

Teaching Tennis In Virtual Environment

Vladimir Stepan¹, Jiri Zara²

Czech Technical Univeristy, Department of Computer Science and Engineering

Abstract

This paper deals with the problems of human animation and motion analysis, particularly in the area of methodological applications in sports training. We describe a particular program tool that has been implemented as a Java applet connected to the VRML browser through the Java EAI interface. Various aspects of this solution are discussed, as well as motion analysis and its significance for sport oriented applications.

Keywords: VRML, H-Anim, human animation, motion analysis, tennis

1 Introduction

The virtual reality and especially virtual human simulation can successfully be used in educational applications, and also in the area of physical education and sports training. The virtual reality means can provide the best demonstration and feedback tools that are available so far. And with the VRML standard for a virtual reality on the Internet, such tools can get easily accessible to arbitrary user

The goal of our project was to create a software tool able to demonstrate and analyze motions of a real human in order to assist to a tennis teacher. It should be able to use the advantages that make the virtual reality superior to the means of methodological demonstration that had been used so far, such as the videotape. The user should be able to change the point of view and gain additional information and visualization from the recorded and viewed motion.

Besides viewing the animation from various points of view there is a few motion analysis experiments implemented in the applet. The goal of these is to gain additional information about the behavior of the body while performing the motion and about the motion itself. One direction of these experiments is the visualization of the changes of angle in particular joints. The other is the analysis of the course of tennis player's racket, focused on finding the right spot to strike the ball, if it exists.

The first part of this paper defines the sport training application and lists several problems that may be encountered while creating it. The second part is a

description of the solutions used in our tennis training application.

2 Theoretical Background

The sport training application must be able to show a motion performed by an athlete and provide detailed information about it. Important part of that information is related to the activity of particular body parts. The rest of the required information would be "sport-specific". A tennis training application would deal with the animations of tennis strokes and it would inform the user of the impact of the racket on the ball.

2.1 Humanoid Body

The analyzed motion is best to show by animating a virtual model of human body.

The use in discussed type of application requires that the model of humanoid body implements a set of joints that would allow to perform realistic copy of the real person's move. The intended use of the body allows to use a very limited set of joints. It is not necessary to support the hand and finger or foot and toes joints, as well as many of the vertebrae, because these joints have no significance in the animated moves or their roles can be substituted by another joints.

Good research has been done on the field of virtual humanoids [1][2][3][4], to discuss it in detail, however is not the topic of this paper.

2.2 The Animation

The term animation in this context means the data that describe the motion and are used to animate the humanoid body.

The animation data can be artificial as a product of an animation software – key-frame animation. Or it can be the result of recording the real motion using the motion capturing technology – recorded animation.

The first approach produces the animation that can serve as an example for demonstration of goods or wrongs performed during the motion. Motion capturing does the same, but is better, because it can provide the actual feedback as it enables the trainees to see their own

¹ stepanv@cslab.felk.cvut.cz

² zara@fel.cvut.cz

moves. On the other hand, Motion capture systems are expensive and it is not quite possible to use them in the real situation such as an ordinary tennis lesson.

In either case the animation data is a description of positions of selected points in space during the motion. It is useful when the sport training application is able to work with both. It is just the matter of transforming the input animation data to a format specific to the application.

The sport training application works with the library of animations that are available for viewing and analysis. This library may be open to adding or removing according to the teacher's intention. An animation selected from the library is connected to the humanoid body.

2.2.1 Animation Libraries

An animation can be stored in the same file with the definitions of the humanoid. This approach is good enough for animated figures such as an avatar for the multi-user virtual environments, but proves not suitable when it is required to often alter the library.

Animations should be stored in one file each separated also from the humanoid definition. This solution is also good for an Internet application because it does not require to transfer large amount of data at once. Although animation is to be stored separately it is recommended to use it with the model it was created for. In combination with different model, special treatment is necessary.

Animation library can be based on an animation prototype that includes a specialized field for the references to the animated humanoids and a script for connecting the interpolators to the joints of the humanoid [5]. This approach allows to create a library for general use. An application chooses the animation and the humanoids to be animated and fills in the appropriate field of an animation prototype. The result is that one or more humanoid bodies are driven by the data of the selected animation.

For the tennis training application this approach is not necessary, since in case of tennis we don't need to animate more than one body.

2.2.2 The Use of an Animation

Animation serves as a mean to view and trace motion. For this purpose the sport training application must implement the functions of starting, stopping and pausing the animation and the speed of animation must be adjustable. Also the function of frame-by-frame tracing should be implemented.

