Skeleton Extraction of Dance Sequences from 3D Points using Convolutional Neural Networks based on a New Developed C3D Visualization Interface

Ioannis Kavouras, Eftychios Protopapadakis, Anastasios Doulamis, and Nikolaos Doulamis

National Technical University of Athens, 15773, Zografou, Athens, Greece,
johncrabs1995@gmail.com, eftprot@mail.ntua.gr,
adoulam@cs.ntua.gr, ndoulam@cs.ntua.gr

Abstract. A combined approach, involving 3D spatial datasets, noise removal prepossessing and deep learning regression approaches for the estimation of rough skeleton data, is presented in this paper. The application scenario involved data sequences from Greek traditional dances. In particular, a visualization application interface was developed allowing the user to load the C3D sequences, edit the data and remove possible noise. The interface was developed using the OpenGL language and is able to parse aby C3D format file. The interface is supported by several functinalities such as a pre-processing of the 3D point data and noise removal of 3D points that fall apart from the human skeleton.
The main research innovation of this paper is the use of a deep machine learning framework through which human skeleton can be extracted. The points are selected on the use of a Convolutional Neural Network (CNN) model. Experimental results on real-life dances being captured by the Vicon motion capturing system are presented to show the great performance of the proposed scheme.

1 Introduction

Intangible cultural heritage (ICH) is a major element of peoples identities and its preservation should be pursued along with the safeguarding of tangible cultural heritage. In this context, traditional folk dances are directly connected to local culture and identity [1]. For this reason, recently, European Union has been funded research projects for preserving, documenting and analyzing intangible cultural heritage aspects and folkoric performing arts [2], [3]. The current technological achievements, in the area of software and hardware engineering, have emerged the use of efficient 3D motion capturing interfaces for digitizing human kinesiology. Examples include the low cost Kinect sensor [4] and the most professional Vicon motion capturing interface [5]. However, these motion capturing interfaces mainly focus on the mechanisms for acquiring raw human data in the form of 3D point clouds, instead of an intelligent 3D oriented processing and visualization methodology.

Towards this direction, methods for estimating human 3D skeleton from the acquired 3D point clouds have been performed. For this reason, the single depth acquired image, taken for example from the Kinect sensor, are processed in order to extract 3D human skeleton data in a real time constraint framework [6]. In other words, the unstructured 3D point clouds are processed, using machine learning paradigms, in order to derive a compact representation regarding human kinesiology. Using this information provided by the Kinect sensors, research methods have been proposed in the literature for analyzing a choreographic pattern. In this context, the work of [7] exploits multiple Kinect sensors for dance analysis. Additionally, the work of [8] classifies pop dances based on skeleton information provided by a Kinect sensor. Other methods exploit a set of cameras for digitizing human activities such as the work of [9] regarding Japanesse dances or the work of [10] for cypriot performance art.

In the area of choreographic data analysis and processing dance summarization methods are recently investigated by the research community. Charasteristic examples are the works of [11] and [12] where a k-means algorithm is proposed for the estimation of the most characteristic choreographic patterns (as in [11]) or an hierarhical spatial-temporal algorithm exploiting principles of Sparse Modeling Representative Selection (SMRS)[13] (as in [12]). Additionally, the work of [12] deals also with kinesiology data structures in order to get the most salient (key) human movements.

Recently, deep learning architectures have been proposed for human action recognition exploiting skeleton information [14]. More specifically, Convolutional Neural Networks (CNNs) Models have been proposed in [15] to classify different type of dances, while in [16] a deep learning pipeline is introduced for dance style classification. Other approaches exploit Grassmannian point structures instead of deep learning towards the classification of time varying data [17].

In this paper, a method for better organization, structuring and visualization of 3D choreographic data is proposed. In particular, as far as choreographic data structuring is concerned, we have investigated the use of deep machine learning in extracting dancer's 3D skeletons from raw point clouds. For this purpose a Convolutional Neural Network model is introduced [18], with the aim of transforming 3D information, fed as input to the neural network, to discrete 3D human joints, produced as output of the network. In this way, we derive a more semantic description and organization of the raw 3D points, captured from depth sensor interfaces, the Kinect in our case. In addition, regarding the visualization of the choreographic movements, we describe an efficient interface developed in openGL framework [19] for editing and manipulating choreographies. The developed editing interface complies with the Coordinated 3 Dimensional (C3D) file format [20], which is a data structure representation for representing 3D moving objects [20]. The C3D data data format has been deployed in various application scenarios, such as visualization of gaits [21] or for bio-mechanical data [22].The developed application is a first step for creating a full featured package for visualizing and editing C3D files with emphasis on dance analysis.

This paper is organized as follows: Section 2 presents the new developed OpenGL based visualization interface based on C3D data. The interface is enriched with some editing and noise removal functionalities. Section 3 presents the skeleton extrcation from the 3D points using deep learning Convolutional Neural Networks. Experimental results are presented in Section 4 while Section 5 concludes the paper.

2 Functionalities of the Editing Interface for Choreographic Representation

In this section, we describe the key functionalities of the proposed openGL interaface for choregraphic representation, analysis and editing. The proposed methodology imports data in C3D format and includes methods for i) noise removal, ii) unsupervised clustering and supervised classification, iii) extracting human 3D skeletons (using a CNN model) and finally iv) exporting capabilities to CSV files. Skeleton extraction includes the estimation of human body parts/joints, including head, left & right palm, upper & lower torso, knees, ankles and shoulders.

