Graph-Based Analysis of Physical Exercise Actions

Oya Çeliktutan[1,2], Ceyhun Burak Akgül[1,3], Christian Wolf[2], Bülent Sankur[1]

[1]Electrical and Electronics Engineering, Boğaziçi University, Istanbul, Turkey
[2]Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR CNRS 5205, F-69621, France
[3]Vistek ISRA Vision, Istanbul, Turkey
oya.celiktutan@boun.edu.tr; cb.akgul@gmail.com;
christian.wolf@liris.cnrs.fr; bulent.sankur@boun.edu.tr

ABSTRACT

In this paper, we develop a graph-based method to align two dynamic skeleton sequences, and apply it to both action recognition tasks as well as to the objective quantification of the goodness of the action performance. The automated measurement of “action quality" has potential to be used to monitor action imitations, for example, during a physical therapy. We seek matches between a query sequence and model sequences selected with graph mining. The best matches are obtained through minimizing an energy function that jointly measures space and time domain deformations. This measure has been used for recognizing actions, for separating acceptable and unacceptable action performances, or as a continuous quantification of the action performance goodness. Experimental evaluation demonstrates the improved results of our scheme vis-à-vis its nearest competitors. Furthermore, a plausible relationship has been obtained between action perturbation, given by the joint noise variances, and quality measure, given by matching energies averaged over a sequence.

Categories and Subject Descriptors

G.2.2 [**Discrete Mathematics**]: Applications; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis–*depth cues, motion, tracking*; I.5.4 [**Pattern Recognition**]: Applications–*computer vision*

Keywords

Hyper-graph matching, action recognition, action quality assessment, sequence alignment

1. INTRODUCTION

Action/activity recognition has captured the interest of the computer vision community due to its practical and wide range of application areas. Recently, many researchers have shifted their focus onto gesture and/or action analysis based on the depth data with the development of low-cost consumer cameras. Video game consoles such as XBox 360, virtual dressing rooms, customizable hand gesture recognition interfaces [11], smart video surveillance, physical therapy instruction can be given as concrete examples for real-world applications. One advantage of depth modality is that it can potentially mitigate the problems encountered in the presence of severe illumination artifacts, camera view-point change and dynamic complex backgrounds. Secondly, more useful representations of human body movements can be obtained based on depth maps and point clouds. A case in point is the real-time skeleton tracking algorithm developed by Shotton et al. [19]. This scheme results in 20 body (skeleton) joints per frame and proves to be very robust to varying background, appearance and viewpoint changes for tracking.

In this paper, we use depth sequences recorded with a Kinect camera [14] and the tracked skeleton joints [19] for action recognition. We also focus on a specific problem in the action recognition domain, that is, automatic action quality assessment. Our goal is to evaluate the performance of a person imitating an action as guided by instructions. These instructions can be in the form of a video of an agent performing an action, a text describing the kinematic details of an action or an instructor/person in interaction. Instances of this problem in the literature are evaluation of karate performance [2], of dancer's performance [9], quality assessment of reach and grasp movements of stroke survivors [20], and the gaming/entertainment industry. In this work, we aim in particular to assess the quality of physical exercises for therapeutic purposes.

State-of-the art methods for skeleton-based action recognition can be coarsely characterized by two main steps. First, a skeleton is transformed into a feature representation robust to intrinsic properties such as body size, variance within the action class etc., and extrinsic conditions like camera position. Secondly, model skeleton sequences and a scene sequence are temporally aligned to each other and compared with respect to their kinematic details, e.g., speed, amplitude, position of skeleton joints, trajectory etc. The emphasis of this work is temporal alignment rather than feature extraction step. The available approaches to temporal alignment typically utilize three methods or variations of them: Dynamic Time Warping (DTW) [2, 5], Hidden Markov Models (HMM) [23] and correlation [1, 16]. The widely used Dynamic Time Warping (DTW) is an appropriate technique for matching two sequences in different lengths. Our proposed method is related to DTW as it exploits the locality constraints and warping operations (insertion or deletion) in a similar way. In contrast, our graph-based formulation allows to take into account much richer structural information encoded in spatio-temporal geometry and sequential nature simultaneously.

In this work, we customize our hyper-graph-based method introduced in [3, 4] for aligning the tracked skeleton sequences. We first model the spatio-temporal relationship between the skeleton

MIIRH'13, October 22, 2013, Barcelona, Spain.
Copyright 2013 ACM 978-1-4503-2398-7/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505323.2505330.

joints by a chain graphical structure. Then, in order to find the model graph corresponding to a scene graph, we minimize an energy function formulated as the matching of two graphs. Since the in-frame point correspondence problem is already solved by skeleton-tracking algorithm, the matching problem is reduced to jointly ensuring the spatial geometry consistency between joints and the temporal alignment. The reduced problem seems to be somewhat easier as compared to classical unconstrained graph matching problem [7], but it is still very challenging due to spikes, noisy or missing data delivered by the Kinect system or severe occlusions. Once the skeleton sequences are aligned, the quality is measured based on the spatial and temporal deformation measures with respect to a reference subject, i.e., an instructor.

Related literature on action recognition. Previous works using depth modality mostly attempted to extend action recognition approaches to the depth domain. Xia et al. [23] represented each skeleton as a histogram of joint positions that was obtained by converting Cartesian coordinates of the joint positions to spherical coordinates and dividing the sphere enclosing the skeleton into several grids. Each action video was described by a sequence of poses, which were actually characterized by histograms, and Hidden Markov Models (HMMs) were used for classification. The concept of the bag of visual words was adapted to 3D points by Li et al. [12]. Bag of 3D points were used to characterize the salient poses, which were probabilistically modeled by so called "action graphs".

Introducing a skeleton joint weighting or a feature selection mechanism has been shown to improve the action recognition performance in previous studies [21, 17, 5, 16]. Wang et al. [21] proposed a discriminative joint mining approach in which the goal was to select the best subset of joints that increases a confidence measure and at the same time decreases an ambiguity measure. Their learning scheme resulted both in most informative joints and training sequences per each action class. Ofli et al. [17] divided each sequence into small temporal segments, namely M-frame length subsequences. They took into account a number joints having maximum variance within each subsequence and then used these most informative joints to construct histograms for action recognition. In another work, Çelebi et al. [5] used weighted DTW where they learned the weights of each joint during a training phase.

