A New Bio-Informatics Framework: Research on 3D Sensor Data of Human Activities

Sajid Ali
College of Information Science and Technology, Beijing Normal University, Engineering Research Center of Virtual Reality and Application, Ministry of Education, Beijing, China

*Wu Zhongke
College of Information Science and Technology, Beijing Normal University, Engineering Research Center of Virtual Reality and Application, Ministry of Education, Beijing, China

Muhammad Saad Khan
School of Electronic Engineering
Beijing University of Post Telecommunication, Beijing, China

Ahmad Tisman Pasha
Department of Information Technology,
Institute of Computing,
Bahauddin Zakariya University
Multan, Pakistan

Muhammad Adnan Khalid
Institute of Automation
Chinese Academy of Sciences,
Beijing, China

Zhou Mingquan
College of Information Science and Technology, Beijing Normal University, Engineering Research Center of Virtual Reality and Application, Ministry of Education, Beijing, China

***Abstract*—Due to increasing attraction of motion capture systems technology and the usage of captured data in wide range of research-oriented applications, a framework has developed as an improved version of MOCAP TOOLBOX in Matlab platform. Firstly, we have introduced a faithful script to deal with public motion capture data, which will be friendly for us. Various functions through dynamic programming, by using the Body Segment Parameters (BSP) are edited and they configured the position of markers according to data. It is used to visualize and refine without the MLS view and the C3D editor software. It has opened a valuable way of sensor data in many research aspects as gait movements, marker analysis, compression and motion pattern, bioinformatics, and animation. As a result, performed on CMU and ACCAD public mocap data, and achieved higher corrected configuration scheme of 3D markers when compared with the prior art, especially for C3D file. Another distinction of this work is that it handles the extra markers distortion, and provides the meaningful way to use captured data.**

***Keywords**—Sensor data; Bioinformtics behavior; Optical actives; BSP domain; Marker Configuration*

I. Introduction

Due to improvement in motion, capture technologies such as optical, mechanical, or magnetic sensors are attached to human joints and their movements are record. Such systems depend on an active source, which emits pulses of infrared light with a high frequency, which is reflected by small spherical markers, or LEDs attached to track the subject (e.g. contribution of subject walking, or running). In motion capture system, every camera captures the position of reflective markers in two-dimensional cameras. Network of system computes position data in 3-dimensions. Now more and more researchers are interested to use 3D-mocap data with different types of application; such as retarget [1-3], analysis [4], animation [5] and surveillance systems [6]. In addition, they have demanded for different types of toolbox to utilize the variety of data ASF/AMC, BVH [7], and C3D [8] file format for multiple research purposes. Some of them have interested to develop mocap toolbox; which are helpful to other scientists. They are focused to use the captured data in various research directions. Recently, Jeroen and Boxtel [9] have developed a toolbox (Biomotion toolbox) in the Matlab environment that can read and display the different types of mocap data by using Psychtoolbox-3[10]. This third party toolbox (Psychtoolbox-3) is not exactly suitable for biomotion. However, some features of the biomotion are limited, and have specific designs to display and manipulate point-light displays (PLD). Charles Vernon [11], developed a toolbox with a limited number of functions. It provides a graphical user interface (GUI). The major mocap toolbox in MathWorks platform is mainly dealing with recorded data by infrared marker based optical motion capture system.

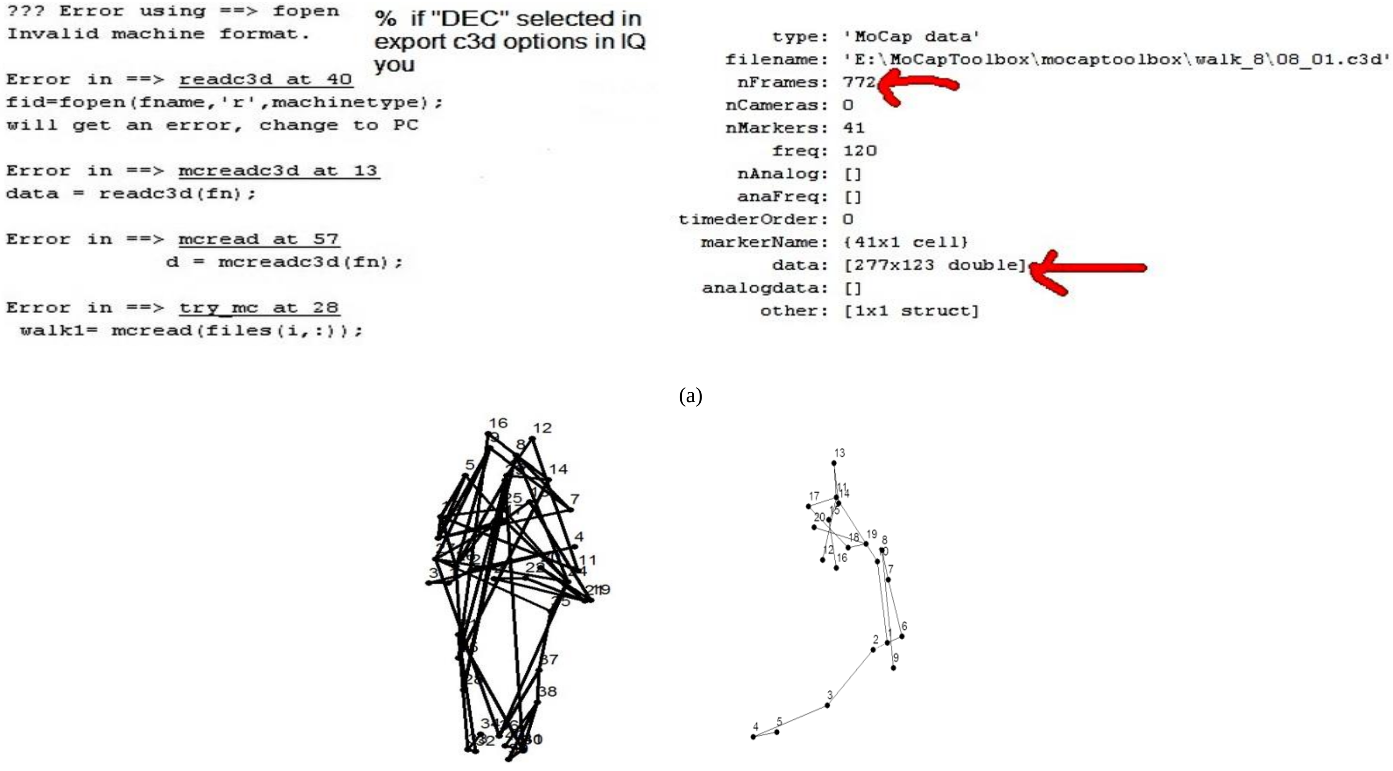
The MathWork provides precompiled functions[12], and they are used on different types of data. Some of the functions are used as part of mocap toolbox, such as PCA and ICA packages, Signal Processing and Statistical Toolbox. So user can design functions and scripts in the Matlab environment according to their requirements. Recently, Burger and Petri [13] developed the MoCap Toolbox having 64 functions, excluding other toolbox packages. These functions have been used to visualize and analyze captured data and have the capability to read the different types of data format. They notified three parts: 1)-Motion Data Structure (MDS), 2)-Segment Data Structure (SDS) and 3)-Normal Data Structure. These structures have interconnection and processed. Their computation implied statistical and mathematical methods in order to propose a homogeneous framework and their analysis and simulation (animation). They also claimed that their work can read C3D file, but still it has some indispensable issues to trade with such type of data. It cannot read the public mocap data. We have inspired of work [14], our focus is to refine functions and use them on data (C3D format) which   will be used for different purposes such as in clinical field, retarget motion and animation. In [13], we have big issues and challenges for mocap data researchers. These are: 1)-reading the C3D file not displaying in human skeleton shape (see Figure 1 a), 2)-Markers configuration which shows the motion data structure of the human skeleton, 3)-Normalized the data of the human skeleton (see Figure 1 b, c); and 4) -required Qualisys software to manage the .dat and.mat data format. We have tried hard to resolve these individual issues see.

