Sequence Alignment for RGB-D and Motion Capture Skeletons

Xi Chen and Markus Koskela

Department of Information and Computer Science
Aalto University School of Science
P.O. Box 15400, FI-00076 Aalto, Finland
xi.chen@aalto.fi, markus.koskela@aalto.fi

Abstract. RGB-D skeletons are nowadays commonly used e.g. for gesture recognition, and so their accuracy and stability have significant influence on further processing. Skeletons obtained with motion capture are considerably more accurate and can be used to assess the quality of RGB-D skeleton extraction algorithms. In this paper, we record motion sequences with both a Kinect RGB-D sensor and a full motion capture system and align the generated skeletons by subsequence dynamic time warping with a varied step size. To evaluate the alignment, we propose two measures: the minimum overall distance between feature vectors and the distance of transformed skeletons. Experimental results show that our proposed method provides a better alignment between skeletons than the comparison methods. The proposed technique can also be used for content-based retrieval from large motion capture databases.

1 Introduction

Human motion analysis is often used in filmmaking, entertainment, sports and other applications. Traditionally, the human motion is recorded by a motion capture (aka mocap) system [7]. Such a system consists of multiple calibrated cameras in a dedicated space, and the recorded data needs to be processed by specific software. The expensive hardware and software make mocap data very costly. On the other hand, RGB cameras with depth sensors (RGB-D), such as the Microsoft Kinect, are nowadays widely used in computer vision due to their low cost and portability. The depth information provided by RGB-D cameras enriches the sensing of the environment, making the analysis of the 3D world considerably easier than with 2D images and e.g. stereo vision approaches.

Detecting the human figure from RGB-D data is relatively straightforward. In [12], a 2D edge detector and a 3D shape detector are used to detect the head and subsequently segment the human figure from the background. A detailed implementation is given in [5]. Several algorithms have been developed to generate the human skeleton based on the depth information only. The algorithm by Shotton et al [10] is implemented by Microsoft in their Kinect drivers and Xbox devices. In addition, some other algorithms for skeleton extraction are proposed in [9, 4]. The extracted skeleton is then often used in gesture recognition [13, 6].

The focus of this paper is to evaluate skeletons generated from Kinect depth images based on skeletons from a motion capture system that we consider here as the groundtruth. The proposed method can be used to compare skeletons generated by different algorithms, or to retrieve motion capture data from large databases by using a RGB-D skeleton sequence as the query. To evaluate the method, we record several human motion sequences simultaneously with both a Kinect sensor and a full motion capture system. Due to different frame rates, different starting and ending times in the recordings, and the possibility of missing frames, we need to align the skeletons on each frame from Kinect to mocap.

The distances between joints have been shown to be effective features in motion classification and retrieval [11]. In this paper, instead of direct pairwise distances between the joints, we use the distances between joints and the centroid of the skeleton, which significantly decreases the feature dimensionality and the complexity of the computations. We calculate a similarity matrix between the features from both systems, and then use subsequence dynamic time warping (SS-DTW) [1] with varied steps to find the minimum distances. We evaluate the similarity between the aligned sequences by visual observation and by quantitative measures. We also compare our approach with other methods for finding the alignment, out of which SS-DTW is shown to produce the best results.

2 Dynamic Time Warping

Dynamic time warping (DTW) is a well-known algorithm to measure the similarity and to find an alignment between two signal sequences [8]. Given two sequences, $P = p_1, p_2, ..., p_N$, and $Q = q_1, q_2, ..., q_M$, the dissimilarity $c(p_n, q_m)$ between elements from each sequence is defined as the distance between the elements

$$c(p_n, q_m) = f(p_n, q_m) \tag{1}$$

where f is a distance function. Commonly used distance functions include the Euclidean and cosine distances. By calculating the dissimilarity between each pair of elements from the sequences, a cost matrix $C \in \mathbb{R}^{N \times M}$ with $C(n, m) = c(p_n, q_m)$ can be obtained. DTW finds the alignment between P and Q with the minimum overall cost. The minimum cost alignment is also referred to as the warping path, as it indicates the indices of the matched elements in the sequences.

