Innovative Framework for Human Motion Tracking for HCI

H.G.L. Sandaruwan, G.D.T. Kumara, H.G.R. Dileepa, C. Gamage, H. D. Gunaratna
Department of Computer Science and Engineering,
University of Moratuwa.
Moratuwa, Sri Lanka.

Abstract— **Human Computer Interaction is an emerging field in the computer science context, which tends to discover new systems that use user-centered view instead of machine centered view. This research specifically targets the full body motion interaction with computer. Therefore it develops an application which is more biased towards a user centered view. The process of building a new framework for HCI is challengeable and should be done very carefully. The system needs to indicate higher performance gain for measurements such as usability, efficiency, accuracy, and reliability. This paper describes a passive marker based approach which feeds input from web cameras and tracks blue colored markers. Input brings the marker position details with respect to time. Algorithms which regulate this framework are used for the calculation of derived attributes, clustering markers, and the prediction of positions coordinates of markers and initializing markers. They will give a meaningful interpretation to the marker position details. The next task of motion detection is facilitated using the Artificial Intelligence techniques which take preprocessed marker details as input. According to the prediction made, the user will be able to move in an immersive virtual environment. Although the project components were implemented in C++ java, they were integrated using java Native Interface (JNI).**

Keywords-HCI; motion tracking; passive markers; tracking algorithm; virtual environment

I. INTRODUCTION

Thisresearch project builds an innovative framework which facilitates human motion tracking using HCI techniques. For virtual reality and augmented reality applications it is often required that collect positions and orientations of real objects, then embed them in to a virtual or augmented environment. Performance of the framework mainly depends on the tracking of the markers and calculating of the derived variables like speed and directions of markers using their position details. Though there are lots of researches carried out on marker based human motion tracking, still it is an emerging field.

This paper introduces a passive marker system which can be used for full body motion tracking. It usesa robust color tracking based technique for motion capture and a real time color tracking so that user's interaction with the system is direct. Here it describes possible various solutions tested and the final solution. Here coloredmarkers on the human body are used to track motions.

This research project has image recognition and processing component which is developedusing Open CV, which is a C++ image processing library. But the last two components are implemented using Java. Those are neural network and Virtual Environment of the project. In the process of integration we had to come up with a mechanism to connect these two distinct code bases. After doing some research, we came up with Java Native Calles and use Java Native Interface to accomplish that. On this approach we have to call Dynamic Link Library (DLL) created using C++ code via JNI, to the java side.

There is no need of special cameras and sophisticated algorithms when markers are used. Though it requirescarefullycontrolledlight conditions, it reduces the complexity of the system. There are several researches [1]carried out to build passive marker systems to track human motions[2]. Major challenges faced while this research are to provide robust and accurate techniques for tracking color based markers, reduce errors caused by the noise, reduce marker occlusion as much as possible, assign initial ID's for markers, etc.

After the capturing coordinates, it finds a meaningful representation of the marker from those details. Fig.1 shows the abstract of operations of the system.

A survey which was conducted till 2005 [3] covers the aspects such as focusing body, gesture, gaze, and affective interaction and also discusses the audio interaction aspect. There is a great explanation on the classification which is an essential part of action recognition (they used an office meeting) in [4]. Survey [5] shows different approaches of the visual analyses of human action detection. A good example of model based human action recognition procedure is presented by [6]. They are discussed in the next section, with reference to our application with more papers and surveys in addition.

Model based approaches fit with marker concept very well. Limb model of human body presented by the [6] is a good structure of pose estimation and it is continued until 2008 [7]. Classification method that found at [4] is agood system which can be evolved as our application needs.

Predicting the missing markers is a major task in order to keep the accuracy high for the project. A real time approach for estimating missing markers based on previous positions have presented by two papers [9] and [10]. They have used two cameras for capturing and 3D information to predict the markers.Markers on a one limb can have a particular maximum distance between them. They have used that assumption as well when they build the system. Furthermore it also uses well known marker prediction filter called Kalman filter to prediction process [11].