`Output= mcread ('E: \walk.c3d');` (1)

We use some of the existing functions after embedding script (`C3D_VaxD2PC`) into the toolbox and read the data successfully by (1). The following functions have been used to display calibrated 3D markers position and form them into the human skeleton shape.

However, it is not like human skeleton. It looks like network connection between marker nodes because some existing toolbox functions are not operated according to marker positions (see Figure-1 (b) & (c)). The functions used on mocap data before modification are:

```
mcplotframe(walk1, 160, mapar);
mcplotframe(walk2j, 160, japar);
```



```
??? Error using ==> fopen
Invalid machine format.

Error in ==> readc3d at 40
fid=fopen(fname,'r',machinetype);
will get an error, change to PC

Error in ==> mcreadc3d at 13
data = readc3d(fn);

Error in ==> mcread at 57
            d = mcreadc3d(fn);

Error in ==> try_mc at 28
 walk1= mcread(files(i,:));
```

% if "DEC" selected in export c3d options in IQ you

```
         type: 'MoCap data'
     filename: 'E:\MoCapToolbox\mocaptoolbox\walk_8\08_01.c3d'
      nFrames: 772
     nCameras: 0
     nMarkers: 41
         freq: 120
      nAnalog: []
      anaFreq: []
timederOrder: 0
   markerName: {41x1 cell}
         data: [277x123 double]
   analogdata: []
        other: [1x1 struct]
```

(a)

(b) (c)

Fig. 1. (a) Indicates the errors to in reading the C3D file, before introducing a script. After removing, the errors and we still have a tremendous data problem such as frame issue, (b) existing setting markers position in function.

Keeping the above issues, we have refined some functions of Mocap Toolbox to stem these issues and compiled them successfully according to data [15]& [16]. It can be associated with other toolboxes in the Matlab environment such as Mocap136 [17] and Robotics [18]. The pictorial structure of the improved version of Toolbox has been given in Figure 2.

Lot of research works have been done on mocap data. Improved version can be used in different research fields such as joint analysis, design the locomotion pattern, retarget motion, human skeleton animation, motion classification, 3D pose estimation and human identification by using 3D mocap databases plausible with the human body segment parameters (BSP).

The rest of this paper is set up as follows. Section II gives a concise story about motion capture systems, and data preprocessing. Section III describes the configuration scheme of sensors and data visualization functions. Brief descriptions of the human Body Segment Parameters (BSP) and data normalization are discussed in section IV. The experimental results, conclusion and future work are explained in section V.

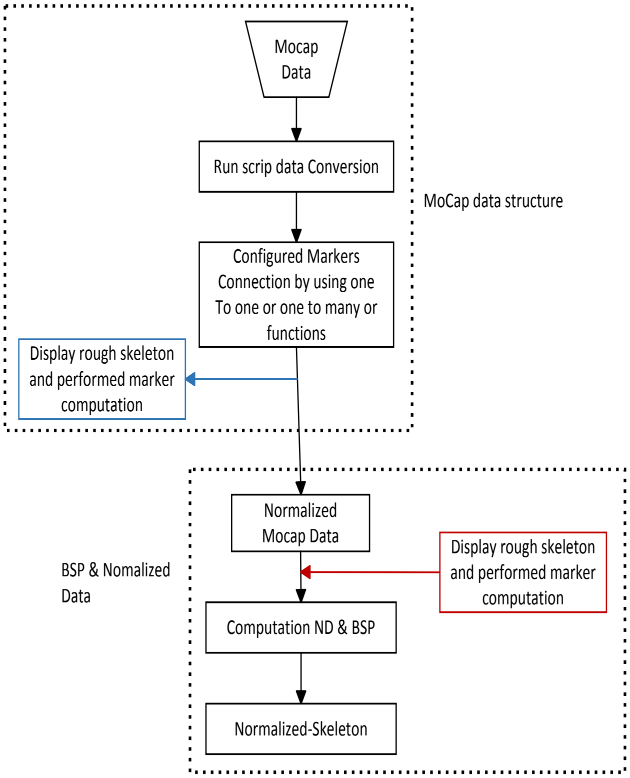

Mocap Data
Run scrip data Conversion
MoCap data structure
Configured Markers Connection by using one To one or one to many or functions
Display rough skeleton and performed marker computation
Normalized Mocap Data
BSP & Nomalized Data
Display rough skeleton and performed marker computation
Computation ND & BSP
Normalized-Skeleton

Fig. 2. Flowchart of the improved version mocap toolbox

II. MOCAP DATA SYSTEMS

Motion Capture Sensor Systems (MCSS) record the MOCAP data. Such systems are highly sophisticated, and require a certain number of motion capture session during subject activities. They are classified into two categories (1) Optical and (2) Non-optical Capture Sensor Systems. They provide different type data format. They are *. C3D, *. BVH, *.txt,*.tvt, and *. ASF/AMC.

A. Mocap Data Preprocessing

Preprocessing of data is a primary step to get accurate results in any scientific research [19]. Data captured by ubiquitous sensors based on subject movement is stored as the C3D file format with necessary attribute such as start and end times, sensor id and sensor values. It depends upon several types of hardware platform issues such as DEC (Digital Equipment Corporation), SGI/MIPS (Microprocessor without Interlocked Pipeline Stages)) and Intel. They represent the different floating-point numbers and are stored accordingly in hardware (VAX-D, IEEE-LE and IEEE-BE Vicon call them a "C3D_VaxD2PC”. It establishes the strong connection between MOCAP TOOLBOX function and data. It is used before the mcread () function, its syntax as

```
Output =C3D_VaxD2PC ('Convert’,’ data
location');                          (2)
```

It converts the C3D files into PC format, because these files depend on several types of hardware and floating-point. After compiling data through (2). MocapToolbox mcread () function, read C3D files, and arrange them into a reliable database. Here its name is called ‘Processed Data for MocapToolbox’.

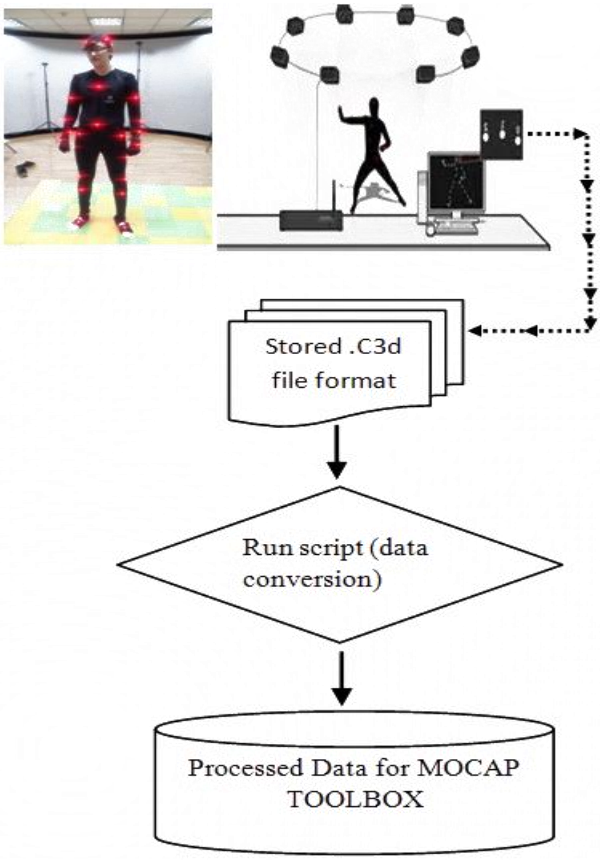

Stored .C3d file format
Run script (data conversion)
Processed Data for MOCAP TOOLBOX

Fig. 3. Procedure of processing public domain mocap data for enhanced MOCAP TOOLBOX

As shown in Figure 1 (a), an error has resolved by emitting this function. The graphical representation of the database can be seen in Figure 3.