The warping path of the basic DTW has to fulfill two conditions:

1. Begin and end conditions: $p_1 \leftrightarrow q_1$ and $p_N \leftrightarrow q_M$
2. Step condition: If $p_n \leftrightarrow q_m$, then $p_n \leftrightarrow q_{m+1}$ or $p_{n+1} \leftrightarrow q_m$ or $p_{n+1} \leftrightarrow q_{m+1}$

That is, the first and last elements of P and Q have to align to each other, no elements in either sequence can be omitted, and no element can match to multiple elements. The most efficient solution to finding the warping path with the minimum overall cost is based on dynamic programming. Let us define the

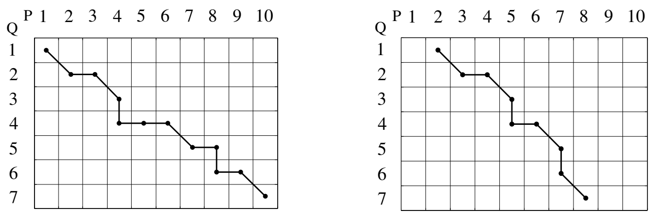


Fig. 1: Example warping paths of DTW (left) and subsequence DTW (right)

matrix D as the accumulated cost matrix, calculated as follows [2]:

$$D(1,1) = C(1,1) \tag{2a}$$
$$D(n,1) = D(n-1,1) + C(n,1) \tag{2b}$$
$$D(1,m) = D(1,m-1) + C(1,m) \tag{2c}$$
$$D(n,m) = \min(D(n-1,m) + D(n,m-1), D(n-1,m-1)) + C(n,m) \ . \tag{2d}$$

$D(n,m)$ is the minimum distance between the subsequences $P^n = p_1, p_2, ..., p_n$, and $Q^m = q_1, q_2, ..., q_m$, and so $D(N,M)$ is the minimum distance between the sequences P and Q. The optimal warping path can be found by tracing back the steps from the bottom-right corner of D to the top-left corner. If $D(n,m)$ is on the optimal path, then $\min(D(n-1,m-1), D(n-1,m), D(n,m-1))$ is the next step on the path. If there is no unique minimum, the element with the lexicographically smallest index is chosen. An example can be seen in Fig. 1.

2.1 Subsequence Dynamic Time Warping

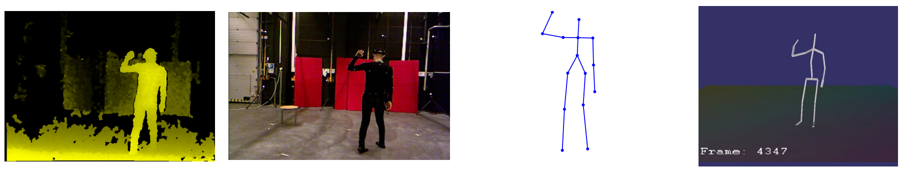
Subsequence DTW (SS-DTW) is a modification of DTW that does not fulfill the first condition of DTW, that is, it does not bound the beginning and end of the two sequences to each other. Instead, the longer sequence is assumed to contain a matching subsequence to the shorter sequence (see Fig. 1). SS-DTW can also be solved with the accumulated cost matrix. The calculation of the matrix is similar to the basic DTW except that Eq. (2c) is replaced by $D(1,m) = C(1,m)$.

The criterion to trace back to find the optimal warping path is also similar to DTW except that it does not start from $D(N,M)$. Instead, the starting point $D(N,m')$, which is also the end of the matching sequence, can be found by:

$$m' = \min_{\arg m} (D(N,m)) \quad , \ m = 1, 2, \dots, M \ . \tag{3}$$

3 Aligning RGB-D and Motion Capture Skeletons

In this project, an OptiTrack motion capture system and one Kinect device are used to collect skeletons of human actors. The motion capture system comprises



(a) Kinect depth image (b) Kinect video image (c) Skeleton from Kinect (d) Skeleton from mocap

Fig. 2: Data aquisition by mocap and Kinect

of 24 cameras set up to surround a large square space. The Kinect is placed outside the camera array. An actor performs in the square space and the performance is recorded by both systems. Part of the setup can be seen in the Kinect video frame in Fig. 2b. Fig. 2a shows the corresponding Kinect depth image. The extracted skeletons from Kinect and mocap can be seen in Figs. 2c–d. The used software for Kinect is based on the OpenNI framework[1] and the skeleton is extracted with the Natural Interaction Middleware (NiTE)[2] from PrimeSense.

During data acquisition, the actors performed several groups of actions, each lasting about two minutes. The start and stop of the recording on both systems was controlled manually. The motion capture system was always started earlier and stopped later than the Kinect recording.