A novel robust skeleton representation was proposed by Raptis et al. [18]. Briefly, Principal Component Analysis (PCA) is applied on the torso joint positions. The resulting basis, called as "torso frame", is used to estimate the orientation of the human body and accordingly to extract a set of features, e.g., limb joint angles with respect to the torso frame. These features were used to train cascaded correlation-based classifiers and the reliability of matching two sequences was evaluated by a distance metric based on DTW. The most recent methods have been built on random decision forests [15, 16]. Miranda et al. [15] extracted similar features as in [18]. They first learned a set of key poses with multi-class Support Vector Machines (SVMs), and then fed these poses to the random decision forests for action recognition. Negin et al. [16] proposed a correlation-based method that also benefits from torso frame features [18] and employed random decision forests to learn the discriminative features per action class. Finally, Ellis et al. [8] used a logistic regression based classifier, and GentleBoost to select a set of best features corresponding to the informative joints.

Related literature on quality assessment. In the literature, we have encountered only few methods for automatic action quality assessment. These prior works have used Motion Capture (MoCap) data [6, 10] or RGB video sequences [13]. A few approaches have been recently proposed based on skeleton tracking. For example, in [2], Bianco and Tisato recognize Karate moves based on skeletal joints. They select triplets of joints manually for hand and foot techniques and represent each move by the angles of joint triplets. A set of key poses is obtained via K-means clustering and Dynamic Time Warping (DTW) is used for aligning two sequences of poses. The resulting normalized DTW distance is used for performance evaluation and a regression analysis is done in order to validate the relationship between the DTW distance and the subjective scoring. Essid et al. [9] used three different scores and then combined them for salsa dance performance evaluation. These scores are computed based on positions, velocities and 3D motion (flow) of the joints. This method uses quaternionic correlation to estimate the time-shift between two dancing sequences. Finally, Venkataraman et al. [20] also proposed a correlation-based approach to quantify the movements of stroke survivors. They differently model the movements of a human from a dynamical system perspective and reconstruct a phase space to infer geometrical and topological information from observations by means of attractor. Their features rely on the shape features from the reconstructed phase space. However, the main drawback of these methods is the use of correlation, as it does not incorporate time warping.

Contributions. In this paper, we treat the sequence alignment as a graph matching problem. We enrich the angle-based pose descriptor proposed in [2] and concatenate it with a distance-based pose descriptor. We also propose an effective method for elimination irrelevant model graphs. Experimental results demonstrate that the proposed framework is useful both for the quality assessment of physical exercises and action recognition.

Outline. In Section 2, we introduce our approach for descriptor extraction and sequence alignment. Section 3 describes our method for action quality assessment. Finally, in Section 4 gives the experimental results and Section 5 concludes.

2. METHODOLOGY

This section describes our proposed approach, covering its three important aspects: i) skeleton representation (normalization and pose descriptor extraction); ii) sequence alignment by graph matching; iii) performance assessment.

2.1 Pose Descriptor Extraction

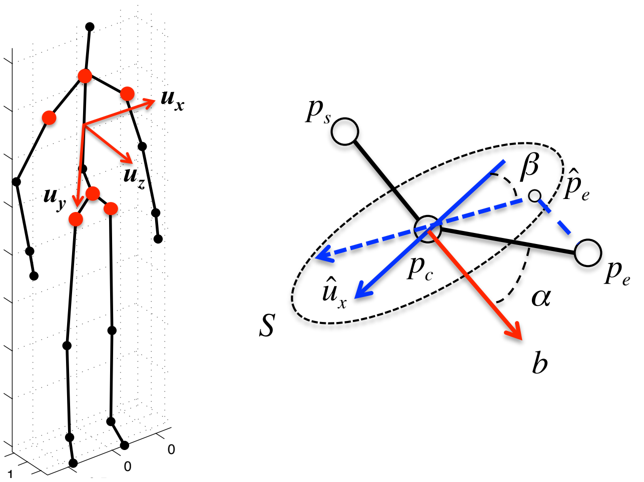
Prior to any feature extraction, we applied a preprocessing stage to normalize skeletons. Each skeleton joint n is represented at a time instant i by its 3 coordinates $p_i(n) = [\ x_i\ y_i\ z_i\ ]$. First, the skeletons are normalized in order to render each skeleton independent from position and body size. For each time instant (frame), we scaled the Euclidean distance between connected skeleton joints so that the inner distance between the hip and the center of shoulders is set to unit length, and then we translated joint positions so that the hip center coincides with the origin of the coordinate system. Secondly, we applied vector median filtering on the time-trajectories of each joint, where for the nth joint the M long joint coordinate sequence is given by

$$P(n) = \begin{bmatrix} x_1 & x_2 & \dots & x_M \\ y_1 & y_2 & \dots & y_M \\ z_1 & z_2 & \dots & z_M \end{bmatrix}.$$

Vector median filtering aims at removing possible spikes and reducing noise. Following the preprocessing stage, we extract two types of pose descriptors for action recognition and quality assessment.

2.1.1 *Angle-based pose descriptor*

In the literature, it has been shown that scale and orientation-invariant features can be obtained by using angles between consecutive joints.



(a) (b)

Figure 2: (a) Illustration of torso basis; (b) Illustration of the spherical coordinates, i.e., radius R, inclination angle α_j and azimuth angle β_j, defined based on the torso frame.

Each skeleton is represented by 14 angles as illustrated in Figure 1. Bianco et al. [2] characterized each joint with only the subtended angle. We believe that to fully describe the orientation in 3D, one should calculate both the subtended angle and the orientation angle of the plane defined by the three points (joints).