The timeline with the progress indication should also be present to connect the viewing of animation with the results of additional analysis.

2.3 Motion Analysis

There are two types of information we can get by analyzing the tennis stroke motion. The first is related to the behavior of the body and its parts when performing the move. The second concerns with the quality of the stroke and is based on tracking the course of the racket.

2.3.1 Joint Activity Analysis

Joint activity analysis helps to show the behavior of selected joint during the motion. One of the ways to visualize the time dependent changes of the joint rotation is the animation itself. We want to discuss the means of additional visualization for better understanding.

The main problem is how to visualize the changes of joint rotation. Generally there are two types of joints in a human body: cylindrical with one degree of freedom and spherical with more (three) degrees of freedom.

To visualize the activity of the cylindrical joints we can just use the values of the angle for graphic representation of the relationship between the angle and the time

Similar situation occurs when we need to analyze the special motion, during which all involved joints act as the cylindrical joints and their axis of rotation doesn't change or the changes of it can be neglected.

The example of this can be the gait analysis [8], where the joints of human legs are observed. Although the hips are the typical example of a spherical joint (joint with 3 degrees of freedom), its axis of rotation does not change much during the usual gait motion.

Usually the moves of spherical joints include more degrees of freedom, the axis is changing as well as the angle. That is the case of tennis strokes. We had to choose different approach here.

According to Euler's rotation theorem, *any rotation may be described using three angles* that are called *Euler angles*. Using the three Euler angles instead of one value of rotation around the arbitrary axis can give the distinct graphical description of the tennis stroke.

2.3.2 The Course of the Racket

When the joint activity analysis shows what the body does, the next step is to find out how it influences the tennis ball.

The understanding of the impact of the racket on the ball is very important in teaching tennis. The analysis here begins with the resolving if the motion is able to hit the ball and send it to specified direction. That means, the racket moves in the specified direction for some time during the motion.

The next question will be, how long the racket holds on to that direction, how long is the ball under its direct influence.

Finally we can inquire the relationship between the racket's normal and velocity vector during the time it influences the ball. All these factors determine the trajectory of the ball after it leaves the racket.