After reading the C3D file successfully, some remaining typical issues have shown in Figure 1 (c & d). They have affected the standard human skeleton; attached markers on human body and joint positions. Both of these issues will be addressed in the section III through `mapar.conn` and japar.conn functions.

III. MOTION SENSORS CONFIGURATION AND VISUALIZATION

A. Motion Sensors Configuration

The markers or motion sensor configuration on the human body has many possible ways for motion recording one of them is shown in Figure-4.

In Figure-4, 41 sensors have been configured on the human body to follow the configuration scheme of CMU motion capture system and are described in the Vicon 512 manual [20]. We analyzed the sensor labels of the C3D file by using 3MAX software. Keeping to these labels, we assessed the sensor values after editing the mcreadC3D function according to the markers order of template (see Figure-A.1). Data keeps maintaining some dumping markers during the motion capture session of human activities such as walking, running, dancing etc.

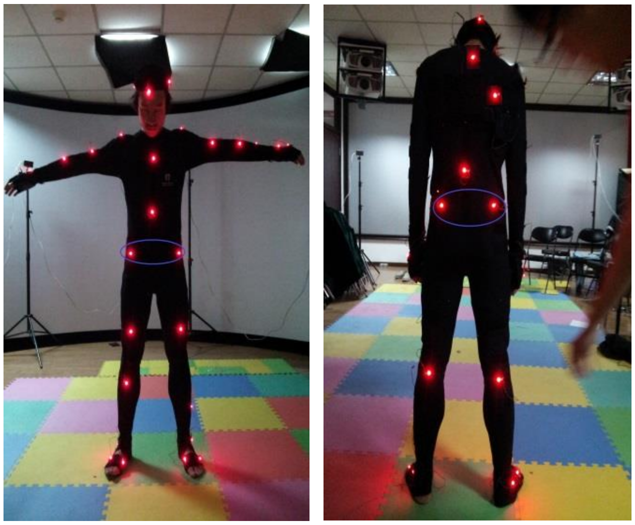


Fig. 4. An example, markers, or sensor configuration scheme on the human body according of CMU data

We modify the mcread C3D function by using the following steps:

1) First access the index of motion sensors which are placed on the human body by using the code (Appendix A.1. 1):

For example, accessing the L_finger index of motion sensor from C3D file of CMU database (see Appendix A.1).

Similarly have accessed the other remaining 40 or 41 labels from file.

2) Some extra markers have stored during the recording we handled and assigned zero values; and adjusted them with marker labels.

3) Many mocap data, scientists use C3D format. They feel hard to export and construct raw motion data to the desired model. They have unmatched position between the calibration motion sensors. These positions are unable to make human model. For this matter, we used a `mapar.conn ();` and configure according to public mocap data. It is configured by using the mathematical methods. They are as one to one and one to many (see Figure 5).The following function parameters can be driven (see Appendix A.1.1 for code of accessing sensor labels),

```
mapar.conn=[12 16;16 9;9 8;12 8;14 7;7 4;4 11;11 19;19 21;11 24;21 24;....
5 10;10 6;6 27;27 26;27 3;3 1;1 26; 22 23;22 20; 20 2; 2 23;31 15;15 28;...
28 34;34 36;36 32;34 33;33 32;35 37;37 38;38 30;30 39;30 41;41 40;39 40;...
 23 31;31 2;23 15;2 15;20 35;22 35;20 37;22 37;5 23;2 5;22 14;20 14;14 13;...
13 5;23 25;2 25;20 17;22 17;13 29;29 25;29 17;17 18;25 18;5 25;25 14;...
17 5; 17 14; 20 25; 5 18; 18 14];
```

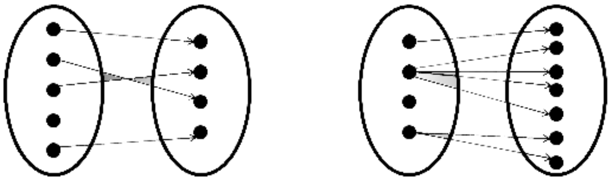


Fig. 5. Configuration scheme between markers, (a) One to one; and (b) One to many mapping

After successful compilation steps 1, 2, and 3 on data, step to construct the human skeleton model by using the following function, seen in Figure 6.

```
mcplotframe (walkm,180, mapar);   (3)
```

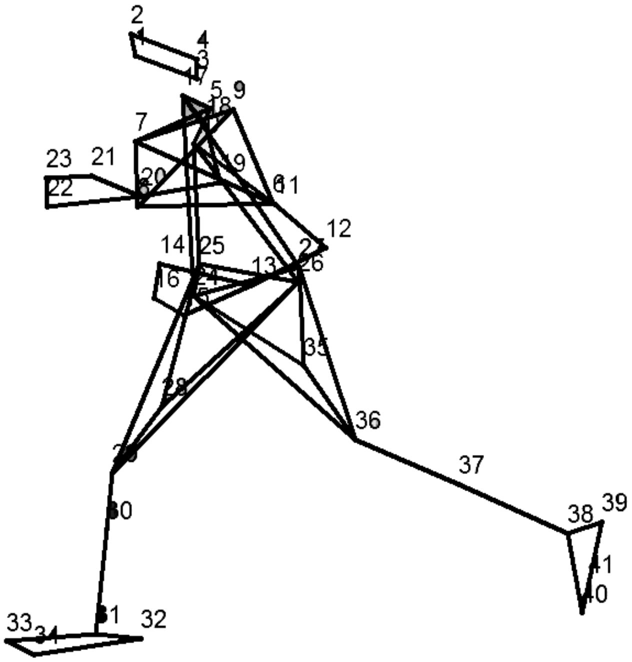


Fig. 6. Specific pose of skeleton after steps 1, 2, and 3 by using function (3)

IV. Human Body Segment Parameters and Data Normalization

There are many methods [21-31] to compute the human body segment parameters. They have trendy source of BSP knowledge of human body. The BSP and mocap data should comply on each other.

A. Human Body Segment Parameters

Body segment parameters perform an important role to generate motion of human activities by motion capture systems. The mcgetsegmpar () function parameters of the Mocap Toolbox use the Dempster computed BSP data [32]. We replaced BSP parameters [32] of mcgetsegmpar () function to Zatsiorsky and Seluyanov adjusted by the de Leva [33], and their computation is [34]. For instance, an example of the BSP computation of specific human body part can be seen Eqs A.1 to A.6 in Appendix A.1.3. These parameters and 3D motion capture data used to visualize the standard human skeleton with 20 joints (see Figure 7).

B. Mocap Data Normalization and Visualization of Skeleton Refinement

In order to investigate the marker's positions of C3D file and their indexing, we configured the mapar.conn () and japar.conn () function according to indices. The source code of markers indexing configuration (Appendix A.1.2**.**).