We calculate the inclination and azimuth angles with respect to the torso basis proposed by Raptis et al. in [18]. We apply PCA to six torso joint positions (shoulder left/right/center and hip left/right/center) and find a torso basis $\{u_x, u_y, u_z\}$ in which u_x aligns with the line that connects the shoulders, u_y coincides with the line along the spine, and finally, u_z corresponds to depth directional vector. The torso joints and the calculated angles are illustrated in Figure 2. Each angle is defined by three joints p_s, p_c and p_e at a time instant. For simplicity, we ignore the time index i and the joint index n for the moment. As illustrated in Figure 2-b, the vector b, extension of the vector $\overrightarrow{p_s p_c}$, is the normal of the plane S centered at the p_c. We calculate the angles as follows:

- The inclination angle $\alpha(n)$ is computed between $\overrightarrow{p_c p_e}$ and b.
- The azimuth angle $\beta(n)$ is defined between $\hat{u}_x$ and $\hat{p}_e$ which are the projections of u_x and p_e on the plane S, respectively.
- We also compute the angle η between the directional vector z from the depth sensor and the body orientation vector u_z from the torso basis, to measure the bending of the body.

Thus, we can define the angle-based pose descriptor vector as $\mathbf{a} = \{a(1), \ldots, a(14), \eta\} \in R^{29}$ where $a(n) = \{\alpha(n), \beta(n)\}$.

2.1.2 *Distance-based pose descriptor*

Complementarily, we calculate the Euclidean distance between joint-position pairs, $p(n)$ and $p(k)$. Let N be the number of joints. In our experiments, we ignore the joints related to hand and foot, since these are more prone to errors, and set $N = 15$. We define the distance-based pose descriptor vector as $\mathbf{d} = \{d(n) | n = 1, 2, \ldots 15\} \in R^{210}$ where $d(n) = \{d(n,k) | k \neq n\} \in R^{14}$ and $d(n,k) = \|p(n) - p(k)\|_2$.

Finally, we simply obtain the joint pose descriptor vector $\mathbf{f}$ as the concatenation of the two pose descriptor vectors, $\mathbf{f} = \{\mathbf{a}, \mathbf{d}\}$.

2.1.3 *Pose quantization*

We quantize the skeleton pose space in order to decrease redundancy in the temporal domain. Since K-means clustering is widely used for extracting the key poses [2, 23], we adapted it to cluster skeleton poses represented by the joint pose descriptor $\mathbf{f}$ and obtained a set of key poses. Given a sequence of key poses, we sample and assign each skeleton at a time instant to its closest cluster center, i.e., its closest key pose, and obtain an abstract representation for each skeleton sequence.

2.2 Graph-based Sequence Alignment

The widely used graph matching technique provides a powerful solution to feature correspondence problem by dealing with the structural information. Consider a graph $G = \{V, \mathcal{E}\}$ in a typical computer vision problem. Interest points constitute the nodes V in the graph. Edge set $\mathcal{E}$ models the structural relationships between the nodes. Matching problem is basically searching the best correspondence between two graphs, i.e., the one represents the model - a model graph - and the other one that represent the scene - a scene graph. However, its major drawback is that the useful formulations are known to be an NP-hard combinatorial problem. For this reason, a great effort has gone to optimization of graph matching algorithms in the machine learning community [7].

In [3, 4], we demonstrated that, in the case of spatio-temporal data, hyper-graph[1]matching problem can be solved with bounded complexity by taking into account the sequential nature of time and causality of human actions. In this work, we customize the same approach for aligning the tracked skeleton sequences. In this context, each action sequence is structured into a chain graph where the nodes V coincide with the frames and the edges $\mathcal{E}$ model the temporal relationship. Namely, in our case, each node is associated to exactly one frame characterized by a skeleton, an edge indirectly models the temporal relationship between the joints of the skeleton. The overall goal is to find a set of frame assignments x between two sequences - two graphs - which is formulated as a global solution of an energy minimization problem. Despite the fact that Kinect system provides the positions of the skeleton joints, the problem remains still very challenging in nature due to noisy, missing or occluded joints. Graph matching offers a robust and flexible solution to the sequence alignment problem against these problems.

2.2.1 *A chain graphical structure*

Let a model sequence and a scene sequence be represented by their respective graphs $G^m = \{V_i^m, i = 1 \ldots M\}$ and $G^s = \{V_j^s, j = 1 \ldots S\}$. The definitions and notations used in the rest of the paper can be summarized as follows:

- G and V represent a generic graph and a node (skeleton or frame), respectively.
- The superscripts "m" and "s" denote the model and scene sequence. The subscript "i" and "j" index the time instants (a discrete frame number) to the nodes. The index "n" indicates the nth skeleton joint.
- Each node (frame) is characterized by the 3 dimensional cartesian coordinates of the joints, $p_i^m(n)$, $p_i^s(n)$, $n = 1 \ldots N$, a pose descriptor vector, $\mathbf{f}_i^m$, $\mathbf{f}_i^s$, and a time instant t_i.
- M and S are the respective number of model and scene nodes where typically $M << S$.

[1] Hyper-graphs are a generalization of graphs allowing for edges, so called hyper-edges, to connect any number of nodes, typically more than two [24].

