Wii-trieve: Retrieving Motion Sequences Using Acceleration Data of the Wii Controllers

Yekaterina Kharitonova
University of Hawaii, Hilo

Abstract

This paper introduces a novel approach for retrieving user-specified motions using the acceleration information obtained from Wii controllers. The major focus of this work is to develop a system, which would help professional animators as well as common users to extract a specified motion fragment from a motion capture database using the acceleration information obtained from the Wii controllers.

1. Introduction

If a user searches for walking motion in the CMU Graphics Lab Motion Capture Database[1] they will find 144 subjects with an average 5 to 8 motions per subject. Which one of the files is best suited for the animation they had in mind? We propose a system designed to help with this decision. The user acts out a motion with the Wii controllers attached to the body, and then a motion file from the motion capture database is suggested based on how closely it matches the acceleration pattern of the user's input motion.

The ability to extract full-body motion, which strictly agrees with the motion given by the user, can benefit animation, movie and game creators. Motion capture is the dominant system for capturing human motion by means of optical detection of strategically placed retro-reflective markers [Liu 2006]. This study was conducted at Carnegie Mellon University, where the motion is captured using the Vicon iQ software. The data is recorded into the ASF file (Acclaim Skeleton File) and the AMC file (Acclaim Motion Capture data), the file containing the movement data.

This research project's focus was on using the features of the Wii controllers to assist with motion sequence retrieval. The motion of the Wii controller is sensed by a 3-axis linear accelerometer. Inside the chip is a small micro mechanical structure which is supported by springs built out of silicon. Differential capacitance measurements allow the net displacement of the tiny mass to be converted to a voltage, which is then digitized. The sensor does not measure the acceleration of the controller, but rather the force exerted by the test mass on its supporting springs. [Wii Linux 2007]

Figure 1: Wii controllers

This research is an ongoing project and this paper's aim is to summarize the experimental efforts made over the summer of 2007 in capturing information from Wii controllers and building a system, which would allow us to compare the Wii controllers' data to motion capture data.

[1] http://mocap.cs.cmu.edu

2. Background

The main objective of this project is to follow Jinxiang Chai and Jessica Hodgins' paper [2005] on creating full-body animation using low-dimensional control signals. With a small set of markers (about 6-9) and only two video cameras they were able to construct a graph of nearest neighbors for a fast search of the motion examples. This work aims to reproduce the experiment using the Wii controllers for the low-dimensional control signals instead of the markers.

There has been extensive research on reconstructing full body motion using low-dimensional control signals. Guodong Liu and his colleagues [2006] estimated human motions from a small set of the most informative markers using the Principal Components Analysis (PCA) based Principal Feature Analysis method. Another work by KangKang Yin and Dinesh K. Pai [2003] featured a system, which allows interactive control of the avatars by extracting full body motions from a motion capture database using a foot pressure sensor pad. The main aspiration of these systems is to reduce or eliminate the use of costly and complicated motion capture systems, thus, providing the user with simple and intuitive control of avatars, virtual reality or performance-driven animation.

3. System Overview

The goal of the presented system is to record motion using a motion capture system while simultaneously capturing acceleration information from Wii controllers. After converting the motion capture's position information into acceleration and comparing the corresponding values to each other, the system retrieves motion clips that are similar to the queried fragment.

At this point of our research, we are able to collect acceleration data and have tested the system by searching for motion fragments in a small database of acceleration files.

3.1. Recording the Acceleration Data

The subject's motion was recorded and processed using the Vicon motion capture system and the Vicon iQ software. Simultaneously, acceleration information from the Wii controllers was captured. The system employed five controllers connected to the PC via a Bluetooth dongle. Two of the controllers were held by the subject (one in each hand), the other two controllers were fixed onto the legs, with the remaining controller mounted in the area of the solar plexus. To record and store the data from the controllers, the presented model uses cWiimote 0.2: Windows API for Wii controllers written by Kevin Forbes [2007].

Figure 2: Experimental placement of the retro-reflective markers and Wii controllers

Using the buttons of the left-hand controller the subject of the experiment would start and pause the recording of the acceleration data. Each acceleration record in the data file is time-stamped. Acceleration data consists of readings from the x, y and z-axis of the instantaneous force applied to the controller by the subject who is holding it.

3.2. Searching for Closest Matches

To search a database for the specified fragment of motion, a window-matching technique is introduced. This method permits storing a fragment of the original motion and then looking for a similar acceleration pattern in other files. The source sequence is stored into a vector with the length *max*, where *max* is a user-defined constant, which specifies how many frames of acceleration data are stored. The user selects a starting time of the motion and the system then loads the specified fragment into the vector. For the conducted experiments *max* was long enough to include a fragment of distinct motion, such as a step, clap or hop, which on average included 40-60 frames at a frame rate of about 40 frames per second.

Another vector is used as a window, which slides over the experimental file. Once the data from the beginning of the file is stored into the experimental sequence each frame of the source data is compared to that of the experimental one. The difference between each controller's data from each of the axes is calculated using the sum of squared distances (SSD). As the window scans through the experimental file the system keeps track of the location of the interval where the smallest sum has occurred. *K* closest matches are saved and extracted into a separate file.

Algorithm: **Window-matching technique**

Require: A and B are the same *max* length; specify the experimental input file in which to look for the specified fragment.