The m2jpar() depends on mcm2j() function. It has the knowledge to compute the translation from markers to joint representation. The joint position has computed by applying the center method around placed marker on human body joints. For instance, the root position of joint is found-out between the 22,23,2 and 20 marker (markers marked with blue ellipses see Figure 4). Similarly, other joint positions are computed. We label each joint with specific name. They can be seen in Table 1. The following function uses to initialize required joint parameters.

```
japar = mcinitanimpar;
```

It contains information that will be helpful to initialize joint parameters and assigns the attributes of japar () structure. One of the fields of this structure is to edit the parameter by putting joint index of m2jpar and make the connection between these indices by applying the faithful methods (see Figure-5). These functions are used by applying japar.conn () Parameters can be accessed (see source code in Appendix A.1.3) and attain human skeleton (see Figure-7 with a specific pose and generated animation motion frame see Figure-8).

```
japar.conn=[1 2; 2 3; 3 4; 4 5...
     1 6;6 7;7 8;8 9;...
     1 10; 10 11; 11 12;...
     11 13; 13 14; 14
     15; 15 16...
     11 17; 17 18; 18 19; 19 20];
```

Expect of two joints 11 and 1, used the one to one function definition. The following function visualize the skeleton having 20 joints (see Figure 7),

```
mcplotframe (walk2j,180, japar);
```

The following function performs the animation pose of extracted skeleton of the C3D file from CMU database. It also has ability to create animation of the mocap data after editing some mocap Toolbox functions. The Figure 8 is created by using the `mcanimate ()` function as,

```
mcanimate (walk2j, 15, japar);      (4)
```

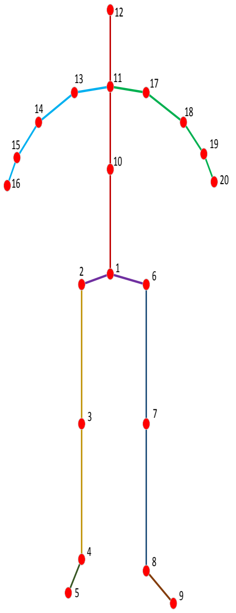


Fig. 7. Normalized Skeleton

TABLE I. SKELETON JOINT NAMES

1	root
2	lhip
3	lknee
4	lankle
5	ltoe
6	rhip
7	rknee
8	rankle
9	rtoe
10	midtors
11	neck
12	head
13	lshoulder
14	lelbow
15	lwrist
16	lfinger
17	rshoulder
18	relbow
19	rwrist
20	rfinger

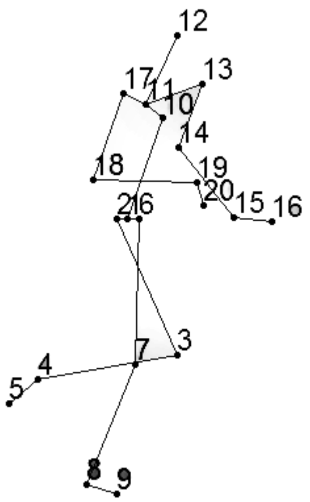
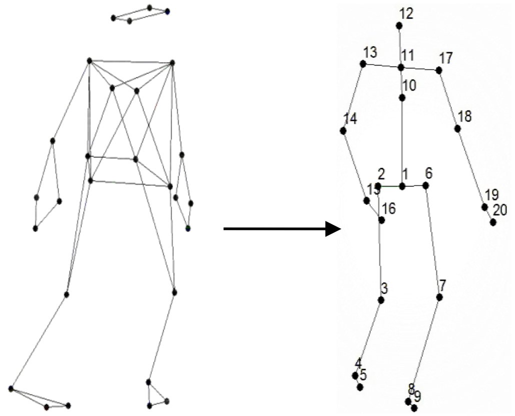


Fig. 8. An example of specific animated pose by using the function (4)



(a). Mesh human skeleton (b). Stick skeleton

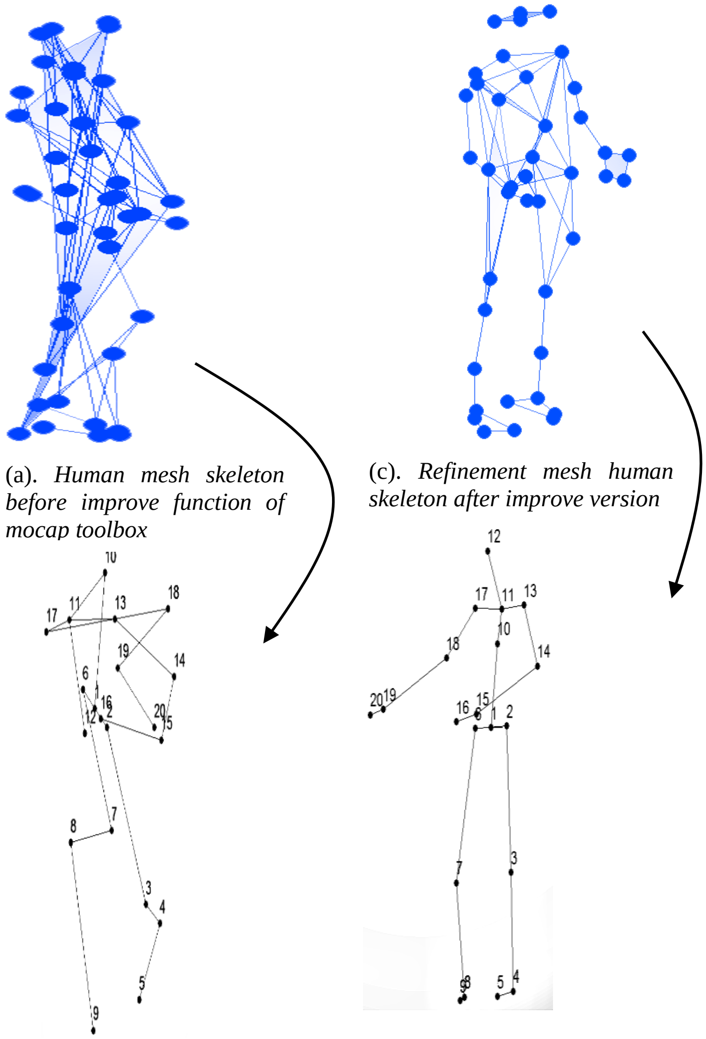
Fig. 9. (a) Triangular mesh skeleton contain 28 sensors, and (b) human skeleton with 20 joints [34]

After modification of toolbox functions, it can be enabled to read the C3D file successfully from public 3D mocap databases. Some of the existing function performances will be discussed in the section V.

V. Results & Discussion

In this paper, we will improve the Mocap Toolbox by introducing new script and editing some functions, it can be used as input mocap data such as CMU, and ACCAD (Advance Computing Centre for the Arts and Design) databases. Earlier, Toolbox demonstrates 28 sensors and formed a triangular mesh human model and extract a skeleton (see Figure-9 a & b) by applying the structure of connection matrix. It had been collected by using the Qualisys motion capture system. After that, we used public mocap data (C3D files) and found the fundamental errors (see Figure 1). This shows that data could not support because of more than 28 sensors.