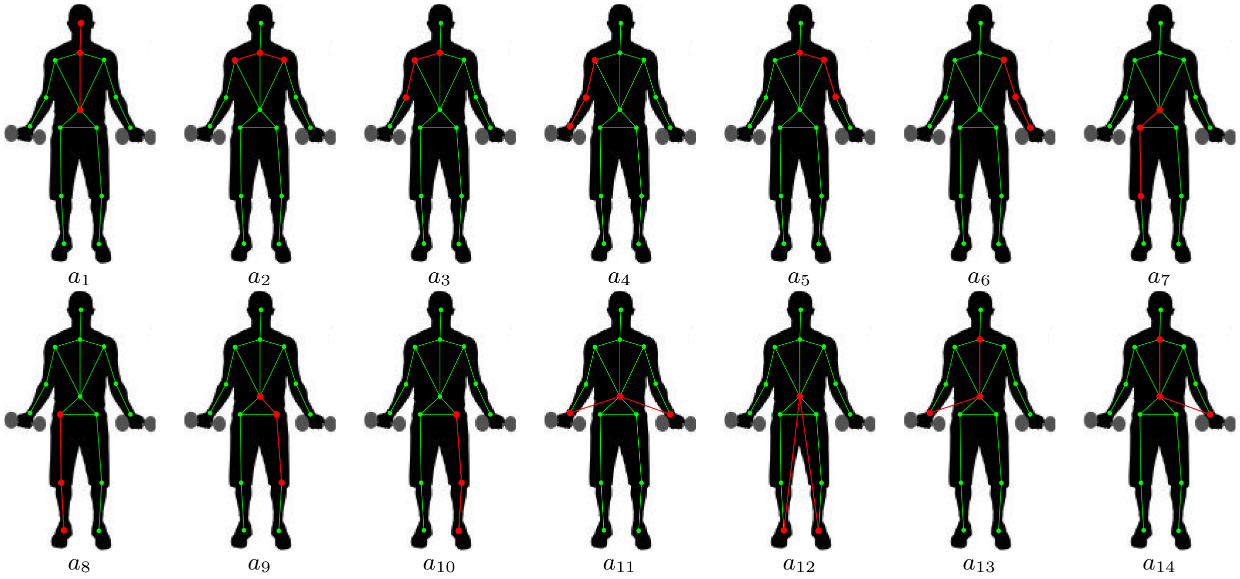


Figure 1: Angles computed for skeleton representation. This figure is best viewed in color.

We model each sequence as a chain graphical structure by taking into account second order dependencies. More explicitly, let $\mathcal{E}_{ijk}$ denotes the edge between the nodes i,j and k in G; the triplet (i, j, k) holds an edge, $\mathcal{E}_{ijk} = 1$, if $|t_i - t_j| < T \wedge |t_j - t_k| < T$, else $\mathcal{E}_{ijk} = 0$. Note that T is a threshold that limits the distance between the nodes in the time domain. As illustrated in Figure 3, this results in a planar graph where each node is connected to its two preceding and succeeding nodes. Each triangle models the temporal relationship of three consecutive skeletons. Note that linking three nodes allows to include ternary potentials in the energy function, and hence provides scale and rotation-invariance.

2.2.2 *Matching*

We denote an assignment of the jth node of the scene sequence, $V^s_{x_i}$, to the ith model node, V^m_i, by $x_i = j$. Our goal is to find a set of assignments $x = \{x_1, x_2, \ldots x_M\}$ minimizing an energy function E where each x_i can take values from $j = 1 \ldots S$. To handle unreliable matches or occlusions, we also admit a dummy variable $x_i = \epsilon$ when there is no plausible assignment.

Taking into account the chain graphical structure, we can formalize the energy function as below

$$E(x) = \sum_{i=1}^{M} U(x_i) + \lambda \sum_{j=3}^{M} D(x_i, x_{i-1}, x_{i-2}) \tag{1}$$

where $U(.)$ measures the deformation between a model node and a potentially assigned scene node to it, $D(.)$ is the cost for assigning three scene nodes to the three consecutive frames in the model sequence and λ is the weighting parameter. On the one side, $U(.)$ is defined as the Euclidean distance between the pose descriptors of joint triples

$$U(x_i) = \begin{cases} W^d & \text{if } x_i = \epsilon, \\ \|\mathbf{f}^m_i - \mathbf{f}^s_{x_i}\| & \text{otherwise} \end{cases} \tag{2}$$

where W^d is the dummy assignment penalty.

On the other side, $D(.)$ measures the spatio-temporal deformation through two terms, a time warping penalty term $D_t(.)$ and spatio-temporal geometry term $D_g(.)$:

$$D(x_i, x_{i-1}, x_{i-2}) = D_t(x_i, x_{i-1}, x_{i-2}) + D_g(x_i, x_{i-1}, x_{i-2}). \tag{3}$$

$D_t(.)$ is simply the truncated time differences between the model nodes and their corresponding scene nodes:

$$D_t(x_i, x_{i-1}, x_{i-2}) = \begin{cases} \Delta(x_i, x_{i-1}) + \Delta(x_{i-1}, x_{i-2}) & \text{if } |t_{x_i} - t_{x_{i-1}}| < T \wedge \\ & \quad |t_{x_{i-1}} - t_{x_{i-2}}| < T, \\ W^t & \text{otherwise.} \end{cases} \tag{4}$$

Here, $\Delta(.)$ penalizes the offset in time and measures the time interval differences in the assignment of model node pair (i, j) to scene node pair (x_i, x_j)

$$\Delta(x_i, x_j) = |(t_i - t_j) - (t_{x_i} - t_{x_j})|. \tag{5}$$

Spatio-temporal deformation, $D_g(.)$, is measured by summing the deformation over the triplets of consecutive joints. More explicitly, each joint triple constitutes a triangle where its vertices correspond to the 3 dimensional cartesian coordinates of nth joint in three consecutive frames (nodes), namely, $\{i, i-1, i-2\}$. $D_g(.)$ is defined as the distance between the angles of the triangle formed by the model triplet $P^m_c(n) = \{p^m_i(n), p^m_{i-1}(n), p^m_{i-2}(n)\}$ and the angles of the triangle formed by its corresponding scene triplet $P^s_{x_c}(n) = \{p^s_{x_i}(n), p^s_{x_{i-1}}(n), p^s_{x_{i-2}}(n)\}$:

$$D_g(x_i, x_{i-1}, x_{i-2}) = \sum_{n=1}^{N} \|\phi(P^m_c(n)) - \phi(P^s_{x_c}(n)))\| \tag{6}$$

where $\phi(.)$ gives the triangle interior angles and the differences are summed over N body joints. Note that the use of triangles provide us to define a measure invariant to scale changes unlike the pairwise geometric relationships.