2.1 The C3D Format

The C3D data file format is originally developed by AMASS photogrammetric software system during 1986-1987. The C3D format provides a convenient and efficient means for storing 3D coordinate and analog data, together with all associated parameters[20]. C3D files are binary data types, consisting of three section blocks, i.e. header, parameters, and data. A short description for each of the blocks is provided bellow.

The Header section covers the first 512-bytes of the file. In this section, the parameter values are saved which are needed for reading the data section followed. It is not recommended to use these values from the header section, because the rational of a C3D file format is the connection between the data and their meanings (parameters). However, for simple projects, where only data are needed, it is easier to program a code, which reads the data without reading the whole parameter section.

In the parameter section, metadata information is saved. Without the parameter section, the C3D file would be just another simple numeric file format, like CSV. The parameter section is divided into the group part and the parameter part. The latter is laid inside each group. In this way, we can have more parameters with the same name, each pointing to a different description. For example, the SCALE parameter in the POINT and the ANALOG group with a different meaning in each group.

Data section includes all numeric values of the data. The numeric values are saved in frames and each frame is described by spatial and analog information.

2.2 The Developed User Interface

The application's interface developed consists of a menu toolbar, the visualization widget and the cluster list widget as shown in Fig. 1. Especially the menu toolbar subdivided into the following categories:

1. **File:** It includes commands for import and export data files. The application can only import C3D file and can export TXT and CSV files.
2. **Edit:** It includes commands for simple editing the visualization scene, such us metric scaling, and clustering editing.
3. **View:** User can use the commands of this category to change the viewpoint of the scene.
4. **Tools:** It includes all the processing commands for noise removal. A k-means clustering is exploited towards this.
5. **About:** It includes information about the application.

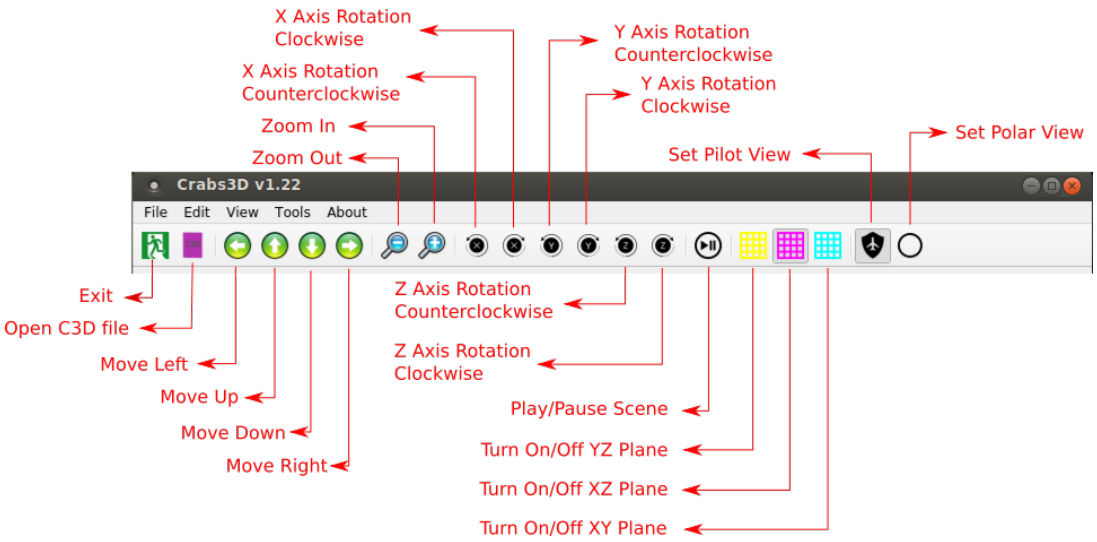

Fig. 1. The Menu developed in our case analyzing, presenting and visualizing 3D points data of a dancer.

2.3 Data Editing, Preprocessing & Noise Removal

The 3D point data are often noisy. This is due to the errors of the motion capturing procedure or due to inferent low resolution of the depth sensors (see Fig. 2). In general, three types of noise is observed: i) the cloud points are far away from the actual human body, ii) 3D points of the human body that suddenly differentiate from the body cloud and iii) body points that suddenly gather around point of origin.

As far as the cloud points that are located far away from the actual body are concerned, the developed functionality of our interface calculates the Euclidean distance between the first frame and the other ones. If this distance is lower than a given threshold value, set by the user, then this point is recognized as immobilized. The second parameter of this algorithm is the frame rate or the tolerance. If the number of frames, where a point has been declared as immobilized, is higher than the frame rate, then this point is set invisible.

The 3D points being located far away from the actual human body are points, which behave either as object points at same frames or as noise points on other frames. To filter this kind of noise, our interface exploits the k-nearest neighbor algorithm. The user set a maximum radius and a tolerance value so that the k neighbors are specified. Then, for each point, the distance to the other points is calculated. If this

point has a neighbor value at least equal to the tolerance value for all frames, then this point is declared as an object point. In any other cases, this point is considered as noise and is set to be invisible in the display mode of our interface. The main drawback of this algorithm occurs when the noise points are too close. Then the application declares them as object points since they cannot be distinguished from the other human points.

To correct the body points that suddenly gather around point of origin, we need to create a fix cloud command. This command uses the tendency of the point and predicts a probability location for the frame, considering this point as noise. This command rewrites the coordinates of a point and the result is irreversible.