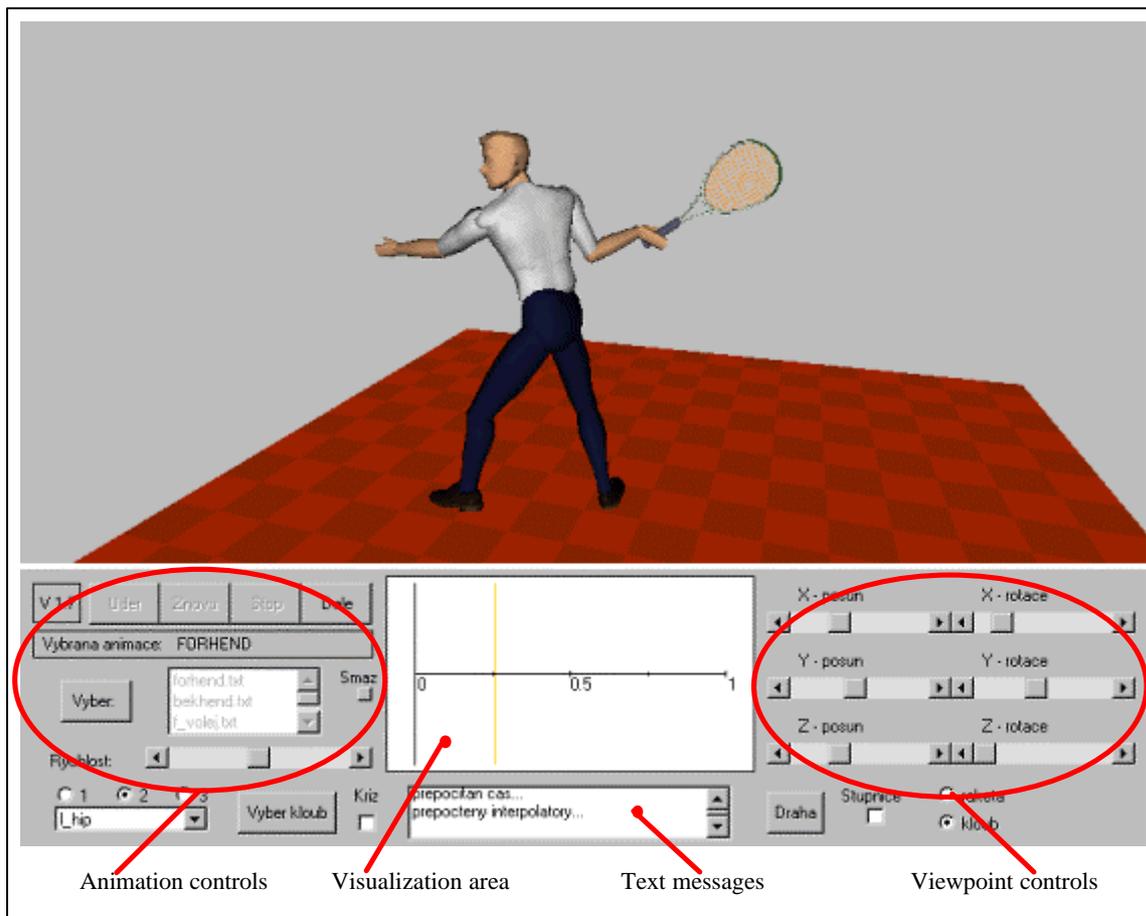


Figure 1: The user interface

In order to obtain this information we have to find the trajectory of the racket.

3 Implementation

The program that has been implemented is Java applet connected to a VRML scene through a Java External Authoring Interface (EAI) as seen in Figure 2.

The VRML file used by the applet is simple as it contains only the static definition of the H-Anim 1.1 compliant humanoid body. All other VRML elements, such as the interpolators are dynamically added or removed.

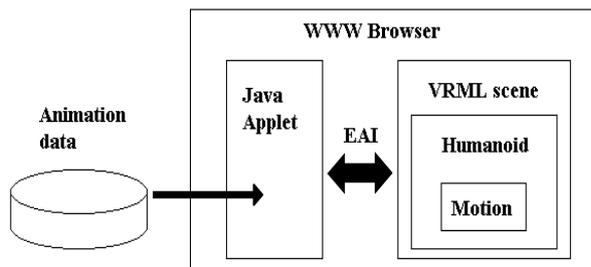


Figure 2: The architecture of the program

The animation data has been stored separately in text files. The information about the animation files available is in a special index file that is given to the applet as a parameter.

The applet allows to start, stop and pause animations and view them from various points of view. It also allows to perform analysis of selected animation. This analysis consists of joint activity visualization and racket trajectory analysis.

3.1 Humanoid Body

The humanoid body that has been used is H-Anim 1.1 compliant. It follows the recommended structure of joint hierarchy and it uses the names recommended for particular joints and segments (see Figure 3). On the other hand it uses non-standard level of articulation that corresponds with neither of the levels described in H-Anim 1.1 specification [4] (LOA 0 – 3).

From the H-Anim joint hierarchy only the 17 joints remained in the body. These are the shoulders, elbows, wrists and hips, knees and ankles on both sides, the skullbase and vc7 (the last cervical vertebrae) and HumanoidRoot, sacroiliac and vl5 (the last lumbar vertebrae) – the three joints that represent the first two levels of H-Anim hierarchy and their location is identical.

Since the applet serves the experimental purposes, it has been created to be as simple as possible. Some of the joints with greater significance for analyzing tennis strokes were left out too. This was the case of sternoclavicular and acromioclavicular joints that are responsible for moving the shoulders (in the hierarchy between the vertebrae and shoulder joint).

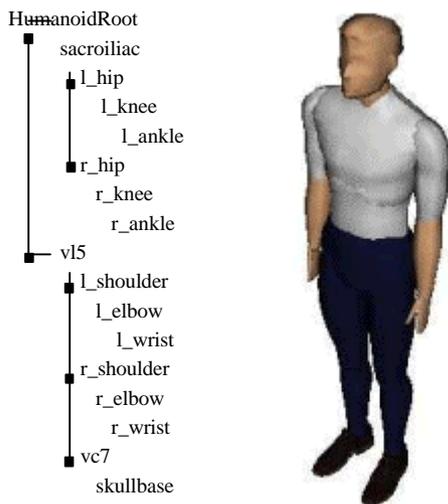


Figure 3: The appearance of the humanoid and its joint hierarchy

The humanoid's geometry has been borrowed from the Body Parts library of the humanoid Baxter [5], that is available to download for non-commercial use.

3.2 The Animation

The representation of the motion that has been used by the applet is closely derived from the VRML interpolators of rotation (OrientationInterpolator) and position (PositionInterpolator).

The applet uses the VRML mechanism of timer (TimeSensor node) connected to an interpolator. The only thing applet does is starting and stopping the timer.