Store "ground truth" acceleration into vector A
Store *max* number of frames into vector B (from the beginning of the input file)
while (did not reach the end of the input file)
do
  Calculate SSD between A and B
    if (new SSD is smaller than the minimum SSD of all K matches) **then**
      if (there are already K matches stored) **then**
        Get rid of the largest SSD
      end if
    Add the current fragment to the K matches
  Discard the first frame stored in B
  Get the next frame from the acceleration file
  Insert the next frame at the back of B
end while

3.3. Extracting Closest Matches

Once the information about the *K* closest matches has been stored into a vector, the system goes through it and extracts the acceleration data associated with each interval. We now have a separate file that only includes the acceleration data for each of the matches.

To help visualize the results a subroutine was written to extract the acceleration data for the specified body part from the given axis.

4. Results

We are currently in the process of testing the system to identify the areas, which need to be improved. Thus far in the study we have run tests to determine whether the acceleration patterns of different motions are in fact discernible. The concern was that the system would not be able to discriminate between the walking and running motions, which are somewhat similar.

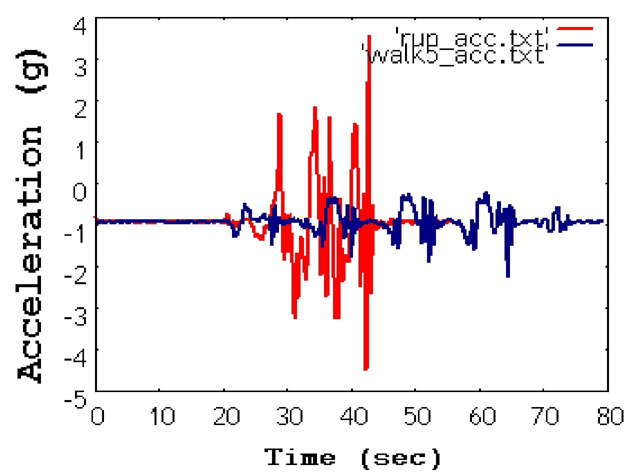

Figure 3: Walking motion (blue graph) versus running motion (red graph)

As shown in Figure 3 the running motion has a much bigger variation than the walking motion does. Once the distinction between the graphs of different motions was evident, we tested the matching procedure by selecting a fragment of motion from one acceleration data file and looked for an analogous pattern in another data file.

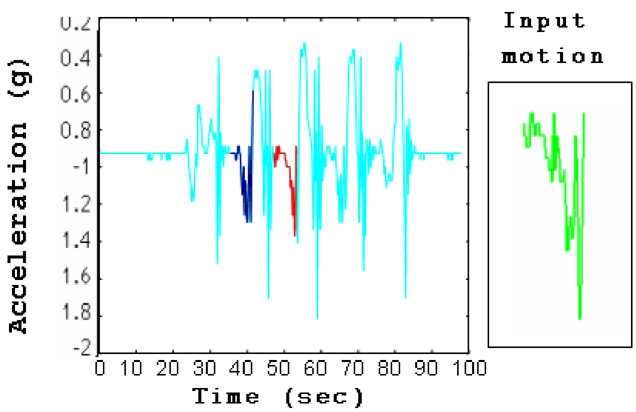

Figure 4: Input motion (acceleration of the left leg on y-axis) and the two closest matches

For one of the tests a step motion was searched in the walking and running motion files. The system was able to correctly identify walking segments in other walking files. The system also was unable to find the correct matches for a walking segment in data files, which included running motions. This is promising because now we know that at least for simple cases the system produces the correct results and is able to distinguish between two types of relatively similar motions.

5. Discussion and Future Work

In our preliminary steps we have tested the system and received successful results for the chosen test cases. We have not assessed how accurate the system is for longer fragments of motion or how well it performs on a larger number of files.

One of the major goals now is to collect more acceleration and motion capture data to expand our database. The next step would be to make the system faster and easier to use: the program should be adapted to search through multiple files in one run.

At this point in our research we have not yet used motion capture data and local modeling. Our next step would be to retrieve motion sequences using the acceleration data from the Wii controllers by combining our current system with the system proposed in Jinxiang Chai and Jessica Hodgins' paper [2005]. Once we are assured that the system produces the correct and effective results we may extend our work to use the retrieved results to reconstruct a new motion just as Jinxiang Chai and Jessica Hodgins did in their paper.

Acknowledgment

This work was supported by the Computing Research Association's Distributed Mentor Project (CRA DMP). This project was conducted in collaboration with Jessica Hodgins, Takaaki Shiratori and James Andrews (Carnegie Mellon University). The author would like to thank them for their expertise and advice on designing the project.

References

Chai, J., and Hodgins, J. 2005. Performance animation from low- dimensional control signals. In ACM Transactions on Graphics. 24(3): 686–696.

Forbes, K. Retrieved from Simulated Comic Blog http://simulatedcomicproduct.com/2006/12/cwiimote-02.php on June 2007.

Liu, D., Zhang, J., Wang, W., and McMillan, L. 2006. Human motion estimation from a reduced marker set. In Proceedings of the 2006 symposium on Interactive 3D graphics and games I3D '06. 32-42

Wii Linux, 2007 http://www.WiiLi.org

Yin, K., Pai D.K. 2003. FootSee: an Interactive Animation System. In Proceedings of the 2003 Eurographics/SIGGRAPH Symposium on Computer Animation. 329-338