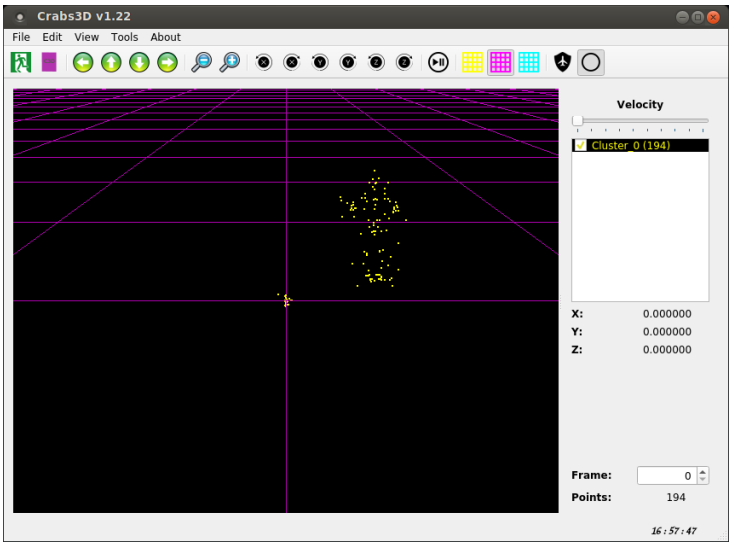

Fig. 2. Perspective view of a C3D file. In this figure, we depict the noise parts

3 Deep Learning for Human Skeleton Extraction

The goal of this section is to develop a new deep learning based framework for extracting human skeletons from the 3D data. The algorithm receives as inputs the 3D points of the C3D format as well as a pre-processed data filtering on the 3D points so as to remove the noise. The main steps of the algorithm are: (i) the estimation of a rough human skeleton through the application of a clustering algorithm and (ii) the application of a deep machine learning scheme for implementing the final refinement of the data.

3.1 Rough Skeleton Creation

The rough estimation of the human skeleton is done through the use of a clustering scheme. In this paper, the simple k-means algorithm is implemented. The goal of the k-means is to categorize the data into k clusters with respect to their position and orientation in the 3D space. The main disadvantage of the k-means algorithm is that fact that it is sensitive to the selection of the initial values of the cluster centers. Upon a different initial cluster center selection, different clusters are estimated and the categorization of the data may be different. Anther drawback point of the k-means clustering scheme is that the number of clusters should be a priori known.To address these limitations in this paper, and as functionality of our developed interface, a modification of the k-means is adopted called k-means++ [23]. The k-means++ can work more robustly with respect to the number of clusters than the conventional k-means algorithm.

3.2 Deep Learning Models-The Convolutional Neural Network Scheme for Skeleton Extraction

Having derived a rough human skeleton through the clustering approach proposed in the aforementioned section, we proceed with the final refinement of the model. To do this, we apply a deep machine learning scheme via the use of a Convolutional Neural Network (CNN) model. A CNN typically comprises three main types of layers, namely, (i) Convolutional Layers, (ii) ReLU Layers, and (iii) Fully Connected Layers. Each layer type plays a different role in the analysis f the data. Figure 3 shows a CNN topological architecture. This topology is useful for object detection from imaginary data. In particular, every layer of the CNN transforms, via convolutions, the inputs into more semantic meanings (descriptions) and eventually leads to the final fully connected layers. This layer performs the actual classification of the skeleton data.

In contrast to conventional neural networks structures, CNNs work well with images or even 3D volumes. This allows us to encode certain properties of the architecture and therefore a CNN can be used as a feature extractor module so that the most suitable features of the input data can be exploited.

In this paper, a modification of the classical CNN structure is proposed using the 3D scheme of [24]. The goal is to exploit a 3D CNN structure which will be useful for detecting objects. This structure is very important in our case since 3D points are reeceived as inputs.

The Convolutional Layer In the convolutional layers, a CNN utilizes various kernels to convolve the whole image as well as the intermediate feature maps, generating various feature maps. Because of the advantages of the convolution operation, several works (e.g., [25] [26]) have proposed it as a substitute for fully connected layers with a view to attaining faster learning times.

ReLU Layers . The units of this layer are called Rectified Linear Units (ReLU). These units apply non-saturating activation function of the form,

$$\mathrm{f(x)} = \max(0, \mathrm{x}) \tag{1}$$

The goal of this layer is to increase the nonlinear properties of the overall network without affecting the receptive fields of the convolution layer [27].

The Fully Connected Layer Following several convolutional and pooling layers, the high-level reasoning in the neural network is performed through the fully connected last layer. The neurons in the fully connected layer have full connections to all nodes of the previous layer, as name of the layer implies. The neurons' output, in this layer, can hence be computed as a matrix multiplication followed by a bias offset. Fully connected layers eventually convert the 2D/3D feature maps into a 1D feature vector. The derived vector could either be fed forward into a certain number of categories for classification [28] or could be considered as a feature vector for further processing [29].

4 Experimental Setup

The application has been developed using C++ programming language. For developing the interface the QtCreator and the OpenGL are adopted [1].