The functions of starting and stopping the animation listed in paragraph 2.2.2 are easily implemented using the VRML mechanism. The applet in its initial phase creates two timers (TimeSensors) and links them together. One of them then starts and stops the animation (sending its *time* eventOut to the *startTime* or *stopTime* eventIn of the second timer). The second timer controls the interpolators.

The adjustable animation speed is implemented simply by adjusting the *cycleInterval* field of the timer that controls the interpolators.

That is what is the VRML mechanism very easily usable for.

The implementation of the pausing and restarting function was more complicated. The timer would start the animation from the beginning again. It is necessary to

change the *cycleInterval* field of the timer. The new value is:

$$ci_n = ci * (1 - T)$$

where T is the last time fraction sent before the TimeSensor was deactivated, ci_n and ci is the new and the previous value of *cycleInterval*.

Then the interpolators must be changed after pausing the animation. The keys and key values already used are left out and the remaining keys transformed:

$$k_n = (k - T) / (1 - T)$$

where T is the last time fraction sent before the TimeSensor was deactivated, k_n and k is the new and the previous value of the key. The key value that belongs to the new first key is read from the joint connected to the interpolator.

The frame-by-frame tracing function has not been implemented. The representation of animation with full description of each animation frame would allow to do so easily. The VRML interpolator representation doesn't. The implementation of this function is still subject of research.

3.3 Motion Analysis

The following paragraphs inform of the implemented motion analysis experiments and their results.

3.3.1 Joint Activity Analysis

The activity of selected joint was visualized as a graph of the changes of Euler angles (see Figure 4). The Euler angles are computed from the interpolator data for each key value and are used as vertices of a polyline graph. The interpolators perform the linear interpolation, so using polyline for the graph describes precisely the behavior of virtual humanoid. This clearly shows the linear interpolation as another weakness of animations represented by the means of VRML.

VRML uses the axis-angle representation of rotation. In order to find the Euler angles, this representation has to be transformed to the Euler parameters, which are a quaternion in a scalar-vector representation. A quaternion is further transformed to a transformation matrix. As a consequence of the Euler's rotation theorem any rotation matrix is a product of three different rotation matrices.

$$A = BCD$$

The matrices B , C and D , or more precisely the angles of rotation they describe, are what we are looking for. The last step is to choose a convention for Euler angles. The one we have chosen is the "xyz-convention" which specifies the B , C and D matrices to be the matrices of rotation around the x , y and z axis in this

order. That gives us the relationship between the elements of the **A** and the Euler angles.

More about various representations of rotation can be found [7].

3.3.2 The Results of Joint Activity Analysis

Visualization of the Euler angles gives us the distinct visual description of the motion, but very difficult to read. The more general the motion is, the more confusing can this visualization be. The special cases such as the gait analysis did not require any complicated computing and were readable. On the contrary the more complex moves, such as the tennis strokes, need different visualization methods.

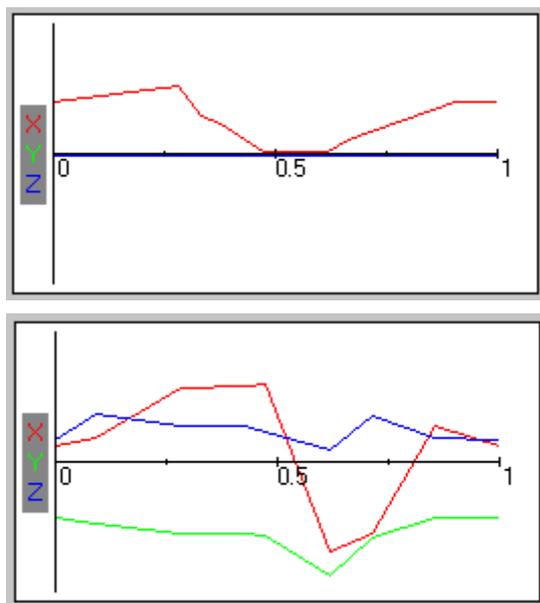


Figure 4: The Euler angles graphs for the cylindrical and spherical joints respectively. (Colors of graphs correspond to the colors of the axis signs on the left)

The possible improvement of described approach might be comparing the graphic results with a similar move that would be considered to be perfect and would be processed the same way.

3.3.3 The Course of the Racket

As it was mentioned in paragraph 2.3.2, finding the trajectory of the racket is the basis for any further analysis of the tennis stroke motion.

For this purpose the racket is represented by one point near the center that can be considered the hitting center of the racket. The racket's orientation that influences the ball even more than the direction of the

move will be in the future probably expressed by adding one more point, thus creating a normal vector.