Kalman filter uses last set of observations to predict the missing details. A paper which describes a method related to the problem of missing markers referenced by [12]. It takes advantage of a device called OptiTrack. It automatically computes the 3D skeleton and track motions from markers attached on the human body.

II. METHODOLOGY

Though there are several solutions available for real time human motion tracking, here we introduce inexpensive passive marker system for tracking human action. In this system it uses single web camera and markers set on user in order to track human actions.

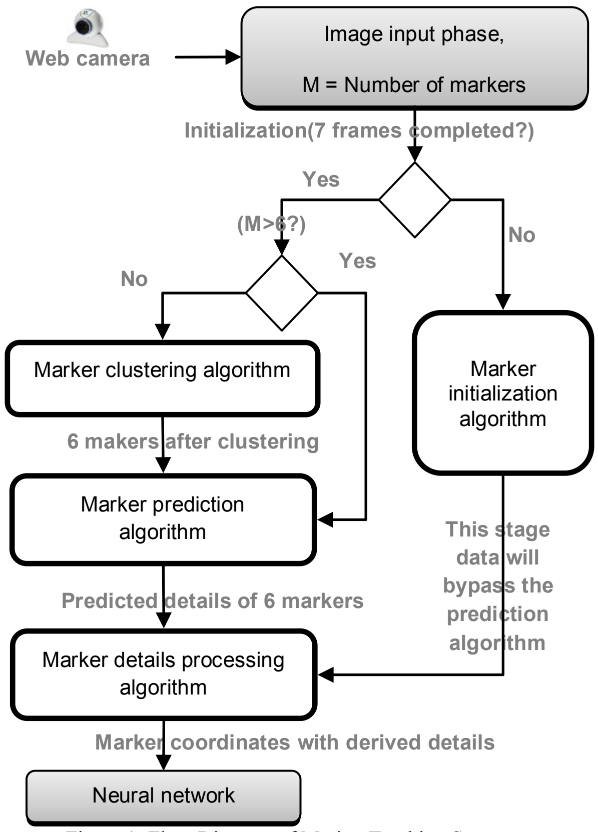
Web camera
Image input phase,
M = Number of markers
Initialization(7 frames completed?)
Yes
No
(M>6?)
Yes
No
Marker clustering algorithm
6 makers after clustering
Marker prediction algorithm
Marker initialization algorithm
This stage data will bypass the prediction algorithm
Predicted details of 6 markers
Marker details processing algorithm
Marker coordinates with derived details
Neural network

Figure 1: Flow Diagram of Motion Tracking System

A. *Select Suitable MarkerSet*

Markers are small objects, often reflective strips or balls placed at certain points of interest on user's body.Active markers, Passive markers [10] which can be used for marker based HCI systems. According to the requirements of the framework passive markers were chosen. When deciding type of markers mainly consider about, Easiness of use, Accuracy level, Cheapness. Simple color based marker set was selected in order to achieve above conditions.

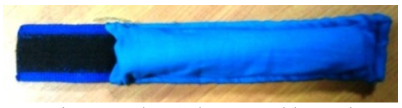

Figure2: Blue Color Wearable Marker

As shown in the Fig. 2 blue and red color markers were prepared and marker set is consists of 12 markers for the full body motion tracking. They are having 6 types of dimensions because markers are placed symmetrically on human subject. Thickness is 1.5cm and width is 7cm for all the markers but with several different lengths for each couple of markers. From user to user length required for marker is different therefore these markers are made such that lengths can be adjusted as it required. All marker sizes are illustrated in Fig.3.