3.1 Feature Extraction

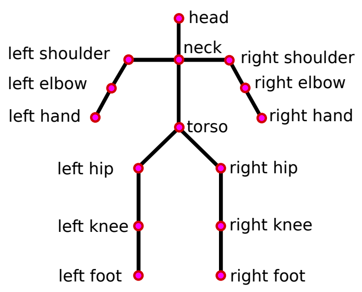
The used mocap system provides 22 joints in the skeleton, while the algorithm we used with Kinect provides 15 joints (see Fig. 3). Compared to Kinect, the mocap skeleton provides additionally the ends of hands, feet and head, and the center hip. Kinect uses one joint (*torso*) to represent the body, whereas mocap uses *ab* and *chest*, neither of which directly match the Kinect *torso*. Therefore, in order to be able to extract the same features, we first need to project the skeletons to the same structure. The mocap skeleton can be simplified by deleting the extra joints. Moreover, the Kinect *torso* is more similar to *ab* than *chest*, so we delete the *chest* joint. Now the skeletons have the same number of joints, and with the exception of *torso* and *ab*, the joints are conceptually matching each other.

We represent the 3D coordinates of each joint[3] as $\mathbf{j}_i = (x_i, y_i, z_i)$. As the *torso* in Kinect and *ab* in mocap do not match, we calculate the center of the body as the centroid of the neck and left and right hips in both systems,

$$\mathbf{c} = \frac{1}{3}(\mathbf{j}_{neck} + \mathbf{j}_{lhip} + \mathbf{j}_{rhip}) \ . \tag{4}$$

For each frame, we define a 15-dimensional feature vector $\mathbf{f} = [d_1 \ d_2 \ \dots \ d_{15}]^T$ whose each element is the Euclidean distance between a specific joint and the

[1] http://www.openni.org/ [2] http://www.primesense.com/Nite/ [3] We use the superscripts as in $\mathbf{j}^k$ and $\mathbf{j}^m$ to distinguish between Kinect and mocap when necessary.



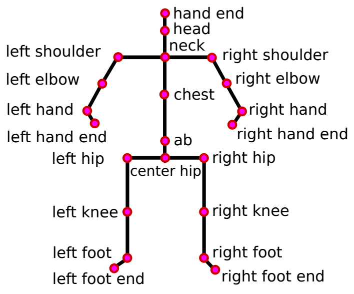


Fig. 3: The skeleton joints from Kinect (left) and motion capture (right)

centroid,

$$d_i = \|\mathbf{j}_i - \mathbf{c}\| \quad , \; 1 \le i \le 15 \; . \tag{5}$$

Due to the varying size of actors, the distances can be different even for the same gesture. Therefore, the feature vector is normalized with the L_1 norm

$$\bar{\mathbf{f}} = \frac{\mathbf{f}}{\|\mathbf{f}\|_1} \; . \tag{6}$$

The elements of the cost matrix are then calculated as $c(n, m) = \|\bar{\mathbf{f}}_n - \bar{\mathbf{f}}_m\|$.

It should be noted that the selection of used features in our method is not restricted. Alternatively, we could have used e.g. the higher-dimensional pairwise distances between the joints as in [11].

3.2 Alignment of the Skeleton Sequences

The frame rate of Kinect is 30 fps, but in practice some frames are missed during the recording and without time stamps it is difficult to know which ones these are. The mocap frame rate is very accurately 100 fps. Since the start and stop of the recording are controlled manually, the recordings of mocap and Kinect are not aligned either at the beginning or at the end. Therefore, we use SS-DTW to find the corresponding frames between the Kinect and mocap skeletons.

In the optimal warping path of the basic DTW, the step condition ensures that each element in both sequences has a match. However, the frame rate of motion capture is more than thrice that of Kinect, that is, between the consecutive Kinect frames there are at least three frames from mocap. Thus we have to modify the step size of DTW to fulfill our requirements. Let us assume that the Kinect frame f_i^k matches with the mocap frame f_j^m (see Fig. 4). Theoretically, the next matching mocap frame lies between the 3rd and 4th frames. We therefore try to match the mocap frames f_{j+3}^m to f_{j+5}^m to the Kinect frame f_{i+1}^k. In addition, some frames are missing in the Kinect recording, so the next frame in the Kinect sequence might actually be the second or even the third frame, so we extend the possible matches to the frames f_{j+6}^m to f_{j+11}^m. Thus, in the step condition of DTW, the next possible matching ranges from the 3rd to the 11th