4.1 Dataset Description

For the purpose of this paper, two C3D files are used. These files have been recorded using the Vicon motion Camera System and describe the Greek traditional dance Syrtos. In each dataset a dance is executed by different dancers, one male and one female. The data of the female dancer are used for

[1] The developed user interface can be found in `https://github.com/JohnCrabs/Crabs3Dv122`

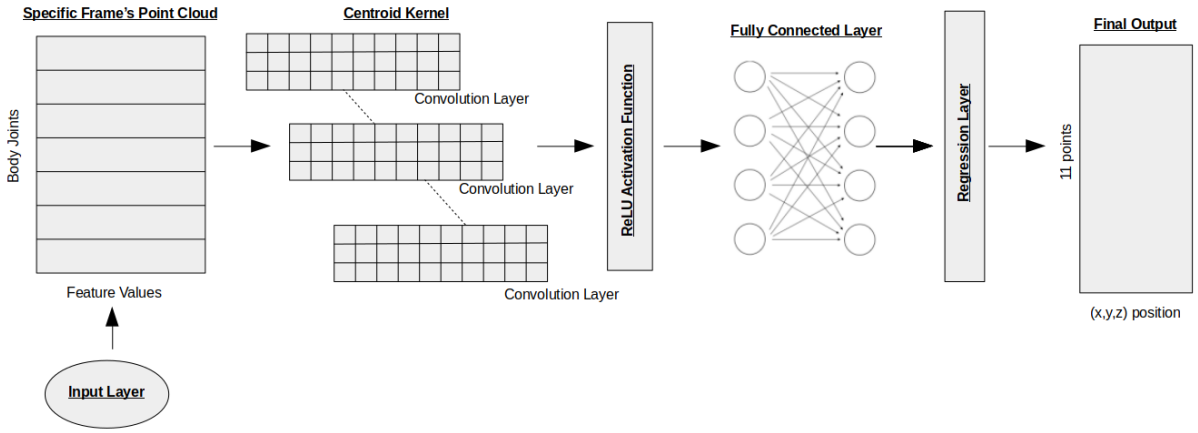

Fig. 3. The topology of the adopted Convolutional Neural Network for human skeleton extraction.

training inputs, while the data of the male dancer are used for testing. Most specifically, the training dataset consists of 1000 frames, each of which compose of 50 3D points. On the other hand, the test data set consists again of 1000 frames, in each of which we have extracted eleven rough 3D skeleton points using the clustering algorithm described in Section 3.1. We have also created another test set consisting of 500 image samples of a different dancer.

4.2 Performance Metrics

For the estimation of the accuracy of convolution neural network, the following four objective criteria have been taken into account.

Minimum Deviation (MIN) The minimum deviation is the lowest absolute value of the difference between the estimated values and the real measurements.

Maximum Deviation (MAX) The maximum deviation is the higher absolute value of the difference between the estimated values and the real measurements.

Mean Absolute Estimation (MAE) The mean absolute error is defined as:

$$\mathrm{MAE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} |\mathrm{x_i} - \mathrm{m(X)}| \tag{2}$$

where $\mathrm{x_i}$ is the estimated value and $\mathrm{m(X)}$ the respective central value.

Root Mean Square Estimation (RMSE) The root mean square error is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{x_i} - \mathrm{m(X)})^2}{\mathrm{n}}} \tag{3}$$

where again $\mathrm{x_i}$ is the estimated value and $\mathrm{m(X)}$ the respective mean operator.

4.3 Experimental Results

In order to evaluate the performance of the CNN model in estimating human skeleton from 3D points, the aforementioned objective criteria are used. Table 1 shows the results regarding the eleven targeted skeleton points for different values of training epochs of the CNN model. The results are depicted for the two considered data sets; the one of the 1000 image frames and the other of the 500 image samples. Both test sets comprise different dancers against the one of the training set. In this table, the error over the three axes (i.e., xyz) are depicted.

Figure 4 shows the RMSE metric for the eleven skeleton human points over all epochs that the CNN model has been trained on. The figure considers the first test set of the 1000 image samples. As is observed, the majority of the human skeleton joints are well identified by the CNN model.

Figure 5 depicts performance evaluation between the two test sets; the one of 500 frames and the one of 1000 frames, in case that 2000 epochs are used. epochs. As is observed, the test set of 500 image samples are slightly better than that the one of the 1000 image samples.

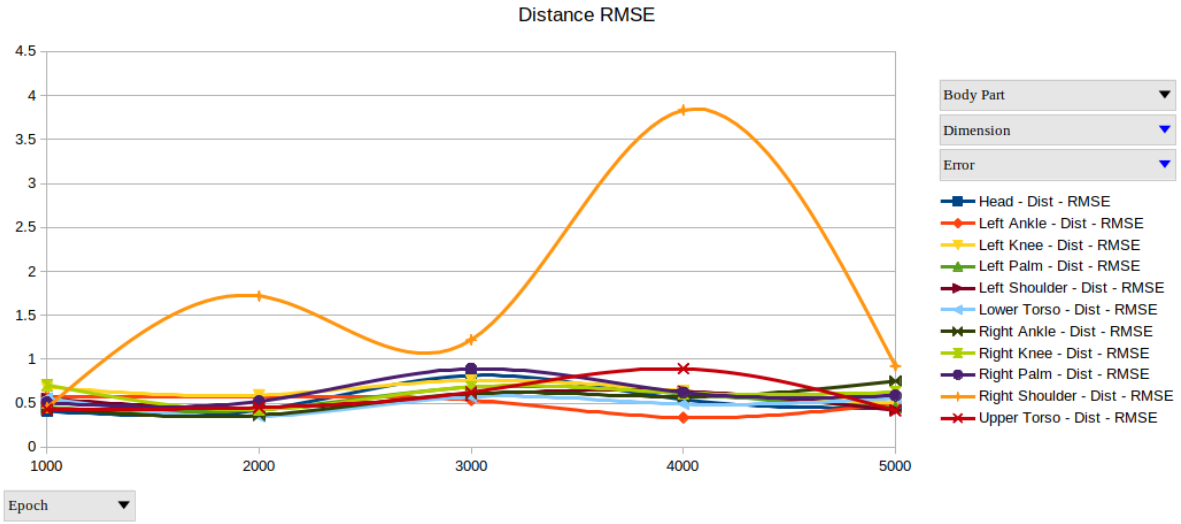

Fig. 4. Test results for all epochs.