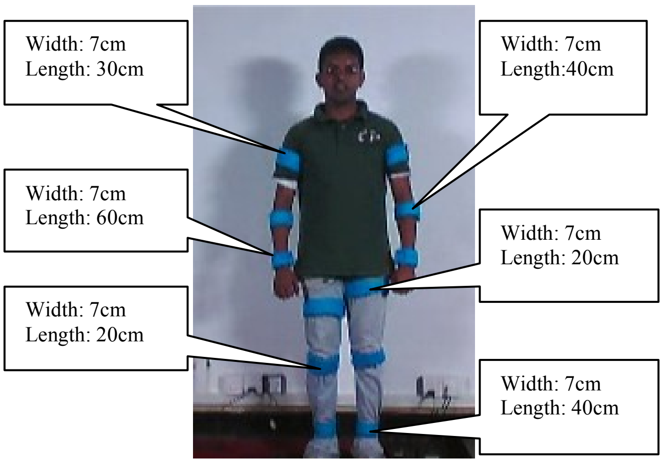

Figure 3: Dimensions of Markers

B. *Marker Setup on User*

Marker set up is important factor because it is necessary to reduce marker occlusion as much as possible to get more accurate results. Crucial markers are occluded or when the algorithm confuses the trajectory of one marker with that of another then the results may not be accurate because then it needs to predict the movements of the markers. That prediction may not give correct prediction and that leads to erroneous result.

The markers are usually attached directly to the skin/garment or they are attached to a user wearing a full body suit designed specifically for motion capture. When the user using this marker set he should not wear any blue color garments. The markers go on joints, but actually placing them on joints makes the data less meaningful, because it can't detect joint rotation. So the markers generally go between joints close to the bones in order to represent joints, like forearms, biceps, etc. InFig. 3 it is illustrated that how these markers are placed on user and this set up uses three markers for each arm and leg.

C. *Marker Tracking Paradigm*

Image analyzing phase of this system is responsible for, Capture video input and Get frames, Extract blue color pixels, Generate binary image, Remove noise and Filter by area, Identify markers and Get centroids.

The key difference is what will be doing with the data after captured them. This system needs cameras to track the markers placed on user. Because thissystem doesn't need any performance capturing like face, fingers and captures subtle expressions. Therefore it uses simple webcams in order to capture video stream and extract marker movement details to detect user's actions.

Major challenges Marker tracking should deal with are Fast (operates in real-time), Robust (track motion sufficiently reliably and accurately so as to enable a compelling interactive experience), Un-intrusive (not require a large amount of preparation before use), widely accessible (anyone can use it, regardless of shape, size, color, etc.), Cheap (require hardware that is easy to find and low cost), Entertaining or useful (consumers must want to use it), etc.

There are four main modules in marker tracking process. As depicted in Fig.4 all the modules are connected each other.

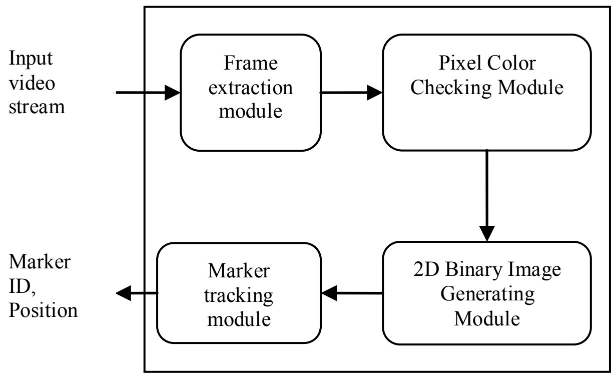

Figure 4: Marker Tracking Process Modules

D. *Algorithm for Calculation of Speed Acceleration and Direction*

Our system uses a neural network for prediction of action of the user depend on the input details of the marker we detect. As the image input phase capture the coordinates of the markers, we have the details for 6 markers in an array based data structure. But in order to get a sensible output we have to provide more details in to the neural network. Therefore our algorithms calculate the speed and direction of the system using marker coordinate details. As the neural network it particularly designed for our application require 4 details per marker, which are in details x and y coordinates, speed and direction.Calculation of speed is done using the average of speeds of recent 4 frames. System stores coordinates of 4 recent frames in array structure.

t <- 0.01 // time gap between two frames in seconds for 10 frames per second, 10/1000

For i: 1-> 4

SumofDistances<- sumofDistances+ distance from i th marker to i+1 th marker

Speed<-sumofDistances / (4 * t)

Direction of marker is represented by the angle which the markers direction made with the vertical axis and shown in the Fig.4.

