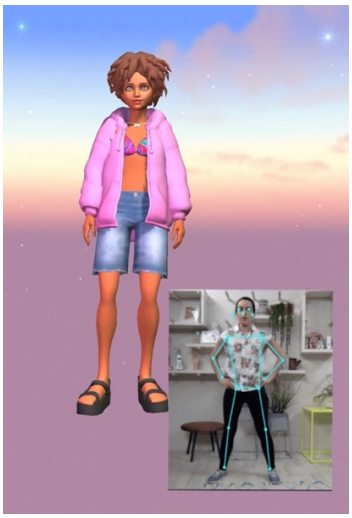


Figure 4. AI-based workout application with a virtual coach (on the main screen) and a trainee (on the supplementary frame) with an estimated pose

ACKNOWLEDGMENT

The work is partially supported by grant No21T-1B314 of the Science Committee of MESCS RA.

REFERANCES

[1] V. Kachurka *et al*., "WeCo-SLAM: Wearable Cooperative SLAM System for Real-Time Indoor Localization Under Challenging Conditions", *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5122-5132, 2021

[2] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation", *CVPR*, 2009.

[3] L. Doherty, K. S. J. pister and L. El Ghaoui, "Convex position estimation in wireless sensor networks", *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, USA, vol. 3, pp. 1655-1663, 2001

[4] A. Faisal, S. Majumder, T. Mondal, D. Cowan, S. Naseh, M.J. Deen, "Monitoring methods of human body joints: State-of-the-art and research challenges", *Sensors 19*, no. 11, 2019

[5] H. Sun, N. Guanghan, Z. Zhiqun, H. Zhongchao and H. Zhihai, "Automated work efficiency analysis for smart manufacturing using human pose tracking and temporal action localization", *Journal of Visual Communication and Image Representation* 73, 2020

[6] N. A. Vikhreva, "Dance Writing", *The Moscow State Academy of Choreography*, Moscow, 2006.

[7] N. A. Vikhreva, "History of Dance Writing", *The Moscow State Academy of Choreography*, Moscow, 2014.

[8] M. Eichner and V. Ferrari, "Better appearance models for pictorial structures", *Conference: British Machine Vision Conference, BMVC* London, 2009.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition", *International Journal of Computer Vision*, pp. 55–79, 2005.

[10] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection", *Computer Vision and Pattern Recognition (CVPR)*, *IEEE Conference*, pp. 3476–3483, 2013.

[11] C. Szegedy, A. Toshev, and D. Erhan, "Object detection via deep neural networks", *NIPS 26*, 2013.

[12] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human pose estimation using body parts dependent joint regressors", *CVPR*, 2013.

[13] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures" *IEEE Transactions on Computers*, vol. C-22, pp. 67–92, 1973.

[14] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet conditioned pictorial structures", *CVPR*, 2013.

[15] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation", *CVPR*, 2011.

[16] G. Mori and J. Malik, "Estimating human body configurations using shape context matching", *ECCV*, 2002.

[17] G. Gkioxari, P. Arbel´aez, L. Bourdev, and J. Malik, "Articulated pose estimation using discriminative armlet classifiers" In *CVPR*, 2013.

[18] C. Ionescu, F. Li, and C. Sminchisescu, "Latent structured models for human pose estimation", *ICCV*, 2011.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks", *NIPS*, 2012.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", *CVPR*, 2014.

[21] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto and X. Parra, "Human Activity Recognition Using Smartphones", *UCI Machine Learning Repository*, 2012.

[22] S. Samsu, N.U. Maulidevi, and P. R. Aryan, "Human action recognition using dynamic time warping", *Proceedings of the 2011 international conference on electrical engineering and informatics, IEEE*, 2011.

[23] P.C. Huu, Q.K. Le, and T.H. Le, "Human action recognition using dynamic time warping and voting algorithm", *VNU Journal of Science: Computer Science and Communication Engineering,* 2014.

[24] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization", *COLT. ACL*, 2010.

[25] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, "Articulated human pose estimation and search in (almost) unconstrained still images", *International Journal of Computer Vision*, 2012.

[26] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation", *BMVC*, 2010.

[27] Y. Yang and D. Ramanan. "Articulated pose estimation with flexible mixtures-of-parts", *CVPR*, 2011.

[28] M. Osadchy, Y. LeCun, and M. L. Miller, "Synergistic face detection and pose estimation with energy-based models", *The Journal of Machine Learning Research,* 2007.

[29] D. Ramanan, "Learning to parse images of articulated bodies", *NIPS*, 2006.

[30] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation", *CVPR*, 2013.

[31] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing", *CVPR*, 2003.

[32] G. W. Taylor, R. Fergus, G. Williams, I. Spiro, and C. Bregler, "Pose-sensitive embedding by nonlinear NCA regression", *NIPS*, 2010.

[33] Y. Tian, C. L. Zitnick, and S. G. Narasimhan, "Exploring the spatial hierarchy of mixture models for human pose estimation", *ECCV*, 2012.

[34] F.Wang and Y. Li, "Beyond physical connections: Tree models in human pose estimation", *CVPR*, 2013.

[35] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation"*, CVPR*, 2008.