2.2.3 *Dynamic programming*

We efficiently solve the matching problem formalized in Equation 1 by using dynamic programming technique similar to the extended Viterbi algorithm for decoding second order Markov Chains. The

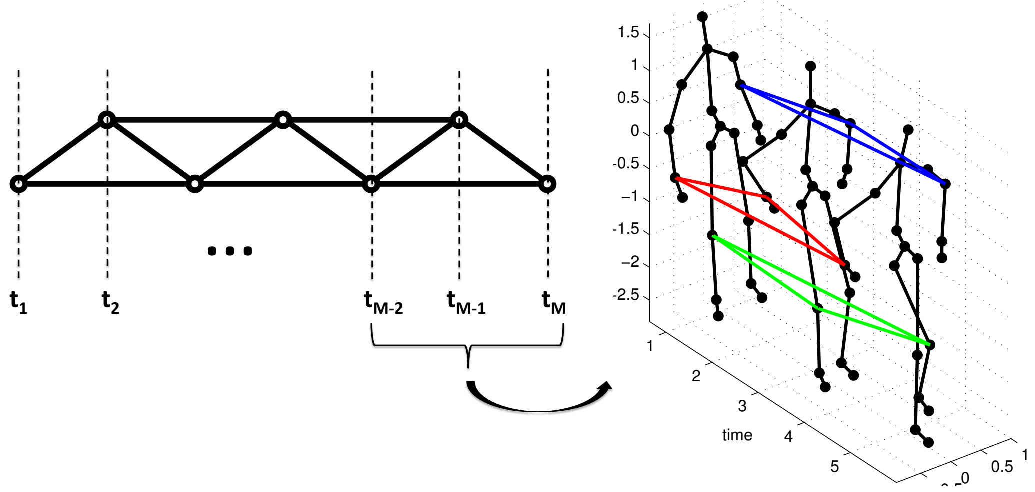


Figure 3: A planar graph with triangular structure. This graph describes the restrictions on the temporal coordinates. Each triangle models the spatial relationship between three consecutive skeletons.

basic idea is to decompose the problem into a set of subproblems such that each subproblem can be solved recursively in terms of the others. Let $B_{i+1}(x_i, x_{i-1})$ denote the cost of the best assignment of values to the last $(M-i)$ nodes with the constraint that the $(i+1)$th node has the assignment x_{i+1}. The algorithm works filling in tables storing cumulative costs of optimal partial solutions. Subproblems can be solved by using the recursive equation:

$$B_i(x_{i-1}, x_{i-2}) = \min_{x_i} \left[U(x_i) + D(x_i, x_{i-1}, x_{i-2}) + B_{i+1}(x_i, x_{i-1})\right]. \quad (7)$$

with the initialization

$$B_M(x_{M-1}, x_{M-2}) = \min_{x_M} \left[U(x_M) + D(x_M, x_{M-1}, x_{M-2})\right]. \quad (8)$$

After the tables are filled in, we can find a global minimum of the energy function by backtracking:

$$x_i^* = \arg\min_{x_i} B_i(x_{i-1}, x_{i-2}). \quad (9)$$

During matching, we bound the complexity by taking into account the constraints due to the sequential nature of time. These constraints can be explained as follows.

- Causality: The temporal order of the nodes should be preserved in a correct match, namely, $t_{x_{i-2}} < t_{x_{i-1}}$ and $t_{x_{i-1}} < t_{x_i}$.

- Temporal closeness: This constraint is formalized in Equation 4. The model node pairs should not be too far from each other; and likewise the scene node pairs, that is, if $|t_i - t_j| < T$, $|t_{x_i} - t_{x_j}| < T$ where T is a small value threshold that bounds the node pair differences in time.

Note that the computational complexity is reduced to $O(MST^2)$ from $O(MS^3)$ under these constraints. Before delving into how these measures can be used for action recognition and action quality assessment, we introduce a method for learning representative and discriminative model graphs in the sequel.

2.3 Learning Model Graphs

We add a step of learning and mining prototypes, which has been frequently reported to increase classification accuracy [21]. Indeed, the choice of representative and discriminative model graphs in a learned dictionary adds additional information to the information carried by the graphs themselves.

Given a set of L model graph prototypes G_j^m obtained from a training set, our goal is to find a subset of representative and discriminative graph prototypes. Let E_i^j be the energy of a model graph G_j^m matched to a scene graph G_i^s which is explicitly formalized in Equation 1. Inspired from [22], we define two measures: discriminability measure and representability measure.

Discriminability measure for G_j^m is formulated as the ratio of between-class variance to within-class variance:

$$DIS_j = \frac{\sum_{k=1}^{K} N_k (\bar{\mu}_k^j - \bar{\mu}^j)^2}{\sum_{k=1}^{K} \sum_{i \in C_k} (\mu_i^j - \bar{\mu}_k^j)^2} \quad (10)$$

where K is the number of action classes. Let N_k be number of samples in each action class C_k, $\bar{\mu}_k^j$ and $\bar{\mu}^j$ can be defined as follows:

$$\bar{\mu}_k^j = \tfrac{1}{N_k} \textstyle\sum_{i \in C_k} E_i^j; \quad \bar{\mu}^j = \tfrac{1}{\sum_{k=1}^{K} N_k} \textstyle\sum_{k=1}^{K} \bar{\mu}_k^j. \quad (11)$$

Representability measure, REP_j, evaluates the ratio of the scene graphs that are covered by the model graph G_j^m. Given a model graph and a scene graph pertaining to the same action class, the model graph covers the scene graph, if it is in the l-nearest neighborhood of the scene graph. REP_j is therefore defined as the ratio of the number of covered scene graphs to all scene graphs.

We choose the model graphs satisfying these two requirements. In our experiments, we normalize these two measures and calculate the mean value. We choose the model graphs that have higher mean and set the number of selected prototypes that gives the best performance on the training set.

3. ACTION QUALITY ASSESSMENT

We test and assess the utility of our scheme in two separate frameworks: i) To classify a variety of action sequences into respective “correctly performed" and “wrongly performed" classes; ii) To automatically grade the similitude of an action as compared with a model sequence, as a measure of the actor’s performance level. In the context of quality assessment, the correct sequences (the model set) will typically be the actions of an instructor or a software agent, and the test sequences, which can prove correct or incorrect, will be those of a novice performer.

3.1 Classification: correct vs. wrong

Given a set of selected model graphs and a set of scene graphs labeled as correct or incorrect, the goal is to discriminate action sequences that are deemed to be wrongly performed from correctly performed ones. For this purpose, we define a feature vector for each scene graph G_i^s in terms of the matching energy to the model graphs $\{G_j^m, j = 1 \ldots L\}$ in the library. The feature vector has the form $F_i^s = \{E_i^j, j = 1 \ldots L\}$, where E_i^j is the matching energy in Equation 1 between the model graph G_j^m and the scene graph G_i^s, and L is the number of model graphs. Then using the feature vector, F_i^s, we train a linear Support Vector Machine (SVM) separately for each action type to classify sequences as correctly and wrongly performed. Notice that due to the intra-variation of action sequences, we cannot use a single prototype sequence, but we aggregate via SVM the scores of the distances of the test sequence to all realizations of the action within its class. In other words, there are multiple ways of doing the action wrongly, and there are variations among the correct ones.