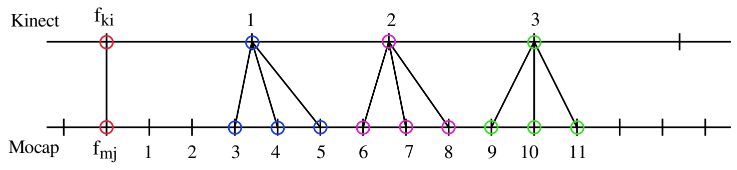


Fig. 4: The matching pattern between Kinect and motion capture skeleton

frame in the mocap sequence. The accumulated cost matrix of SS-DTW with the new step condition can be calculated by replacing Eq. (2d) with

$$D(n,m) = \min_{3 \leq i \leq 11} \left(D(n-1, m-i)\right) + C(n,m) \ . \tag{7}$$

4 Evaluation Methods

4.1 Visual Observation

After aligning the skeletons, it is useful to be able to observe the alignment. We wrote a simple visualization software that is able to play the skeletons in the same framework. It can be set to different frame rates to play the skeletons and to show the related information with each frame.

4.2 Minimum Overall Distance between Feature Vectors

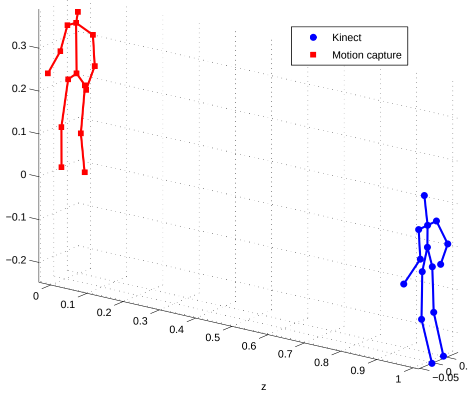
We also evaluate the skeleton alignment quantitatively. With DTW, the alignment is obtained by finding the minimum value of the accumulated cost matrix, which can also be used as a criterion to evaluate the alignment. For SS-DTW, this value is accessible during the calculation of the accumulated cost matrix

$$E_{dist} = \min_m (D(N,m)) \quad , \ m = 1, \ldots, M \ . \tag{8}$$

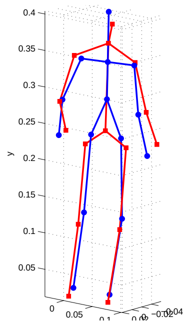
Even if the aligned sequences were not obtained by DTW, they can still be evaluated by the same criterion. The procedure is as follows:

1. Calculate the features of each frame in the Kinect sequences (Eqs. 4–6).
2. Simplify the mocap sequence to the same skeleton structure as Kinect.
3. Calculate the features of each frame in the simplified mocap sequences.
4. Calculate the Euclidean distance of the feature vectors between each corresponding frames, and sum up all the distances in the sequence

$$E_{dist} = \sum_{i=1}^{N} \|\mathbf{f}_i^k - \mathbf{f}_i^m\| \ . \tag{9}$$



(a) Skeletons in normalized coordinates



(b) Skeletons after transformation

Fig. 5: Matched skeletons in different coordinate systems

4.3 Skeleton Transformation

As the above evaluation method uses the same criterion as obtaining the alignment we develop another evaluation method that uses a different criterion. In this method, we try to overlap the two skeletons and calculate the distances between the corresponding joints. During data recording, the coordinate systems of mocap and Kinect are not aligned and so the skeletons are in different coordinate systems. First, we normalize the scales of the skeletons by dividing the coordinates of each joint by the total distance of the connected bones. In Fig. 5a, the matched skeletons are drawn with their normalized coordinates.

Next, the two coordinate systems should be aligned. The Kinect skeleton is not very accurate especially for hands and feet, so the transformation can be largely influenced by outliers within these joints. The joints of the main body are, however, relative stable, so the skeletons can be transformed based on the body joints. We use the neck and hips from the two skeletons for aligning so the sum of the distances between these corresponding joints is minimized,

$$E(\mathbf{R},\mathbf{t}) = \sum_{i\in\{neck,lhip,rhip\}} \left(\|\bar{\mathbf{j}}_i^m - (\mathbf{R}\bar{\mathbf{j}}_i^k + \mathbf{t})\|\right) , \tag{10}$$

where $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ a translation vector for the Kinect skeleton. Eq. (10) can be minimized by the Horn's method [3]. We compute $\mathbf{R}$ and $\mathbf{t}$ for each frame and then transform the joints of the Kinect skeleton to the new coordinates. Fig. 5b shows an example Kinect skeleton aligned to a mocap skeleton. The evaluation measure is the sum of all Euclidean distances between the corresponding joints in each frame of the sequences,

$$E_{trans} = \sum_{j=1}^{N} \sum_{i=1}^{15} \|\bar{\mathbf{j}}_{i,j}^m - (\mathbf{R}_j\bar{\mathbf{j}}_{i,j}^k + \mathbf{t}_j)\| . \tag{11}$$