TanTheata<- (x_2 - x_1) / (y_1 - y_2) //Current frame coordinates: (x_2, y_2)Last frame coordinates: (x_1, y_1)

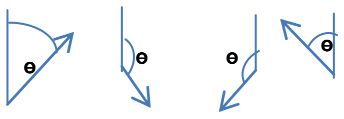

Figure4: Direction of Marker is Representedby theAngle

System understands the marker positions with derived details like speed and direction as it shown in the Fig.5 below.

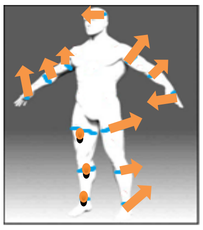

Figure5: Speed Vector Representation of Markers

E. *Algorithm for Clustering Markers*

When a human computer application is being run in real timethere can be noise introduced depend on the environment it use. Therefore we design an algorithm to remove noise marker form our input. This methodology uses Euclidian square distances for the clustering task. Then we cluster the markers which have smallest distance from one to other.

Euclidian square distance relevant to i th and j th marker = (x_i–x_j) * (x_i – x_j) + (y_i – y_j) * (y_i – y_j)

F. *Algorithm for Prediction of Positions Coordinates of Markers*

When this application runs in real time there can be missing markers in real time positions. Therefore we will have to predict the positions of those if we require a high accuracy output. This algorithm allows the system to find the position in a situation where some markers are missing with reference to initialized marker system. For the calculation of these components we use acceleration of the marker in recent frame and speed and direction which were discussed in the previous algorithm.

Kalman filter which discussed in the introduction uses motion equations to predict the markers [13]. Limb distance limitation method is not suitable for this approach as the prediction of the markers with this restriction will lead to huge error at some stage of the application.

It uses following simple motion equations to find the x coordinate and y coordinate for the next frame.

$v = u + a * t$

$s = u * t + 0.5 * a * t * t$

Where “v” is ending speed, “u” is stating speed, “t” is time moved, and “a” is acceleration.

-End speed = speed of recent frame + acceleration * time

-Distance = speed * t + 0.5 * acceleration * t * time

Then we calculate the components of x and y which should be added to the recent coordinates.

- For x coordinate,

```
IF (theta <180 )
x <- x + distance / (sqrt(1 + tan(thet) * tan(thet)))
ELSE IF(theta >= 180)
x <-x - distance / (sqrt(1 + tan(thet) * tan(thet)))
```

- For y coordinate,

```
IF (theta < 90 || theta >270)
y <- y - distance/100
ELSE IF(theta > 90 && theta < 270)
```

```
y <- y + distance /(100 *sqrt(1 + tan(thet) * tan(thet)))
//Thet is the angle we calculated before
```

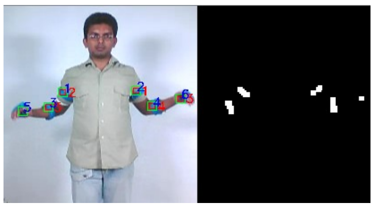

Figure 6: Marker Prediction

Marker prediction is shown in the Fig.6. It can be seen that 6 markers are represented using green color boxes. But red color boxes are the detected markers and only 5 of them are there. Identifier number 5 is the marker predicted by the algorithm of the system.

G. *Algorithm for Initializing Markers*

System has to assign the identifiers for So that the markers can be tracked continuously. This algorithm accomplishesthat task and helps the accurate tracking trough out the project. Markers are placed on user's body as shown in Fig.8 (left hand side figure) initially.

H. *Motion Detection using AI techniques*

Motion detection is another critical component within this framework. This unit is supposed to provide the intelligence for the system when do decision making. After a research carried on available methodologies against the features required for real time motion detection it is being concluded that Artificial Neural Networks is going to be the most appropriate solution.

An Artificial Neural Network (ANN) is an artificial representation of the human brain that tries to simulate its learning process. It is an adaptive system that changes its structure based on external or internal information that flows through the network. It is composed of a large number of a large number of highly interconnected processing elements which are called as neurons to solve specific problems [14].

This unit deduces the motion of the user where the input is taken from the "BubbleBoy Core" component and the final output is fed to the "Virtual Environment" component. That means even though how accurate your image processing information taken from "BubbleBoy Core" component and how attractive your virtual environment if the predicted motion is not harmonized with the actual motion of the user the purpose of the system is lost.