Table 1. Convolutional Neural Network Errors.

Training Test for 1000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0903	0.4682	0.3189	0.3325	0.2612	0.4391	0.3477	0.3500	0.0003	0.2938	0.1346	0.1576	0.4906	0.5078
Left Palm	0.1120	0.5133	0.3088	0.3216	0.1170	0.4458	0.2655	0.2742	0.0004	0.2866	0.1372	0.1557	0.4297	0.4504
Head	0.1944	0.5034	0.3793	0.3876	0.0002	0.1252	0.0404	0.0486	0.0016	0.2589	0.1110	0.1264	0.3973	0.4106
Upper Torso	0.1873	0.4737	0.3643	0.3706	0.0001	0.2882	0.0525	0.0601	0.0004	0.3687	0.1807	0.2064	0.4100	0.4284
Lower Torso	0.3648	0.6569	0.5204	0.5261	0.0364	0.2058	0.1256	0.1304	0.0002	0.2137	0.1127	0.1247	0.5471	0.5562
Right Shoulder	0.0008	0.5644	0.2305	0.2668	0.0001	0.2544	0.0689	0.0840	0.0003	1.0740	0.2645	0.3513	0.3575	0.4491
Left Shoulder	0.1680	0.8349	0.4355	0.4666	0.0001	0.2746	0.0653	0.1032	0.0002	0.4657	0.2459	0.2746	0.5044	0.5512
Right Knee	0.3059	0.8789	0.6361	0.6554	0.0557	0.2345	0.1435	0.1486	0.0016	0.4408	0.1913	0.2189	0.6796	0.7068
Left Knee	0.3693	0.7366	0.5597	0.5675	0.2087	0.5211	0.3584	0.3657	0.0001	0.1639	0.0768	0.0894	0.6690	0.6810
Right Ankle	0.0001	0.1639	0.0768	0.0894	0.1292	0.6807	0.4112	0.4343	0.0001	0.0900	0.0283	0.0361	0.4193	0.4449
Left Ankle	0.2071	0.6947	0.4906	0.5016	0.0682	0.2191	0.1330	0.1371	0.1329	0.2071	0.2346	0.2439	0.5598	0.5744
Training Test for 2000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0002	0.4353	0.1787	0.2280	0.2268	0.5457	0.3758	0.3834	0.0019	0.4412	0.2509	0.2694	0.4859	0.5211
Left Palm	0.0001	0.3698	0.1916	0.2167	0.2224	0.5088	0.3353	0.3420	0.0002	0.2595	0.1159	0.1354	0.4032	0.4269
Head	0.0005	0.5309	0.3000	0.3253	0.1518	0.4070	0.2521	0.2596	0.0001	0.1751	0.0551	0.0732	0.3957	0.4226
Upper Torso	0.0002	0.4093	0.1831	0.2205	0.0065	0.3475	0.2300	0.2359	0.0006	0.6280	0.2621	0.3166	0.3939	0.4522
Lower Torso	0.0003	0.3279	0.1812	0.1955	0.1466	0.3421	0.2185	0.2218	0.0003	0.4234	0.1498	0.1879	0.3210	0.3503
Right Shoulder	0.0015	2.6338	0.8822	1.1857	0.0032	0.6290	0.4075	0.4212	0.0012	2.6413	0.8618	1.1712	1.2989	1.7190
Left Shoulder	0.0002	0.4799	0.1630	0.2280	0.0224	0.3284	0.1561	0.1745	0.0007	0.6828	0.3144	0.3411	0.3870	0.4459
Right Knee	0.0004	0.5411	0.2341	0.2721	0.1595	0.3374	0.2190	0.2218	0.0001	0.4506	0.2057	0.2436	0.3809	0.4273
Left Knee	0.0001	0.3307	0.1266	0.1612	0.3838	0.7579	0.5526	0.5600	0.0001	0.2641	0.0861	0.1086	0.5734	0.5928
Right Ankle	0.0001	0.2641	0.0861	0.1086	0.0339	0.6287	0.3047	0.3432	0.0001	0.1351	0.0354	0.0506	0.3186	0.3635
Left Ankle	0.0912	0.6240	0.3557	0.3827	0.0573	0.2639	0.1565	0.1643	0.1409	0.5429	0.3781	0.3918	0.5422	0.5718
Training Test for 3000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0515	0.6427	0.2697	0.3175	0.4703	1.0074	0.7712	0.7843	0.0025	0.4389	0.2624	0.2816	0.8581	0.8918
Left Palm	0.0170	0.3021	0.1112	0.1283	0.3348	0.8011	0.6197	0.6324	0.0623	0.3231	0.2046	0.2167	0.6620	0.6807
Head	0.2290	0.7120	0.4461	0.4724	0.3330	0.7028	0.5570	0.5662	0.2460	0.5291	0.3363	0.3450	0.7889	0.8141
Upper Torso	0.0466	0.5526	0.2991	0.3150	0.0156	0.5784	0.4380	0.4494	0.0128	0.5129	0.2759	0.3008	0.5979	0.6258
Lower Torso	0.2496	0.6172	0.4612	0.4673	0.1730	0.4548	0.3216	0.3288	0.0001	0.1664	0.0350	0.0469	0.5633	0.5733
Right Shoulder	0.0018	1.5806	0.6349	0.6859	0.0003	0.9963	0.6150	0.6936	0.0043	1.5896	0.5176	0.7348	1.0243	1.2213
Left Shoulder	0.0003	0.4796	0.1775	0.2427	0.1434	0.5412	0.3963	0.4149	0.0038	0.3205	0.3205	0.3429	0.5397	0.5904
Right Knee	0.3267	0.9705	0.6378	0.6474	0.0740	0.2828	0.1690	0.1745	0.0004	0.3357	0.1085	0.1432	0.6687	0.6856
Left Knee	0.0001	0.3472	0.1487	0.1744	0.4003	0.8814	0.6918	0.7022	0.0561	0.3591	0.2192	0.2346	0.7408	0.7606
Right Ankle	0.0561	0.3591	0.2192	0.2346	0.3417	0.7271	0.5357	0.5465	0.0417	0.2033	0.1320	0.1380	0.5937	0.6105
Left Ankle	0.0001	0.2673	0.1080	0.1248	0.1728	0.3148	0.2335	0.2362	0.0758	0.6393	0.4517	0.4640	0.5198	0.5354