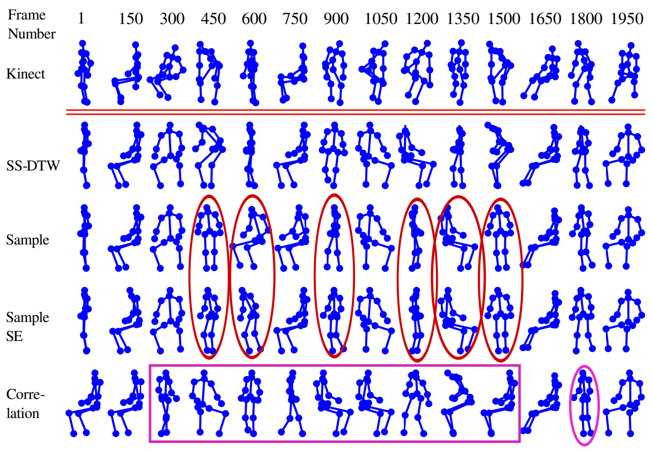


Fig. 6: Skeletons aligned with different methods

5 Experiments

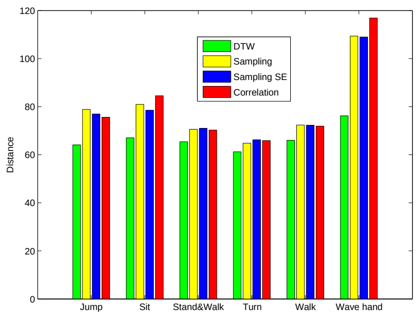
During data acquisition, we recorded six RGB-D videos and the corresponding mocap data of human actors. The videos are named according to their main type of motion as *jump*, *sit*, *stand & walk*, *turn*, *walk* and *wave hand*. We aligned the skeletons with SS-DTW and with the following three comparison methods:

Direct sampling. The direct sampling method samples the mocap skeleton data with the sample rate of $\frac{M}{N}$, where M and N are the total numbers of frames in the corresponding mocap and Kinect sequences. The sampled mocap skeleton thus has the same number of frames as Kinect, and we assume implicitly that the recordings have been started and stopped simultaneously.

Sampling with DTW start and end. In this method, we use the start and end points detected by the SS-DTW for sampling the mocap sequence. The sample rate of the mocap data is thus $\frac{E-S}{N}$, where E and S are the frame indices of the SS-DTW end and start points.

Normalized cross-correlation. Theoretically, we can match the sequences by sampling the mocap data with the sample rate of $\frac{10}{3}$. We start by aligning the start points of the Kinect and mocap sequences, and calculate the sum of normalized cross-correlation between the aligned feature vectors. We then slide the Kinect sequence by one frame, calculate again the sum of normalized cross-correlation, and repeat this until the ends of the two sequences align. The maximum value then corresponds to the best alignment between the sequences.

The aligned skeletons can be first observed visually. Fig. 6 shows some frames from the *sit* sequence aligned with these four methods. We can see that the SS-DTW alignment is visually very close to the Kinect sequence. The skeletons from direct sampling and sampling with DTW start and end are very similar, which means that the start and end found by DTW are close to the beginning



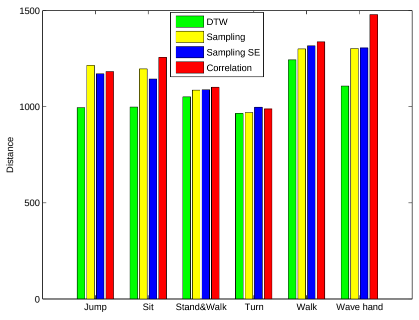


Fig. 7: The distances E_{dist} (left) and E_{trans} (right) for each recording

and end of the original recording. However, some frames are not matching well with Kinect (marked by ellipses). Correlation gives many ill-matching skeletons (marked by a rectangle), as the missing frames in Kinect make it hard to find a long well-matching subsequence from mocap.