Meanwhile, the public data has at least 41, or 42 markers, it is as standard form but more than that such as 80, 90, and 356 create the ambiguity and difficult trouble to understand in many research fields. This issue can be handled by adding the code into mcreadC3Dc3d (). This code would be seen in (Appendix A.1.2.). We modified a list of functions of the Mocap Toolbox according to CMU and ACCAD mocap database. The list of functions is as mcread C3Dc3d (), mapar.conn (), japar.conn , mcm2j() (), mcgetsegmpar (), and m2jpar (). The visualization of placed markers on the human body and is transferred into skeleton by using some editing functions. It is illustrated in Figure 10. It shows the effects of the edited functions. The rest of functions also can also be used for public mocap data. Some examples are as follows:



(a). *Human mesh skeleton before improve function of mocap toolbox*

(c). *Refinement mesh human skeleton after improve version*

(b). *Human stick skeleton before editing function of mocap toolbox*

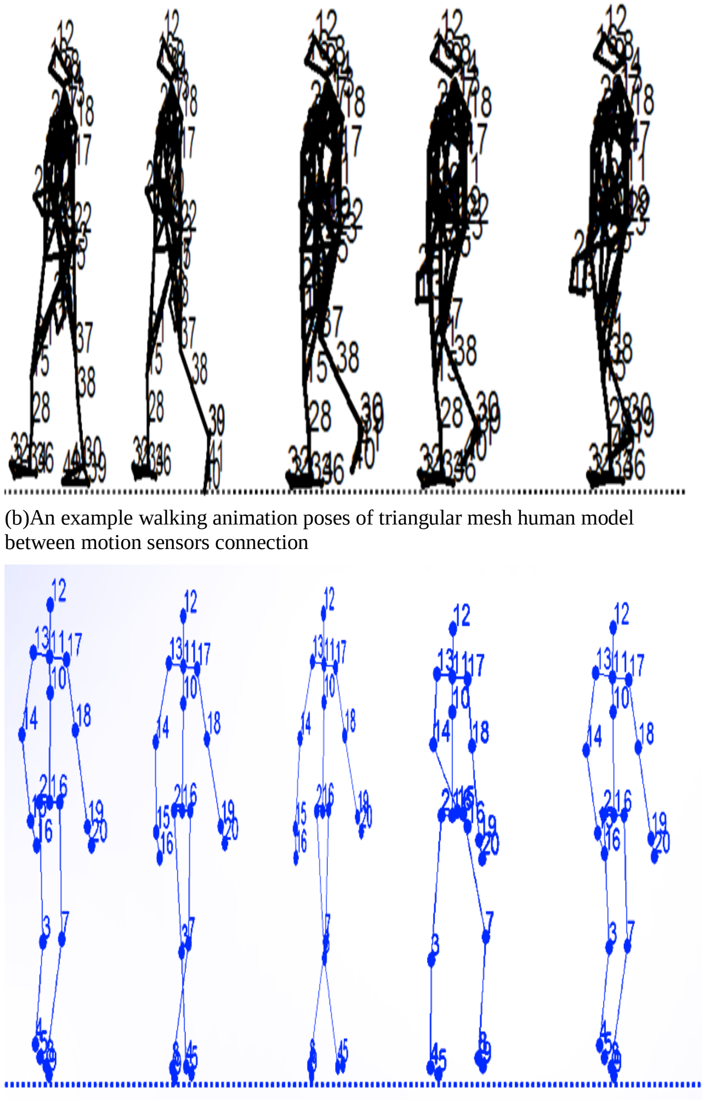
(d). *Refinement of 3D human stick skeleton after editing function*

Fig. 10. (a) & (b) describe the sensor connection through mapar.conn structure field to construct a mesh human skeleton and human stick skeleton extracted by using mcm2j function is used to transform these sensors information to the joint positions;(c) and (d) proved the edited functions that explained in section III, IV

```
newpar1 = mcanimate(walk1, mapar);   (5)
newpar2 = mcanimate(walk2j, japar);  (6)
```

These two functions (5 & 6) executed successfully on mentioned data, and demonstrate the following animation poses. It will be used to analyze the performance of placement marker pose by pose of human movements.

As shown in Figure 11. (a), the successful result of animation poses of the human mesh skeleton, between markers interconnection, which complied with all required parameters in function (8). Figure 11. (b) shows the successful result of animation poses of normalized human skeleton, which extracted from 41 markers, and compiled all required parameters in function (9). The parameters (walk1 and mapar) deal with placing markers on the human body; and they (walk2j, japar) deal with human joint positions.



(b)An example walking animation poses of triangular mesh human model between motion sensors connection

(a)An example of animation poses of normalized human stick model

Fig. 11. After modification functions results (a) marker positions of animation poses of the human waking results, and (b) normalized human skeleton of walking animation poses of CMU mocap data

The following functions are helpful to assess the accuracy of placing sensors on the human body during the motion recording sessions giving the information of missing markers, which are very useful for researchers. (See Figure 12). In Figure-12, black shaded colors indicate the missing markers and number of frames from a C3D file in CMU database.

```
[mf, mm, mgrid] = mcmissing (walk1);
subplot(3,1,1),
bar(mf), xlabel('Marker'), ylabel('Num.
of Missing frames')
subplot(3,1,2),
bar(mm), xlabel('Frame'), ylabel('Num. of
Missing markers'subplot(3,1,3),
imagesc(-mgrid'), colormap gray,
xlabel('Frame'), ylabel('Marker'
```

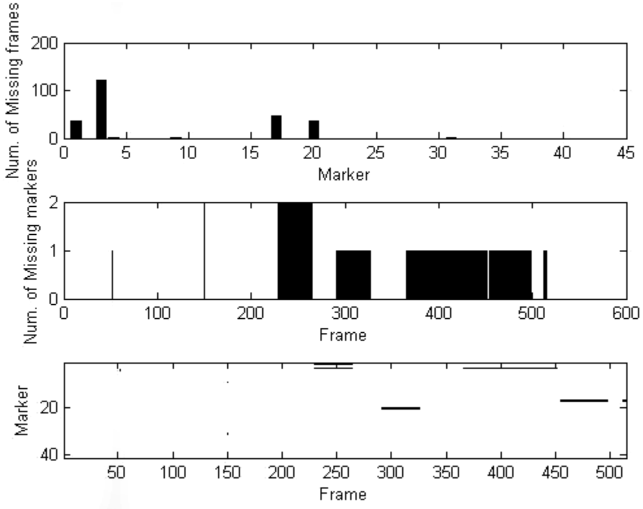


Fig. 12. Analysis of sensors and frames

The rest of other existing functions in Mocap toolbox (framework) can be applied on mocap data, and some of them can plot with respect to time as a `mcplottimeseries` function. For example, hip joint sensor with xyz coordinates missing frames, the missing area marked with a circle (see Figure 13) information is related to female subject no B20 in ACCAD mocap database.

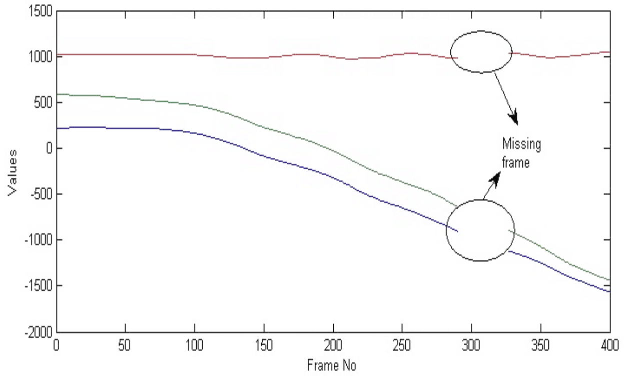


Fig. 13. An example of hip joint movement missing frame

We compared some of the above functions (see Table 2) to early version of the toolbox with respected to response time under a platform. We performed this evaluation on two public mocap databases and existing data of the toolbox. The details of these experiments are presented above. We have performed all above experiments under some specifications such as 2 GB

RAM, Intel(R) Core™ i5 CPU M520 @ 2.4 GHz Dell i5 Intell core 2 CPU 2.4 and window 7 ultimate 64-bit and MATLAB R2012a. In Table 2, fourth column contains two colors red and blue ellipse. The red denotes the edited functions and blue indicate early function response time. Finally, we concluded that red has taken more time to blue because these function use public data but opened for everyone who want to test mocap data for multiple applications. The conclusion is formed in Figure 14 with functions time response.