I. *System Integration*

It has a major part of project which is written in C++ and runs separately from the components written in Java. Therefore it is required to create a class to call C++ methods, in the java program and define what native methods are in the DLL. To call this upon from the Java virtual machine we need an interface to call particular method from C++.

Java Native Interface (JNI) is a fully java written interface which runs on the java virtual machine and enables writing java native methods [15 and 16]. This special library enables the binary compatibility of native method libraries among given platforms [17].

III. RESULTS AND DISCUSSION

BubbleBoy framework is designed for building a low cost, consistent and user centered application for human action recognition and interpretation. This section discusses the challenges and performances faced when developing this system.

Initial task was to find most appropriate marker set for tracking full body motion. Most of the previous researches had used red color markers.

A. *Marker Color Selection*

Initially we chose red color markers because of the properties that red color markerpossessed and also red is one of major color which helps when it's come to image processing and which can track easily when it's come to color based tracking.

But when the red color is being used for markers caused erroneous results because it tracks human skin in some illumination conditions.Similarity of the red color and the human skin is the reason for this effect.

That can be controlled in several ways, by changing threshold value for red color but for accurate marker tracking; it needs to be adjusted manually or we can enforce restrictions to dress which covers up entire body so that the skin will not be detected with markers.Tracking a particular color is done using RGB restriction. Putting a more specified RGB values to tracking condition we can contrast the human skin from red color successfully.

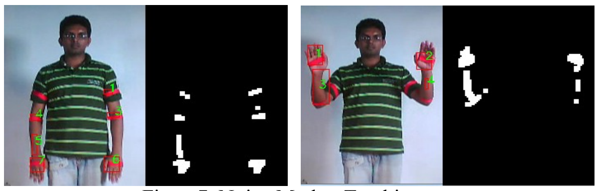
Figure7: Noisy Marker Tracking

Here it only used 6 markers in order to track the rowingaction. For the full body motion tracking we can use other markers and capture the marker coordinates to track user's action.

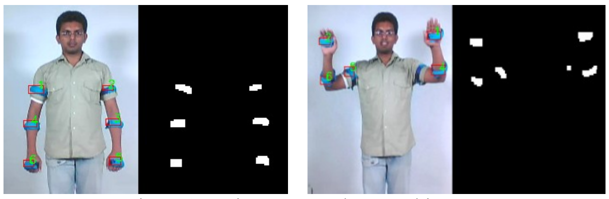
Figure 8: Noise Free Marker Tracking

After changing the marker color into blue, marker tracking gave results with some accuracy but the level of accuracy is not high. Then we used two color (blue and red) markers set to improve the accuracy level of results.Multiple color marker tracking can be used to give more accurate details because using multiple color we can predict marker movements with higher accuracy.

Problems arose when we are using the red markers are solved after practically testing.

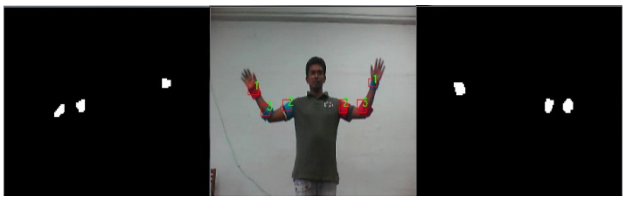
Figure 9: Two Color Marker Tracking

B. Performance Analysis

Experimental results of the algorithms are shown in Table I. Algorithm for calculation of derived attributes took the highest average of time as it processes large amount of details. It is tested with real time data and demonstrated a higher accuracy. Others are quite light whencompared with algorithm for calculation of derived attributes. Qualities of them are discussed in the table below.