3.2 Quality Assessment

The second task is to gauge the goodness of the performance given that the action has been performed correctly. To this effect we can use the matching energy per sequence, or per frame, per limb, or even per joint. Consider two aligned sequences, where MT is some model sequence and ST some scene sequence aligned to the model sequence. These sequences are characterized by their respective sequences of pose descriptors, namely $MT = \{\mathbf{f}_i^m; i = 1 \ldots M\}$ and $ST = \{\mathbf{f}_i^s; i = 1 \ldots M\}$. Our goal is to infer the offsets between the joints in MT and ST

The offset of a joint n of an descriptor between the pair of $\mathbf{f}_i^m(n)$ and $\mathbf{f}_i^s(n)$ at a time instant i is computed as

$$\delta_n = |\mathbf{f}_i^m(n) - \mathbf{f}_i^s(n)|. \qquad (12)$$

We define $\Delta_i = \sum_{n=1}^{N} \delta_n + \delta_t$ for each skeleton of N joints, δ_t as the time difference, and $d = \{\Delta_1, \ldots, \Delta_M\}$ for the entire sequence of M frames. A quality measure (QM) is defined in terms of the overall distance, i.e., $QM = \sum_{m=1}^{M} \Delta_m$. We have observed, as expected that QM (distance) is proportional to the deterioration in the action quality.

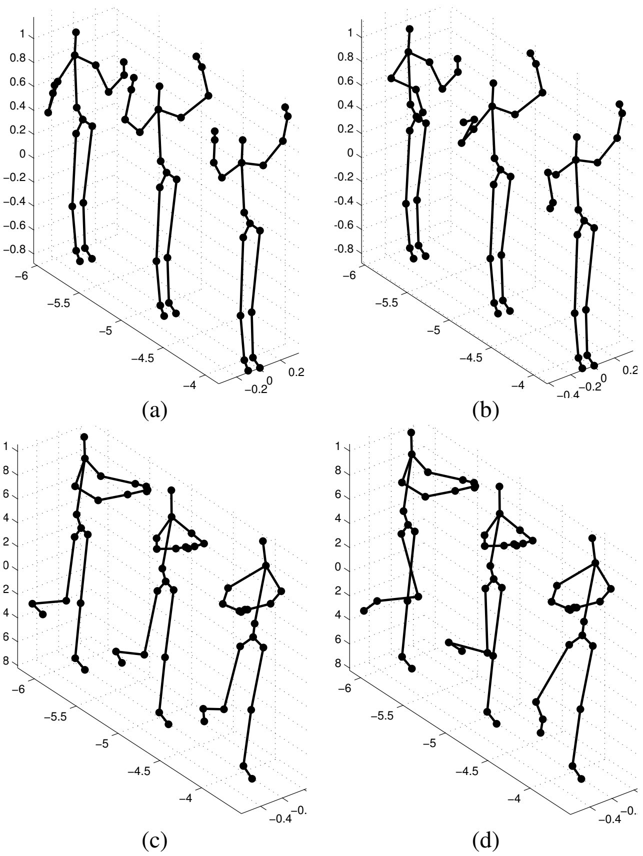
This distance measure can be also specialized for different subsets of body joints, for example upper body or lower body joints or to only one joint in order pinpoint the deficiencies or differences between the performance of the test subject and that of the instructor.

4. EXPERIMENTS AND RESULTS

The proposed framework was tested on MSR Action 3D dataset [12] and WorkoutSu-10 Gesture dataset [16]. Experimental evaluations have demonstrated the utility of the proposed scheme for the following two tasks: action recognition and action quality assessment.

4.1 Datasets and Experimental Setup

MSR Action 3D Dataset [12]. MSR Action 3D dataset is one of the early collections recorded with a depth sensor. There are 20 actions performed by 10 subjects. Each subject performs each action 2 or 3 times, resulting in 567 recordings in total. However, we have used 557 recordings in our experiments as in [21]. Each recording contains a depth map sequence with a resolution of 640 × 480 and the corresponding 20 skeleton joints. Actions are selected in the context of interacting with game consoles, for example, arm wave, forward kick, tennis serve etc.



(a) (b)

(c) (d)

Figure 4: Example illustrations. (a)-(b) Respective original sequence and perturbed sequence of action type C1; (c)-(d) Respective original sequence and perturbed sequence of action type A2. Note that, while (b) is a severe noise case, (d) is an example of mild noise case.

Workout SU-10 Gesture dataset [16]. This dataset has the same recording format as MSR Action 3D dataset. However, the context is physical exercises performed for therapeutic purposes. Lateral stepping, hip adductor stretch, freestanding squats, oblique stretch etc. can be given as examples of exercises (actions). There are 15 subjects and 10 different exercises. Each subject repeats an exercise 10 times, resulting in 1500 action sequences in total. In our experiments, we used 600 sequences for training and tested our algorithm on the unseen part of the dataset (we ignored one subject, and tested on 800 sequences).

The action types used in the experiments are listed in Table 1.