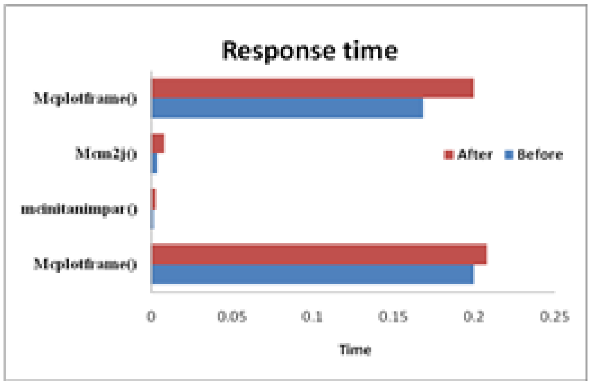


Fig. 14. Response timing comparison between different function. The red bar indicates our editing function according to public mocap data, and blue indicates the early functions with 28 markers configuration

VI. Conclusion and Furtur Work

In this study, improved version is used partly due to the inherent structure information from Mocap toolbox. The key assumption of this improved version is that there is a high probability of using public mocap data. It is in native C3D file format. It has rich and plentiful information of human, which offers various types of numerical treatment. In future, we hope that it will be allowed for a prosperous variety of functions used for motion capture data for numerous research fields like animation, human joint analysis, gender and human identification (bioinformatics), retarget motion in real time environment.

Our improved version is proved to be effective in finding human mesh and stick skeleton model from public data as compared to capture with previous development platform [34] because it support only 28 3D markers. These skeleton models will be useful for joint movements' analysis and multiple purposes in indoor and outdoor environment. In addition, it gives the information about the quality of mocap-captured data (see Figure 12, 13).

Based on the investigation, we conclude that which type of data is more reliable. The Xense[1] has introduced a state-of-the-art of inertial high sensory quality systems to capture the motion of subjects with wide range of area. It is capable of supporting and generating different motion formats, such as BVH, C3D etc. These types of data is very different from the image and video data. It has real and truthful relics of the human and object activities.

TABLE II. Time Evaluation of Before and after Editing Toolbox Functions

S. No	Function Name	3D MoCap Data	CPU Elapsed time With respect to functions & data	Comments
1	mcreadc3d()	CMU &ACCAD databases	0.204703s	Before improved of it, it could not be operated on public Mocap data. It prepare data by Qualisys Mocap system then author used. After improved Toolbox, we used public Mocap data. This is response time to read C3D file from Mocap database.
2	Mcplotframe()	Qualisys Mocap System with 28 makers	0.199483s	This is response time of Mcplotframe(),and display See (Figure 9.a). It contains 28 sensor moreover, showsTriangular human mesh model.
3	Mcplotframe()	CMU &ACCAD databases	0.207661s	This is response time of Mcplotframe(), and display See (Figure 10.c). It contain 41 or 42 sensor moreover, showsTriangular human mesh model.
4	mcinitanimpar	Qualisys Mocap System with 28 makers	0.000762s	It is response time to initialize the parameter to 28 joints
5	mcinitanimpar	CMU &ACCAD databases	0.002590s	It is response time to initialize the parameter to 41 or 42 joints
6	Mcm2j()	Qualisys Mocap System with 28 makers	0.003352s	It is response time transfer 28 sensors to 20 joints position
7	Mcm2j()	CMU &ACCAD databases	0.007916s	It is response time our configured 41 or 42. They transfer into 20 joints position.
8	Mcplotframe()	Qualisys Mocap System with 28 makers	0.168140s	This is response time of mcplotframe(walk2j,100,japar). It is extracted stick skeleton human model From 28 markers display See (Figure 9.b). It contains 20 joints and, Shows human stick model.
9	Mcplotframe()	CMU &ACCAD databases	0.199701s	This is response time of mcplotframe(walk2j,50,japar). It is extracted stick skeleton human model from 41 markers display See (Figure 10.d). It contains 20 joints and, shows human stick model.

[1] www.xsens.com

ACKNOWLEDGMENT

We are thank [15-16]; which provide the data for public usage. We are also thankful to the National Natural Science Foundation of China under Grant No.61170203, 61170170, The National Key Technology Research and Development Program of China under Grant No.2012BAH33F04, Beijing Key Laboratory Program of China under Grant No.Z111101055281056, supports sharing with us.

REFERENCES

[1] M. K. Hsieh, B. Y. Chen, & M. Ouhyoung, "Motion retargeting and transition in different articulated figures", Proceedings of the 9th International Conference on Computer Aided Design and Computer Graphics, 2005, Hong Kong, China, pp. 6-pp.

[2] Yang, G. H., Won, J., & Ryu, S,"Online retargeting for multi-lateral teleportation", 10th International Conference on In Ubiquitous Robots and Ambient Intelligence (URAI), 2013.

[3] W.Gao, Y. Chen, G. Fang, C.Yang, D. Jiang, C. Ge, & C.,Wang, "A Motion Retargeting Method for Topologically Different Characters", Proceedings of the sixth International Conference in Computer Graphics, Imaging and Visualization, 2009, Tianjin, China, pp. 96-100.

[4] J. Tilmanne, R. Sebbe, & T. Dutoit, "A database for stylistic human gait modeling and synthesis", Proceedings of the Workshop on Multimodal Interfaces, 2008, Paris, France.

[5] X. Xu, C. Leng, & Z. Wu, "Rapid 3D Human Modeling and Animation Based on Sketch and Motion Database", Proceedings Workshop on Digital Media and Digital Content Management, 2011, Hangzhou, Zhejiang, China, pp. 121-124.

[6] S. Ali, Z. Mingquan, W. Zhongke, A. Razzaq, M. Hamada, & H. Ahmed, "Comprehensive use of Hip Joint in Gender Identification Using 3-Dimension Data", TELKOMNIKA Indonesian Journal of Electrical Engineering, 2013, 11.6, 2933-2941.

[7] Meredith M., and Maddock S., "Motion capture file formats explained," in Proceedings of Technical Report of Department of Computer Science, University of Sheffield, 2001

[8] The explain the information of c3d file format www.c3d.org

[9] Jeroen J.A and Van Boxtel "A biological motion toolbox for reading displaying, and manipulating motion capture data in research settings," Journal of Vision, 2013, 13(12). pp 1-16.

[10] Kleiner, Mario, et al. "What's new in Psychtoolbox-3." Perception 36.14 (2007): 1-1.

[11] C. Verron, Traitement et Visualisation de Vonnées Gestuelles Captées par Optotrak. IDMIL Report, 2005.

[12] Matlab platform for precompiled Matlab toolbox: www.mathworks.com, accessed 2014

[13] Burger, Birgitta, and Petri Toiviainen. "MoCap Toolbox-A Matlab toolbox for computational analysis of movement data." Proceedings of the Sound and Music Computing Conference 2013, SMC 2013, Logos Verlag Berlin.

[14] Mocap systems http://www.qualisys.com/ , accessed 2014.

[15] Motion capture lab (Advance Computing Centre for the Arts and Design) 2014. http://accad.osu.edu/research/mocap/mocapdata.htm

[16] Carnegie Mellon University Graphics Lab. CMU Motion Capture Database, 2014. mocap.cs.cmu.edu.

[17] MocapToolbox136, http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/mocap/

[18] RoboticsToolbox www.petercorke.com/Robotics_Toolbox.html.

[19] Mitchell, T. Machine Learning; McGraw Hill: Columbus, OH, USA, 1997.

[20] Marker placement guide on human body http://users.aber.ac.uk/hoh/CS390/512ViconSWManual.pdf.