In the first pass of this analysis the animation is sampled. It is started five times slower to generate more time fractions and with every fifth time fraction the position of the "hitting" point is computed. In order to do this, the transformation matrix has to be found each time. This matrix is a product of all the transformation matrices describing the joint hierarchy above the "hitting" point, each of them has to be computed from the axis-angle specified rotation and the center of this rotation. The more joints in the hierarchy, the more demanding the computing is.

The sample positions stored together with the values of relevant time fractions are further processed. The "velocity" vectors specified by each two successive positions are compared with the direction in which the stroke is supposed to be performed. To simplify it, it was chosen to be the positive direction of the *z* axis. For the general direction of the stroke only the projection of the "velocity" vector to the *xz* plane is important. The results of this comparing is transformed to a color:

$$\text{Red} = 255 * (x / z)$$

$$\text{Green} = 255 * (1 - x / z)$$

The *x* and *z* are the components of the "velocity" vector and Red and Green the components of RGB color representation.

The last step is to paint the points to the animation's time-line and color them. To do this, we use the time fractions stored in the process of sampling and the list of colors described above. The "greener" the point is, the closer the move is to the specified direction in that moment as shown in Figures 5 and 6.

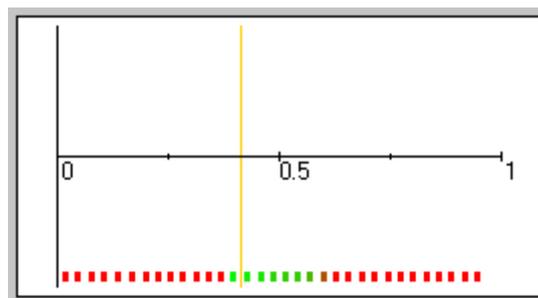


Figure 5: The racket's course analysis (forehand volley)

The first point that is colored with a color with certain level of green component can be considered to represent the moment when the racket should meet the ball. The length of the sequence of green points indicates how long the racket "pushes" the ball forth.



Figure 6: Position of the player in the moment pointed out in Figure 5 (forehand volley)

To determine the trajectory of the reflected ball we will have to do more research and it is planned for the future.

4 Conclusions

The result of the project is an applet capable of viewing and analyzing VRML based animations. It brought knowledge usable in further work with VRML animations, an insight into the function of the mechanism of VRML timer and interpolator and possible ways to make use of it. This research can evolve into the system of creating animations of more complex actions by putting together simple moves stored in some sort of a library.

Second part of the research was the motion analysis. In case of the visualization of the joint activity, the chosen approach proved not to be very suitable for presenting a general spherical joint and the complex move such as the tennis stroke. On the other hand the experiments inquiring the possibilities to strike the ball were more promising. To make use of the results here will require more research.

5 Future Work

In order to create useful application based on the results of this project, more research has to be done in two following directions.

- Adding the tennis ball into the scene
- Acquiring the animations or recorded motions

The first will require more research on the influence of the racket on the ball and its trajectory. The question to be answered is: “What will be the trajectory of the ball if it is hit by the analyzed move in the point detected as the best?” The methodology may also require to solve advanced problems of this type, for instance: “What if

the approaching ball is in certain height and has certain spin?”

The second is even more closely related to the functionality of the developed system as a training device. In the best case, the teacher would have the possibility to record the motions of the player using the motion capture technology. Then the task is to transform the recorded data to the form the system would work with. But since the motion capturing hardware is not really affordable, the system should be equipped with a library of animations that would cover the methodological needs. In this case the task is wider – to find the ways (animation software or motion capturing or both) and create the library.

Acknowledgements

This work has been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic under research program No._Y04/98: 212300014 (Research in the area of information technologies and communications).

Our thanks belong to people from Czech Tennis Association, which spent their time discussing various topics presented in this work.

References

- [1] Official pages of the LIG laboratory
ligwww.epfl.ch
- [2] Official pages of the Miralab laboratory
www.miralab.unige.ch
- [3] www.cis.upenn.edu/~beitler
- [4] Human Animation Task Group within Web3D Consortium
www.h-anim.org
- [5] ligwww.epfl.ch/~babski/StandardBody
- [6] Wilcox, S. K: *Web developer.com Guide to 3D Avatars*, 1998
- [7] A free service for mathematical community provided by Wolfram Research
mathworld.wolfram.com
- [8] An example motion analysis application.
www.frontiernet.net/~imaging/gait_model.html

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Spring Conference on Computer Graphics, Budmerice, Slovakia
Copyright © ACM 2002 1-58113-608-0/02/009...\$5.00