Perturbed dataset. We used Workout SU-10 Gesture dataset [16] for action quality assessment. However, this dataset is lacking meta-data, in other words, the subjects or the sequences are not labeled with subjective scores by one or more experts. For this reason, we decided to perturb these sequences artificially with Gaussian noise added at the joint positions. We considered mild to severe perturbations applied to various subsets of joints, i.e., left/right arm and left/right leg. We applied 10 values of perturbation strength

MSR Action 3D Dataset [12]				Workout SU-10 Gesture dataset [16]	
Acronym	Action type	Acronym	Action type	Acronym	Action type
AS2	high arm wave	AS2	two hand wave	A1	single leg balance with hip flexion
AS1	horizontal arm wave	AS2	side-boxing	A2	single leg balance trunk rotation
AS1	hammer	AS1	bend	A3	lateral stepping
AS2	hand catch	AS2, AS3	forward kick	B1	thoracic rotation bar on shoulder
AS1	forward punch	AS3	side kick	B2	hip adductor stretch 1
AS1,AS3	high throw	AS3	jogging	B3	hip adductor stretch 2
AS2	draw x	AS1, AS3	tennis swing	C1	dumbbell curl-to-press
AS2	draw tick	AS3	tennis serve	C2	freestanding squats
AS2	draw circle	AS3	golf swing	C3	transverse horizontal dumbbell punch
AS1	hand clap	AS1, AS3	pick & throw	C4	lateral trunk/oblique stretch

Table 1: The action types used in the experiments.

represented by standard deviations, $\sigma = 0.1 : 0.15 : 1.5$. Recall that the spine length of the skeleton was set to 1.

Sample perturbed skeleton sequences are illustrated in Figure 4. These perturbations are not constrained in that the kinematics of the arm or leg movements are not taken into account. However, for mild perturbations movies of the skeleton and Figure 4 quite plausible for the performance evaluation under limb joint uncertainties.

Parameter learning. The parameters, λ, W^d, are set on the training set. We have defined W^d as the mean value of triangle matching energies. It should be noted that we applied pose quantization only on the model graphs where we find setting $K = 512$ adequate in pose quantization.

4.2 Action Recognition Results

We used nearest neighbor classifier where the distance measure was the matching energy in Equation 1. The average performance is found to be 72.9% and 99.5% on MSR Action 3D Dataset and WorkoutSU Gesture dataset, respectively. These results are given in Table 2.

The performance when joint subsets were perturbed with additive Gaussian noise are presented in Table 3. We observe that the performance degrades gracefully with increasing limb noise.

Dataset	Number of action classes	Performance w/o Graph Mining	Performance w/ Graph Mining
MSR	20	71.8 ($L = 291$)	72.9 ($L = 271$)
WSU	10	94.9 ($L = 600$)	99.5 ($L = 507$)

Table 2: Recognition performances (%) on MSR: MSR Action 3D Dataset and WSU: WorkoutSU Gesture datasets. L is the number of model graphs used in the experiments.

Perturbation Type	Performance w/o Graph Mining	Performance w/ Graph Mining
Mild noise	93.4	98.8
Medium noise	86.4	96.2
Severe noise	75.7	87.8
No perturbation	94.9	99.5

Table 3: Recognition performances (%) under additive Gaussian noise. We set σ value to 0.1, 0.5 and 1 for mild, medium and severe noise cases, respectively.

Comparison with state-of-the-art methods. For completeness, we compared our algorithm with the state-of-the-art methods [20, 12, 16] which were described in Section 1.

In Table 4, we tabulated our results on MSR Action 3D Dataset [12]. The performance is computed using a cross-subject test setting where half of the subjects were used for training and testing was conducted on the unseen portion of the subjects. Li et al. [12] proposed to divide the dataset into three subsets in order to reduce the computational complexity during training. Each action set (AS) consists of eight action classes similar in context. We repeated the same experiment 100 times where, each time, we randomly selected five subjects for training and used the rest for testing. Finally, we computed the average recognition rate over all repetitions. As seen from Table 4, the proposed method performs better in AS1 and AS2, and has a competitive performance in AS3. Our proposed method is more successful in overall performance, especially in discriminating actions with similar movements.

In Table 5, we compared our results on Workout SU-10 Gesture dataset [16] with the method proposed in [16]. In Table 2, we have used 14 subjects as opposed to 12 subjects by Negin et al. [16]. For a fair comparison, we used the same experimental setup, namely, cross-subject test setting where we used the same six subjects for training and the same remaining six subjects for testing as in [16]. As seen, our proposed method with graph mining scheme performs better as compared with Negin et al. [16].

Action Set	Venkataraman et al. [20]	Li et al. [12]	Proposed Method w/o Graph Mining
AS1	77.5	72.9	**84.5**
AS2	63.1	71.9	**85.0**
AS3	**87.0**	79.2	72.2
Overall	75.9	74.7	**80.5**

Table 4: Recognition performances (%) for MSR Action 3D Dataset [12] in cross-subject test setting.

4.3 Quality Assessment Results

Classification results. In Table 6, we present the performance of our framework for discriminating correctly performed and wrongly performed sequences. We consider a set of 400 sequences. We obtained a set of wrongly performed sequences by distorting each sequence with $\sigma = 0.5, 1$, which results in 800 sequences in total. We trained a separate SVM classifier for each action class on the features as described in Section 3.1 and used a leave-one-subject-

Action	Negin et al. [16]	Proposed Method w/o Graph Mining	Proposed Method w/ Graph Mining
A1	100	96.7	100
A2	93.3	100	100
A3	98.3	100	100
B1	98.3	100	100
B2	96.6	100	100
B3	100	73.3	98.3
C1	100	100	100
C2	98.3	100	100
C3	100	100	100
C4	95.0	91.5	98.3
Overall	98.0	96.1	**99.6**

Table 5: Recognition performances (%) for Workout SU-10 Gesture dataset [16] in cross-subject test setting.

out test setting. Our average classification performance is found to be 86.6%.

	Correct. Perform.	Wrong. Perform.
Correct. Perform.	85.1	14.9
Wrong. Perform.	11.8	88.2

Table 6: Classification performance of the proposed framework: correctly performed sequences vs. wrongly performed sequences. Overall classification performance is 86.6%.

Regression analysis. The rendition of the action obviously deteriorates with the addition of the joint noise and we expect our action quality measure as in Section 3.2, that is, the calculated total distance between two matched sequences, to be proportional. We used a regression analysis between the noise variance and the matching energies to the model sequences. We considered 10 different σ values and four different body parts (left/right arm and left/right leg) and generated 3600 different sequences for action classes A1, A2 and A3. We matched each perturbed sequence with its original sequence and calculated the quality measure (QM) as in Section 3.2. As discussed before, in the absence of human evaluations, in order to find the relationship with the calculated distance and the perturbation variances, we applied Support Vector Regression (SVR) with polynomial kernel variety. Figure 5 verifies the relationship between the calculated quality measure and the perturbation variance.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an action recognition and action quality assessment method that is based on graph-based sequence alignment in conjunction with skeleton pose descriptors. The experimental results show its competitive action recognition capability and its utility as an action quality monitor on a continuous scale. In fact, since the Kinect system delivers the skeleton joint positions in real-time, our scheme can be used to give instantaneous feedback on the quality of action, whether over a portion of the action, a subset of joints, or over all the scene.