[21] M. Yeadon and M. Morlock, "The appropriate use of regression equations for the estimation of segmental inertia parameters." Journal of Biomechanics, vol. 22, no. 6-7, pp. 683–689, 1989.

[22] P. de Leva, "Adjustments to zatsiorsky-seluyanov's segment inertia parameters," Journal of Biomechanics, vol. 29, no. 9, pp. 1223–1230, 1996.

[23] D. Pearsall and J. Reid, "Inertial properties of the human trunk of males determined from magnetic resonunce imaging," Annals of BiomedicalEngineering, vol. 22, pp. 692–706, 1994.

[24] J. Durkin and J. Dowling, "Analysis of body segment parameter differences between four human populations and the estimation errors of four popular mathematical models," Biomechanical Engineering, vol. 125, no. 4, pp. 515–522, 2003.

[25] C. Cheng, H. Chen, C. Chen, C. Chen, and C. Chen, "Segment inertial properties of Chinese adults determined from magnetic resonance imaging," Clinical Biomechanics, vol. 15, no. 8, pp. 559–566, 2000.

[26] D. Pearsall, J. Reid, and L. Livingston, "Segment inertial parameters of the human trunk as determined from computed tomography," Annals of Biomedical Engineering, vol. 24, pp. 198–210, 1996.

[27] V. M. Zatsiorsky and V. M. Seluyanov, "The mass and inertia characteristics of the main segments of the human body," Biomechanics VIIIB (Edited by Matsui, H. and Kobayashi, K.), pp. 1152–1159, 1983.

[28] J. Durkin, J. Dowling, and D. Andrews, "The measurement of body segment inertial parameters using dual energy x-ray absorptiometry,"Journal of Biomechanics, vol. 35, no. 12, pp. 1575–1580, 2002.

[29] P. L. Davidson, S. J. Wilson, B. D. Wilson, and D. J. Chalmers, "Estimating subject-specific body segment parameters using a 3-dimensional modeller program," Journal of Biomechanics, vol. 41, no. 16, pp. 3506– 3510, 2008.

[30] J. Wicke, G. Dumas, and P. Costigan, "A comparison between a new model and current models for estimating trunk segment inertial parameters," Journal of Biomechanics, vol. 41, no. 4, pp. 861–867, 2008.

[31] J. A. Reinbolt, R. T. Haftka, T. L. Chmielewski, and B. J. Fregly, "Are patient-specific joint and inertial parameters necessary for accurate inverse dynamics analyses of gait?" IEEE Transactions on Biomedical Engineering, vol. 54, no. 5, pp. 782–793, May 2007.

[32] www.health.uottawa.ca/biomech/csb/Archives/ dempster.pdf.

[33] De leva,"Adjustments to zatsiorsky-seluyanov's segmentinertiac parameters", Biomechanice, vol.29, Page.1223-1230, 1996.

[34] The detail of human body (BSP) computation "http://oregonstate.edu/instruct/exss323/CM_Lab/Center%20of%20Mass.htm"

APPENDIX:

A. Markers Configuration and BSP Computation

1) Source Code for retrieving the maker index

We edited the mcreadc3d () function as follow

```
d= mcreadc3d (fn);
 % Here fn is file name of the data

       [poss  jointj] = size(d.markerName)

        % This is for L_finger-----------1

        for labelInd=1: poss %- This is for L_finger

       if(strcmpi(d.markerName{labelInd,1},’liu:LFIN’)
       ||strcmpi(d.markerName{labelInd,1},’steve:LFIN’
       ||strcmpi(d.markerName{labelInd,1},’TakeoMonday
       :LFIN’)||strcmpi(d.markerName{labelInd,1},’Just
       in:LFIN’)||strcmpi(d.markerName{labelInd,1},’Ta
       iChi:LFIN’)||strcmpi(d.markerName{labelInd,1},’
       Style1:LFIN’)||strcmpi(d.markerName{labelInd,1}
       ,’Style2:LFIN’)||strcmpi(d.markerName{labelInd,
       1},’Style4:LFIN’)||strcmpi(d.markerName{labelIn
```

```
        d,1},’Style7:LFIN’)||strcmpi(d.markerName{label
        Ind,1},’rory6:LFIN’)||strcmpi(d.markerName{labe
        lInd,1},’Female1:LFIN’)||strcmpi(d.markerName{l
        abelInd,1},’male2:LFIN’))
        dataindex_1=labelInd  end
 end
```

2) Source Code for handling extra markers index

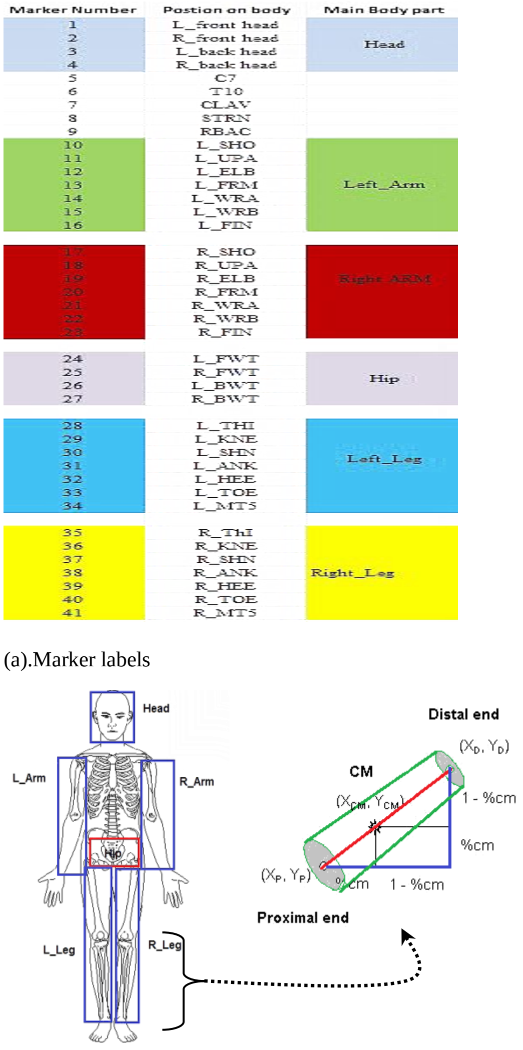
We add the following code to **mcreadc3d()function** and **manage** the extras marker index during reading c3d files, those were stored during motion capture session

```
         [Pose  numjoints]=size(d.data);
          numjoints_maker =numjoints;
         total_joints= numjoints_maker/3
        if total_joints > 41_use_able_joints
        Extrajoints=total_joints - 41_use_able_joint
        set_extract_joints_zerosuse=3*Extrajoints;
           tempmatrix=zeros(1,
           set_extract_joints_zerosuse)
        save(’tempmatrix’)
         end
        if numjoints/3==41 % This is for 41 markers or
more
        for pose = 1: size(d.data(:,:),1)
        tem_data = reshape (d.data (pose,:),3,41)’;
        d.data(pose,:)=[tem_data(dataindex_1,:),tem_dat
        a(dataindex_2,:),tem_data(dataindex_3,:),tem_da
        ta(dataindex_4,:),tem_data(dataindex_5,:),tem_d
        ata(dataindex_6,:),tem_data(dataindex_7,:),tem_
        data(dataindex_8,:),tem_data(dataindex_9,:),tem
        _data(dataindex_10,:),tem_data(dataindex_11,:),
        tem_data(dataindex_12,:),tem_data(dataindex_13,
        :),tem_data(dataindex_14,:),tem_data(dataindex_
        15,:),tem_data(dataindex_16,:),tem_data(dataind
        ex_17,:),tem_data(dataindex_18,:),tem_data(data
        index_19,:),tem_data(dataindex_20,:),tem_data(d
        ataindex_21,:),tem_data(dataindex_22,:),tem_dat
        a(dataindex_23,:),tem_data(dataindex_24,:),tem_
        data(dataindex_25,:),tem_data(dataindex_26,:),t
        em_data(dataindex_27,:),tem_data(dataindex_28,:
        ),tem_data(dataindex_29,:),tem_data(dataindex_3
        0,:),tem_data(dataindex_31,:),tem_data(datainde
        x_32,:),tem_data(dataindex_33,:),tem_data(datai
        ndex_34,:),tem_data(dataindex_35,:),tem_data(da
        taindex_36,:),tem_data(dataindex_37,:),tem_data
        (dataindex_38,:),tem_data(dataindex_39,:),tem_d
        ata(dataindex_40,:),tem_data(dataindex_41,:)];
end
elseif jsize>41 % set zero values to extra makers
for pose = 1: size(d.data(:,:),1)
 tem_data = reshape (d.data (pose, :), 3, jsize)’;
        d.data(pose,:)=[tem_data(dataindex_1,:),tem_dat
        a(dataindex_2,:),tem_data(dataindex_3,:),tem_da
        ta(dataindex_4,:),tem_data(dataindex_5,:),tem_d
        ata(dataindex_6,:),tem_data(dataind
        ex_7,:),tem_data(dataindex_8,:),tem_data(datain
        dex_9,:),tem_data(dataindex_10,:),tem_data(data
        index_11,:),tem_data(dataindex_12,:),tem_data(d
        ataindex_13,:),tem_data(dataindex_14,:),tem_dat
        a(dataindex_15,:),tem_data(dataindex_16,:),tem_
        data(dataindex_17,:),tem_data(dataindex_18,:),t
        em_data(dataindex_19,:),tem_data(dataindex_20,:
        ),tem_data(dataindex_21,:),tem_data(dataindex_2
        2,:),tem_data(dataindex_23,:),tem_data(datainde
        x_24,:),tem_data(dataindex_25,:),tem_data(datai
        ndex_26,:),tem_data(dataindex_27,:),tem_data(da
        taindex_28,:),tem_data(dataindex_29,:),tem_data
        (dataindex_30,:),tem_data(dataindex_31,:),tem_d
        ata(dataindex_32,:),tem_data(dataindex_33,:),te
        m_data(dataindex_34,:),tem_data(dataindex_35,:)
        ,tem_data(dataindex_36,:),tem_data(dataindex_37
        ,:),tem_data(dataindex_38,:),tem_data(dataindex
        _39,:),tem_data(dataindex_40,:),tem_data(datain
        dex_41,:),tempmatrix(1, :)] ;
end
```