Training Test for 4000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0010	0.4986	0.2828	0.3073	0.2629	0.6454	0.4464	0.4551	0.0002	0.5030	0.2540	0.2937	0.5863	0.6227
Left Palm	0.0003	0.5658	0.2140	0.2137	0.4038	0.6819	0.5455	0.5494	0.0003	0.2927	0.1136	0.1352	0.5969	0.6048
Head	0.0002	0.5493	0.3645	0.3909	0.2055	0.4214	0.2989	0.3026	0.0023	0.4572	0.1871	0.2220	0.5072	0.5419
Upper Torso	0.0005	1.1522	0.5110	0.6128	0.1976	0.5850	0.3579	0.3691	0.0007	1.1237	0.4745	0.5305	0.7838	0.8906
Lower Torso	0.0019	0.4606	0.2819	0.3071	0.1560	0.3496	0.2440	0.2481	0.0010	0.5580	0.2403	0.2915	0.4436	0.4908
Right Shoulder	0.0006	5.8187	1.7217	2.6620	0.1786	2.2193	0.8004	1.0930	0.0003	5.5726	1.6070	2.5340	2.4874	3.8343
Left Shoulder	0.0001	0.4768	0.3057	0.3286	0.1968	0.4898	0.3460	0.3528	0.0015	0.6745	0.3614	0.4114	0.5863	0.6338
Right Knee	0.0024	0.6906	0.3877	0.4458	0.2336	0.4173	0.3282	0.3310	0.0005	0.5148	0.2212	0.2641	0.5540	0.6149
Left Knee	0.0029	0.4495	0.2284	0.2589	0.0984	0.6692	0.3715	0.4082	0.0031	0.8519	0.3435	0.4144	0.5551	0.6367
Right Ankle	0.0031	0.8519	0.3435	0.4144	0.0004	0.5885	0.3441	0.3943	0.0001	0.1210	0.0492	0.0581	0.4887	0.5750
Left Ankle	0.0056	0.3608	0.2369	0.2467	0.0001	0.1577	0.0623	0.0765	0.0001	0.3914	0.1841	0.2147	0.3064	0.3359
Training Test for 5000 Epochs														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0532	0.5253	0.2693	0.2995	0.1912	0.6608	0.3824	0.3959	0.0007	0.4666	0.2908	0.3141	0.5507	0.5874
Left Palm	0.0003	0.3306	0.1316	0.1622	0.0003	0.4748	0.2175	0.2540	0.1097	0.4872	0.3174	0.3357	0.4067	0.4511
Head	0.0006	0.5617	0.3068	0.3451	0.0037	0.4176	0.2500	0.2688	0.0001	0.1962	0.0866	0.1026	0.4051	0.4493
Upper Torso	0.0640	0.5373	0.3141	0.3520	0.0001	0.2765	0.1252	0.1537	0.0001	0.3749	0.1293	0.1587	0.3620	0.4156
Lower Torso	0.0007	0.6104	0.3843	0.4031	0.0001	0.2654	0.1126	0.1397	0.0006	0.5303	0.3511	0.3630	0.5326	0.5602
Right Shoulder	0.0069	1.6632	0.6372	0.7003	0.0001	0.5389	0.1091	0.1628	0.0001	1.4517	0.5098	0.5770	0.8233	0.9219
Left Shoulder	0.0235	0.4891	0.2197	0.2633	0.0001	0.3267	0.1540	0.1812	0.0001	0.4595	0.2356	0.2715	0.3571	0.4194
Right Knee	0.0036	0.8403	0.4710	0.5172	0.0001	0.2285	0.1105	0.1253	0.0027	0.4644	0.3158	0.3275	0.5777	0.6249
Left Knee	0.0004	0.4307	0.1821	0.2147	0.0002	0.2383	0.0715	0.0917	0.0658	0.6808	0.4157	0.4411	0.4594	0.4991
Right Ankle	0.0658	0.6808	0.4157	0.4411	0.0585	0.8037	0.5383	0.5940	0.0027	0.2006	0.1240	0.1304	0.6913	0.7513
Left Ankle	0.0001	0.1659	0.0795	0.0890	0.0006	0.2574	0.1064	0.1206	0.1384	0.7087	0.4468	0.4754	0.4661	0.4985
Training Test for 2000 Epochs with 500 frames														
	X Axis				Y Axis				Z Axis				Distance	
	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	Min [m]	Max [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Right Palm	0.0005	0.6301	0.3165	0.3665	0.0827	0.1969	0.1326	0.1358	0.0155	0.2825	0.1784	0.1977	0.3868	0.4380
Left Palm	0.0008	0.7446	0.3310	0.4119	0.1791	0.3081	0.2412	0.2433	0.0785	0.3187	0.1874	0.2004	0.4504	0.5187
Head	0.0003	0.5994	0.3360	0.3821	0.0497	0.1373	0.0855	0.0873	0.0892	0.2604	0.1686	0.1750	0.3855	0.4292
Upper Torso	0.0009	0.5671	0.3374	0.3816	0.0039	0.1041	0.0473	0.0516	0.3032	0.5148	0.3892	0.3946	0.5173	0.5514
Lower Torso	0.0005	0.6655	0.3384	0.3942	0.0664	0.1474	0.0958	0.0972	0.2230	0.4908	0.3287	0.3393	0.4814	0.5291
Right Shoulder	0.0006	0.6393	0.3420	0.3930	0.0223	0.1106	0.0581	0.0608	0.1213	0.3555	0.2383	0.2495	0.4209	0.4695
Left Shoulder	0.0011	0.6299	0.3369	0.3862	0.0254	0.1277	0.0695	0.0725	0.3141	0.5410	0.4099	0.4153	0.5351	0.5717
Right Knee	0.0003	0.6497	0.3588	0.4078	0.1580	0.2771	0.1985	0.1998	0.0290	0.3530	0.2049	0.2288	0.4584	0.5085
Left Knee	0.0005	0.7429	0.3560	0.4303	0.1265	0.2615	0.1757	0.1788	0.4147	0.7033	0.5248	0.5310	0.6580	0.7065
Right Ankle	0.4147	0.7033	0.5248	0.5310	0.0022	0.5565	0.3746	0.4142	0.0079	0.0951	0.0381	0.0438	0.6459	0.6749
Left Ankle	0.0032	0.6513	0.3033	0.3544	0.0198	0.1398	0.0649	0.0737	0.1112	0.4546	0.2783	0.2885	0.4167	0.4629