6. ACKNOWLEDGMENTS

This work was supported by Boğaziçi University Scientific Research Projects (BAP) (Project No. 6533). The work of Ceyhun Burak Akgül was supported by the research project ViPSafe supported partially by TÜBİTAK (agreement 109E134).

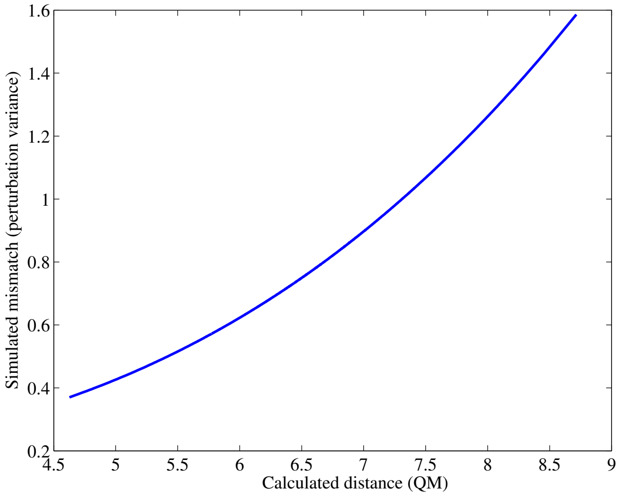


Figure 5: Calculated distance (QM) vs. simulated mismatch (perturbation variance).

7. REFERENCES

[1] D. Alexiadis, P. Daras, P. Kelly, N. E. O'Connor, T. Boubekeur, and M. B. Moussa. Evaluating a dancer's performance using kinect-based skeleton tracking. In *ACM Multimedia*, Scottsdale, AZ, USA, 2011.

[2] S. Bianco and F. Tisato. Karate moves recognition from skeletal motion. In *3D Image Processing (3DIP) and Applications*, San Francisco, CA, USA, 2012.

[3] O. Çeliktutan, C. Wolf, and B. Sankur. Fast exact matching and correspondence with hyper-graphs on spatio-temporal data. *LIRIS UMR 5205 CNRS/INSA de Lyon/Universit'e Claude Bernard Lyon 1/Universit'e Lumi'ere Lyon 2/Ecole Centrale de Lyon*, Report No. RR-LIRIS-2012-002, 2012.

[4] O. Çeliktutan, C. Wolf, B. Sankur, and E. Lombardi. Real-time exact graph matching with application in human action recognition. In *HBU*, 2012.

[5] S. Celebi, A. S. Aydın, T. T. Temiz, and T. Arici. Gesture recognition using skeleton data with weighted dynamic time warping. In *VISAPP*, Barcelona, Spain, 2013.

[6] P. Chua, D. Ventura, R. Crivella, T. Camill, B. Daly, N. Hu, J. Hodgins, R. Schaaf, and R. Pausch. Training for physical tasks in virtual environments: tai chi. In *Proc. of the IEEE Virtual Reality*, pages 87–94, 2003.

[7] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 18(3):265–298, 2004.

[8] C. Ellis, S. Masood, M. Tappen, J. L. Jr., and R. Sukthankar. Exploring the trade-off between accuracy and observational latency in action recognition. *International Journal of Computer Vision*, 101(3), 2013.

[9] S. Essid, D. Alexiadis, R. Tournemenne, M. Gowing, P. Kelly, D. Monaghan, P. Daras, A. Drémeau, and N. E. O'Connor. An advanced virtual dance performance evaluator. In *ICASSP*, 2012.

[10] W. Ilg, J. Mezger, and M. Giese. Estimation of skill levels in sports based on hierarchical spatio-temporal correspondences. In *DAGM Symposium*, 2003.

[11] C. Keskin, E. Berger, and L. Akarun. A unified framework for concurrent usage of hand gesture, shape and pose. In *CVPR 2012 Workshop on Gesture Recognition*, 2012.

[12] W. Q. Li, Z. Y. Zhang, and Z. C. Liu. Action recognition based on a bag of 3d points. In *CVPR4HB10*, pages 9–14, 2010.

[13] S. Michelet, K. Karp, E. Delaherche, C. Achard, and M. Chetouani. Automatic imitation assessment in interaction. In *HBU*, 2012.

[14] Microsoft. Introducing kinect for xbox 360. `http://www.xbox.com/en-US/kinect`. accessed at June, 2013.

[15] L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A. W. Vieira, and M. F. M. Campo. Real-time gesture recognition from depth data through key poses learning and decision forests. In *Sibgrapi*, pages 268–275, 2012.

[16] F. Negin, F. Özdemir, C. B. Akgül, K. A. Yüksel, and A. Erçil. A decision forest based feature selection framework for action recognition from rgb-depth cameras. In *ICIAR*, 2013.

[17] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. In *HAU3D*, 2012.

[18] M. Raptis, D. Kirovski, and H. Hoppe. Real-time classification of dance gestures from skeleton animation. In *ACM SIGGRAPH*, 2011.

[19] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, Colorado Springs, USA, 2011.

[20] V. Venkataraman, P. Turaga, N. Lehrer, M. Baran, T. Rikakis, and S. L. Wolf. Attractor-shape for dynamical analysis of human movement: Applications in stroke rehabilitation and action recognition. In *Human Activity Understanding from 3D data HAU3D'13*, 2012.

[21] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012.

[22] L. Wang, Y. Qiao, and X. Tang. Motionlets: Mid-level 3d parts for human motion recognition. In *CVPR*, 2013.

[23] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *The 2nd International Workshop on Human Activity Understanding from 3D Data (HAU3D) in conjunction with IEEE CVPR*, Providence, RI, 2012.

[24] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008.