3) Source code for normalized skeleton refinement between 41 markers (see Figure 4 & A.1 (a)) of mocap data (C3D).

```
m2jpar = mcinitm2jpar;
m2jpar.markerName{1} = ’root’;
m2jpar.markerNum{1} = [22:23 2 20];
m2jpar.markerName{2} = ’lhip’;
m2jpar.markerNum {2} = [23 2];
m2jpar.markerName{3} = ’lknee’;
m2jpar.markerNum{3} = [31 15 28];
m2jpar.markerName{4} = ’lankle’;
m2jpar.markerNum{4} = [34];
m2jpar.markerName{5} = ’ltoe’;
m2jpar.markerNum{5} = [32:34 36];
m2jpar.markerName{6} = ’rhip’;
m2jpar.markerNum{6} = [20 22];
m2jpar.markerName{7} = ’rknee’;
m2jpar.markerNum{7} = [35 37:38];
m2jpar.markerName{8} = ’rankle’;
m2jpar.markerNum{8} = [30];
m2jpar.markerName{9} = ’rtoe’;
m2jpar.markerNum{9} = [39:41 30];
m2jpar.markerName{10} = ’midtorso’;
m2jpar.markerNum{10} = [18 29 25 17];
m2jpar.markerName{11} = ’neck’;
m2jpar.markerNum{11} = [14 13 5 29];
m2jpar.markerName{12} = ’head’;
m2jpar.markerNum{12} = [8:9 12 16];
m2jpar.markerName{13} = ’lshoulder’;
m2jpar.markerNum{13} = [5];
m2jpar.markerName{14} = ’lelbow’;
```

```
m2jpar.markerNum{14} = [10 6];
m2jpar.markerName{15} = 'lwrist';
m2jpar.markerNum{15} = [26 1 27 3];
m2jpar.markerName{16} = 'lfinger';
m2jpar.markerNum{16} = [1];
m2jpar.markerName{17} = 'rshoulder';
m2jpar.markerNum{17} = [14];
```



Marker Number	Postion on body	Main Body part
1	L_front head	Head
2	R_front head	
3	L_back head	
4	R_back head	
5	C7	
6	T10	
7	CLAV	
8	STRN	
9	RBAC	
10	L_SHO	Left_Arm
11	L_UPA	
12	L_ELB	
13	L_FRM	
14	L_WRA	
15	L_WRB	
16	L_FIN	
17	R_SHO	Right ARM
18	R_UPA	
19	R_ELB	
20	R_FRM	
21	R_WRA	
22	R_WRB	
23	R_FIN	
24	L_FWT	Hip
25	R_FWT	
26	L_BWT	
27	R_BWT	
28	L_THI	Left_Leg
29	L_KNE	
30	L_SHN	
31	L_ANK	
32	L_HEE	
33	L_TOE	
34	L_MT5	
35	R_ThI	Right_Leg
36	R_KNE	
37	R_SHN	
38	R_ANK	
39	R_HEE	
40	R_TOE	
41	R_MT5	

(a).Marker labels

(b). *Human body segment* (c). *Segmental CM*

Fig. 15. A. Human Anatomical landmarks Template

4) *Computation of BSP*

The information of human body segments is used to determine the location of each segment's center of mass and each segment's mass. Each of these body segment parameters (BSP) usually expressed as a percentage value. Computing the center of each segment and each segment mass. It can be computed by the following equations:

$$x_{CM} = x_D(\%cm) + x_p(\%cm) \qquad A.1$$

$$Y_{CM} = Y_D(\%cm) + Y_p(\%cm) \qquad A.2$$

Where (XCM, YCM) = X & Y coordinates of the segmental CM, (XD, YD) = coordinates of the **Distal end** of the segment, (XP, YP) = coordinates of the **proximal end**, and %cm = CM.

In short, the general formulas for computing the position of each segment center of mass and each segment's mass:

$$x_{cm} = \frac{\sum_{s=1}^{n}\left(B_{seg_s} \times x_{seg_{cm\ s}}\right)}{B_{body}} \qquad A.3$$

$$Y_{cm} = \frac{\sum_{s=1}^{n}\left(B_{seg_s} \times Y_{seg_{cm\ s}}\right)}{B_{body}} \qquad A.4$$

Where B is the mass of the body, X, Y is a position coordinate and n is the number segments and each segment parameter (BSP) is routinely indicated as a percentage values.

The body center mass can be estimated from the CMs and the masses of the segments:

$$x = \frac{\sum_i (B_i * x_i)}{\sum_i B_i} \qquad A.5$$

$$Y = \frac{\sum_i (B_i * Y_i)}{\sum_i B_i} \qquad A.6$$

From the equations A.5 & A.6 describe as (X, Y) = coordinates of the body CM, i = segment number, (X_i, Y_i) = the X & Y coordinates of the CM of segment i, and B_i = mass of segment i. We can say in other words, the body CM coordinates are equal to the sum of the segmental mass products and segmental CM coordinates divided by the body mass ($\sum B_i$).