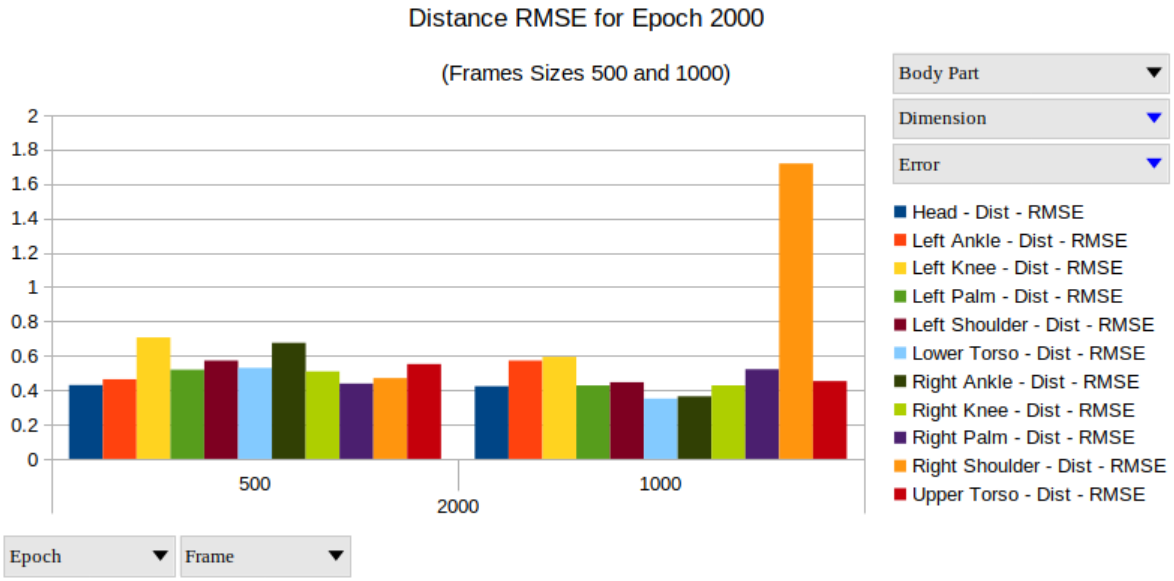

Fig. 5. Comparison between datasets 500 and 1000 frames for epoch 2000.

5 Conclusions

An advanced user interface for visualizing and editing C3D dataset was proposed in this paper. The application considers the case of editing traditional Greek dance sequences, obtained using VICON sensors. The interface allows the user to monitor the dance choreography, step-by - step, filter the noise using clustering approaches and create a rough estimation of the dancers body joints.Different types of functionalities are supported such as noise removal of 3D points that do not correspond to human skeletons and clustering. The developed interface was built on C ++ programming on the exploitation of the OpenGL language. The interface is extensible in the sense that it can parse any C3D type following the precise instructions and recommendations of the standard.

Another key innovation of this paper is the use of a novel deep machine learning framework for the extraction of human skeleton. Deep machine learning is implemented through Convolutional Neural Networks. Initially a rough human skeleton is extracted using clustering approaches.

Experimental results and validation on real-life dance 3D point video sequences are conducted. The results show the excellent performance of the proposed method and the ability of the developed interface to support any type of dance sequence.

6 Acknowledgments

This work was supported by the EU H2020 TERPSICHORE project Transforming Intangible Folkloric Performing Arts into Tangible Choreographic Digital Objects under the grant agreement 691218.