The aligned sequences are also evaluated by measuring the minimum overall distance between the feature vectors, E_{dist} (Sec. 4.2) and the skeleton transformation, E_{trans} (Sec. 4.3). The results are shown in Fig. 7. Accoding to both measures, for each recording the distances between Kinect and the SS-DTW alignment are smaller than with the other methods. Sampling with DTW start and end points has notably lower performance than DTW, which shows that the missing frames make the alignment nonlinear. Some distances from the cross-correlation method are very high, partly due to it also ignoring the missing Kinect frames. When the number of mocap frames is fixed, the matching time with DTW is almost linear to the number of frames in the Kinect sequence. With our Matlab implementation using Intel i5 CPU at 3.3 GHz, aligning a one-minute Kinect sequence with two minutes of mocap takes about 50 seconds.

6 Conclusion and Future Work

In this paper, we align skeleton sequences from Kinect and a motion capture system by feature extraction and subsequence dynamic time warping. The aligned sequences show a good visual alignment. We also propose two quantitative methods to numerically evaluate the alignment: the minimum overall distance between feature vectors and the distances of transformed skeletons. Both evaluation methods show that subsequence dynamic time warping is better than comparison methods based on sampling and cross-correlation.

The proposed technique to align the sequences can also be used for content-based retrieval of similar motions from large motion capture databases. Instead of textual or faceted search or requiring the user to have existing motion capture data for similarity search, the user can produce RGB-D motion sequences on-line and then retrieve similar motion sequences from the database, in order to be able to reuse the expensive motion capture data. The proposed evaluation methods can also be used for evaluating the accuracy of different RGB-D skeletons.

In the future, we will build a complete system for content-based retrieval from a motion capture database by Kinect skeletons with easy access and low computation requirements. Additionally, we continue to work on our alignment method to take the confidence values of the Kinect joints into account so that the low confidence joints will not negatively influence the alignment.

Acknowledgments. The authors would like to thank Klaus Förger at Department of Media Technology, Aalto University, for helping to collect the mocap data. This work was funded by Aalto MIDE programme (project UI-ART) and grants Multimodally grounded language technology (254104) and Finnish CoE in Computational Inference Research COIN (251170) of the Academy of Finland.

References

1. Bautista, M., Hernández-Vela, A., Ponce, V., Perez-Sala, X., Baró, X., Pujol, O., Angulo, C., Escalera, S.: Probability-based dynamic time warping for gesture recognition on RGB-D data. In: International Workshop on Depth Image Analysis. Tsukuba Science City, Japan (2012)
2. Deng, L., Leung, H., Gu, N., Yang, Y.: Automated recognition of sequential patterns in captured motion streams. In: Proc. 11th International conference on Web-age information management (WAIM'10). pp. 250–261 (2010)
3. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A 4(4), 629–642 (1987)
4. Kar, A.: Skeletal tracking using Microsoft Kinect. Methodology 1, 1–11 (2010)
5. Krishnamurthy, S.N.: Human Detection and Extraction using Kinect Depth Images. Master's thesis, Bournemouth University, the United Kingdom (2011)
6. Lai, K., Konrad, J., Ishwar, P.: A gesture-driven computer interface using Kinect. In: Proc. Image Analysis and Interpretation (SSIAI 2012). pp. 185–188 (2012)
7. Menache, A.: Understanding motion capture for computer animation and video games. Morgan Kaufmann Pub (2000)
8. Müller, M.: Information Retrieval for Music and Motion. Springer (2007)
9. Shen, W., Xiao, S., Jiang, N., Liu, W.: Unsupervised human skeleton extraction from Kinect depth images. In: Proc. 4th International Conference on Internet Multimedia Computing and Service (ICIMCS '12). pp. 66–69. New York, USA (2012)
10. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: Proc. Computer Vision and Pattern Recognition (June 2011)
11. Vieira, A., Lewiner, T., Schwartz, W., Campos, M.: Distance matrices as invariant features for classifying MoCap data. In: 21st International Conference on Pattern Recognition (ICPR). Tsukuba, Japan (2012)
12. Xia, L., Chen, C.C., Aggarwal, J.K.: Human detection using depth information by Kinect. In: Workshop on Human Activity Understanding from 3D Data in conjunction with CVPR (HAU3D). Colorado Springs, USA (2011)
13. Xia, L., Chen, C.C., Aggarwal, J.K.: View invariant human action recognition using histograms of 3D joints. In: CVPR Workshops. pp. 20–27. IEEE (2012)