TABLE I. ALGORITHM PERFORMANCE ANALYSIS

Algorithm	Average time	Quality of the Output
Calculation of speed acceleration and direction	0.014	High accuracy (95%)
Clustering markers	0.004	Good (70%)
Prediction of positions coordinates of markers	0.001	Good (80%)
Initializing markers	0.001	High accuracy (100%)

All These comparative performance analysis are carried out based on speed, time requirements and accuracy. These algorithms were implemented in same programming environment and compiler: C++ and Visual Studio 2010, computer vision library: OpenCV 2.1, Operating System: Windows 7 Ultimate 32 bit, Configuration of the tested Machine: Intel(R) Core(TM) 2 Duo CPU T7250 at 2.00GHz core, 2GB RAM are the tools and system specifications used.

C. Future Work

This project can be extended to use multiple cameras to track 3D details of user to track more details. Using multiple colors improve accuracy of marker details. Adjustments can be done for the framework as it automatically adapts to the illumination changes.Adding more actions and train framework according to the actions is possible. Intel TBB can be used to improve running time and performance.Adding higher degree of environment talk back to the project is possible as a future work. More advanced graphic development tools may help creating better virtual environments.

REFERENCES

[1] S.E. Slawson, P.P. Conway, L.M. Justham, and A.A. West, “The development of an inexpensive passive marker system for the analysis of starts and turns in swimming,” *Procedia Engineering*, vol. 2, 2010, p. 2727–2733

[2] A. Kolahi, M. Hoviattalab, T. Rezaeian, M. Alizadeh, M. Bostan, and H. Mokhtarzadeh, “Design of a marker-based human motion tracking system,” *Biomedical Signal Processing and Control*, vol. 2, 2007, p. 59–67.

[3] A. Jaimes and N. Sebe, “Multimodal Human Computer Interaction: A Survey,” *IEEE International Workshop on Human Computer Interaction in conjunction with ICCV 2005*, Oct. 2005.

[4] A. Hakeem and M. Shah, “Ontology and Taxonomy Collaborated Framework for Meeting Classification,” *17th conference of the International Conference on Pattern Recognition, ICPR*, 2004.

[5] D.M. Gavrila, “The Visual Analysis of Human Movement: A Survey,” *Computer Vision and Image Understanding*, 1997, pp. Computer Vision and Image Understanding.

[6] L. Sigal and M. Black, “Predicting 3d people from 2d pictures,” *Articulated Motion and Deformable Objects*, 2006, p. 185–195.

[7] L. Sigal, *Continuous-state GraphicalModels for Object Localization, PoseEstimation and Tracking*, Department of Computer Science at Brown University, 2008.

[8] P. Rai, K. Tiwari, P. Guha, and A. Mukerjee, “A cost-effective multiple camera vision system using FireWire cameras and software synchronization,” *Int. Conf. High Performance Computing*, 2004, p. 17–20.

[9] A. Aristidou, J. Cameron, and J. Lasenby, “Predicting missing markers to drive real-time centre of rotation estimation,” *Articulated Motion and Deformable Objects*, pp. 238–247, 2008.

[10] A. Aristidou, J. Cameron, and J. Lasenby, “Real-time estimation of missing markers in human motion capture,” in *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on*, pp. 1343–1346.

[11] R. E. Kalman and others, “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[12] Q. Wu and P. Boulanger, “Real-Time Estimation of Missing Markers for Reconstruction of Human Motion,” in *Virtual Reality (SVR), 2011 XIII Symposium on*, 2011, pp. 161–168.

[13] N. Funk “A Study of the Kalman Filter applied to Visual Tracking”. *University of Albert*a ,2003. pp.*652*, 1-26. Available at: http://www.njfunk.com/ research /courses/652-probability-report.pdf.

[14] G. P. Zhang, “Neural networks for classification: a survey,” Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 30, no. 4, pp. 451–462, 2000

[15] Bothner P. (2009, November) Java/C++ integration. [Online].Available: http://gcc.gnu.org/java/papers/native++.html [Accessed: 10 Mar.2011]

[16] Gabrilovich E and Finkelstein L, "JNI – C++ integration made easy," *C/C++ Users Journal*, vol. 19, no. 1, pp. 10-21, January 2001.

[17] Oracle Corporation. (2010) Chapter 5: JNI Technology. [Online].Available: http://java.sun.com/developer/onlineTraining/Programming/JDCBook/jni.html[Accessed: 3 Dec. 2010]