References

1. S.-Y. B. Shay, A., *The Oxford Handbook of Dance and Ethnicity*. Oxford University Press: Oxford, UK, 2016.
2. K. Dimitropoulos, S. Manitsaris, F. Tsalakanidou, B. Denby, L. Buchman, S. Dupont, S. Nikolopoulos, Y. Kompatsiaris, V. Charisis, L. Hadjileontiadis, F. Pozzi, M. Cotescu, S. Ciftci, A. Katos, A. Manitsaris, and N. Grammalidis, "A multimodal approach for the safeguarding and transmission of intangible cultural heritage: The case of i-treasures," *IEEE Intelligent Systems*, 2018.

3. A. D. Doulamis, A. Voulodimos, N. D. Doulamis, S. Soile, and A. Lampropoulos, "Transforming intangible folkloric performing arts into tangible choreographic digital objects: The terpsichore approach.," in *International Conference on Computer Vision, Theory and Applications (VISIGRAPP)*, (Porto, Portugal), pp. 451–460, 2017.
4. Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimedia*, vol. 19, pp. 4–10, 2012.
5. M. Windolf, N. Gtzen, and M. Morlock, "Systematic accuracy and precision analysis of video motion capturing systems-exemplified on the vicon-460 system," *Journal of Biomechanics*, vol. 41, pp. 2776–2780, 2008.
6. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," pp. 1297–1304, 2011.
7. A. Kitsikidis, K. Dimitropoulos, S. Douka, and N. Grammalidis, "Dance analysis using multiple kinect sensors," in *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, vol. 2, pp. 789–795, 2014.
8. D. Kim, D.-H. Kim, and K.-C. Kwak, "Classification of k-pop dance movements based on skeleton information obtained by a kinect sensor," *Sensors MDPI*, vol. 17, 2017.
9. K. Hisatomi, M. Katayama, K. Tomiyama, and Y. Iwadate, "3d archive system for traditional performing arts: Application of 3d reconstruction method using graph-cuts," *International Journal of Computer Vision*, vol. 94, pp. 78–88, 2011.
10. E. Stavrakis, A. Aristidou, M. Savva, S. Himona, and Y. Chrysanthou, "Digitization of cypriot folk dances," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7616 LNCS, pp. 404–413, 2012.
11. I. Rallis, I. Georgoulas, N. Doulamis, A. Voulodimos, and P. Terzopoulos, "Extraction of key postures from 3d human motion data for choreography summarization," in *Proc. of the IEEE 9th International Conference on Virtual Worlds and Games for Serious Applications, (VS-Games)*, pp. 94–101, 2017.
12. I. Rallis, N. Doulamis, A. Doulamis, A. Voulodimos, and V. Vescoukis, "Spatio-temporal summarization of dance choreographies," *Computers and Graphics (Pergamon)*, vol. 73, pp. 88–101, 2018.
13. E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1600–16007, 2012.
14. H. Wang, "A survey on deep neural networks for human action recognition based on skeleton information," *Advances in Intelligent Systems and Computing*, vol. 541, pp. 329–336, 2017.
15. E. Protopapadakis, A. Voulodimos, A. Doulamis, S. Camarinopoulos, N. Doulamis, and G. Miaoulis, "Dance pose identification from motion capture data: A comparison of classifiers," *Technologies*, vol. 6, no. 1, p. 31, 2018.
16. S. Dewan, S. Agarwal, and N. Singh, "A deep learning pipeline for indian dance style classification," vol. 10696, 2018.
17. K. Dimitropoulos, P. Barmpoutis, A. Kitsikidis, and N. Grammalidis, "Classification of multidimensional time-evolving data using histograms of grassmannian points," *IEEE Transactions on Circuits and Systems for Video Technology*, to appear.
18. A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, 2018.
19. T. McReynolds and D. Blythe, *Advanced Graphics Program. Using OpenGL*. 2005.
20. M. L. S. Inc, *The C3D File Format User Guide*. United States of America, 1997-2008.
21. S. Alfalah, W. Chan, S. Khan, J. Falah, T. Alfalah, D. Harrison, and V. Charissis, "Gait analysis data visualisation in virtual environment (gadv/ve)," in *Proceedings of 2014 Science and Information Conference, SAI 2014*, pp. 742–751, 2014.
22. "Biomechanical toolkit: Open-source framework to visualize and process biomechanical data," *Computer Methods and Programs in Biomedicine*, vol. 114, no. 1, pp. 80 – 87, 2014.
23. T.-H. Nguyen and V.-N. Huynh, "A k-means-like algorithm for clustering categorical data using an information theoretic-based dissimilarity measure," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9616, pp. 115–130, 2016.
24. K. Makantasis, A. Doulamis, N. Doulamis, and K. Psychas, "Deep learning based human behavior recognition in industrial workflows," in *Proceedings - International Conference on Image Processing, ICIP*, vol. 2016-August, pp. 1609–1613, 2016.
25. I. L. M. Oquab, L. Bottou and J. Sivic, "is object localization for free? - weakly-supervised learning with convolutional neural networks,," in *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, pp. 685694*, June 2011.
26. Y. J. e. a. C. Szegedy, W. Liu, "going deeper with convolutions,," in *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 15), pp. 19, Boston, Mass, USA*, June 2011.

27. Wikipedia, "Convolutional neural network." "https://en.wikipedia.org/wiki/Convolutional_neural_network".
28. I. S. A. Krizhevsky and G. E. Hinton, "imagenet classification with deep convolutional neural networks,," in *in Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS 12), pp. 10971105, Lake Tahoe, Nev, USA*, December 2012.
29. T. D. R. Girshick, J. Donahue and J. Malik, "rich feature hierarchies for accurate object detection and semantic segmentation,," in *in Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, 580587*, June 2015.