Computer Methods in Biomechanics and Biomedical Engineering
Vol. 15, No. 9, Jul 2012, 919–924

RESEARCH ARTICLE

A virtual environment for learning to view during aerial movements

Maurice R. Yeadon* and Jon P. Knight

Sports Biomechanics and Motor Control Laboratory, School of Sport, Exercise and Health Sciences, Loughborough University, Leics, ENGLAND, LE11 3TU

(*June 2010*)

Training gymnasts to view the landing area when learning aerial skills may lead to more consistent landings but can be problematic and potentially dangerous. A virtual environment allowing gymnasts to be introduced to viewing techniques safely is presented. The system is based on existing simulation models and visualisation software, and is implemented using client-server technology to allow reuse with new simulation models in the future.

Keywords: virtual environment, gymnastics, OpenInventor, client-server, simulation

1. INTRODUCTION

There is anecdotal evidence that many gymnasts performing aerial movements view the landing area for only a small part of the aerial phase whereas certain elite gymnasts view the landing mat for a large part of the flight phase. This suggests that a gymnast who is able to see the landing area throughout the aerial phase of an apparatus dismount may be expected to perform more consistent and higher quality landings.

Training a gymnast to acquire such viewing skills is difficult however. The aerial phase of a dismount lasts little more than one second, and the gymnast's body, and therefore head, will be rotating rapidly. When learning a multiple twisting double somersault dismount the gymnast has many competing demands and aspects such as viewing may fail to be developed.

There is also the question of safety. Gymnastics training, even with suitably qualified coaches and the required specialist support equipment, can lead to injury, especially when the gymnast is attempting to learn a new skill. Instructing a gymnast to view during the aerial phase of a movement may have a detrimental effect on other important aspects of technique during learning. This can mean that there is a higher risk of failing to produce the correct rotations, and thus increase the risk of injury.

Training users in difficult tasks until they become second nature is a fairly common problem, especially if early failures during training could lead to catastrophic results. Virtual environments have been used in a variety of fields for training, where the trainee can safely make mistakes without fear of injury or loss. Pilots Macchiarella et al. (2005), train drivers Tichon et al. (2006) and submariners

*Corresponding author. Email: M.R.Yeadon@lboro.ac.uk

ISSN: 1741-5977 print/ISSN 1741-5985 online

DOI: 10.1080/10255842.2011.566563
http://www.informaworld.com

Hamilton (2004) regularly spend time in various types of simulators. Virtual reality based training systems have also been used in the medical field Fulk (2005).

In the sporting world, virtual environments have been used to train the USA bobsled team using representations of existing bobsled tracks Kelly and Hubbard (2000). American Football players also use virtual reality training aids to examine different player viewpoints during given plays Beier (2002). Both of these sports are potentially dangerous, so being able to do part of the training in the safety of a virtual environment can reduce the risk of training injuries.

The widespread use of ever more powerful computers and games consoles has lead to the use of "exertainment" games that couple exercise, training and computer games Yang et al. (2007). The benefit of these, coupled with willing user acceptance based on previous experience with video games, has been recognised by some local authorities that have introduced such systems into schools.

There are some drawbacks to using virtual reality techniques and simulated environments to provide training for skill acquisition. The major disadvantage is accidental training of responses to artifacts that only exist in the simulated environment rather than the real world.

However it is not always the case that increasing simulation fidelity will improve training responses Salas and Burke (2002). Simulations need to be designed so that they capture adequate detail to allow the learner to acquire the skills required without being distracted by differences from the real environment. This will depend not only on the cognitive requirement of the skill being learnt but also on the familiarity that the participant has with the environment and other tasks that take place around it.

With the above in mind we set out to build a system that would allow gymnasts to learn to view landing areas during aerial movements within a safe virtual environment. The aim was to allow a gymnast to build up viewing technique using appropriate head movements, initially without having to be distracted by actually performing the movement. Once the viewing technique had been learnt it could be applied to a progression of movements in the gym leading up to the desired skill.

We already had a number of components, both hardware and software, that we felt could form the basis of such a system. This paper documents these components and how we combined them together to produce a working virtual training environment. We include a case study to demonstrate the use of the system.

2. METHODS

2.1 System Requirements

We listed a number of basic requirements for a system to allow gymnasts to learning viewing technique in a virtual environment:

- The mathematical model of aerial movement must be able to run in real time. However it should also be possible to slow the simulation model down to allow novice users to gradually develop their techniques of viewing during flight phase.
- The system must be interactive, tracking movement of the gymnast's head and providing appropriately matched views of the virtual environment. This must always happen in real time, even if the simulated movement is occurring slower than real time.
- The system must be able to provide multiple views, including the gymnast's view out from the head of their avatar into the virtual environment and a

view that allows the gymnast's coach to observe the simulated movement from the side of the apparatus much as they would in real life.

- In order to utilise the gymnast view in training the user will need to be immersed in the virtual environment. This implies the use of a Head Mounted Display (HMD) so that the user can see the virtual scene when moving the head.
- The system must allow for a variety of human movements to be simulated. This not only permits a number of different gymnastic movements and disciplines to use it as a training aid, but may also allow it to be used in other sports that have similar aerial flight phases where viewing the landing area can be of benefit.

2.2 *Existing Facilities*

The system we developed was based upon a number of existing tools and facilities that we had access to in the biomechanics and motor control research group.

2.2.1 *Mathematical Models of Human Movement*

The first, and probably most important, of these tools comprises the existing mathematical models simulating human movement. These have been developed over the last thirty years by members of the group and allow simulations of twisting somersaults Yeadon et al. (1990), high bar giant circles Hiley and Yeadon (2001), long swings on rings Yeadon and Brewin (2003), tumbling Yeadon and King (2002), vaulting King and Yeadon (2005) and springboard diving Yeadon et al. (2006).

These models are mostly written in FORTRAN 77, and some have been ported from the original mainframes and minicomputers on which they were developed to run under more modern systems, such as PCs running Linux. These simulation models were not originally intended to run in real time, and do not themselves produce any graphical output. Instead they generate text files containing frames of angle data for the joints of a segmented body model, along with the position of the pelvis in each frame.

2.2.2 *Rendering Images from the Models*

A variety of post processing code has been developed over the years to handle these angle files. Code was developed within the group to produce images for books and journal articles, along with MPEG movies for inclusion in power point slides and web pages. This software uses the Silicon Graphics OpenInventor modelling environment to produce a three-dimensional model of the segmented human body, with the joint angles and pelvis position set according to the data returned from the simulation model. Originally the code ran on an SGI workstation under the IRIX version of UNIX, but some years ago was ported to run on a standard PC under Linux.

The code for this is split into a set of Perl scripts for processing the angle files and turning them into appropriate ASCII OpenInventor files, and then a couple of programs written in C++ to take these OpenInventor files and then render them into images.

The graphics rendering system permits a number of different sets of body data to be used with any angle data file. For example we have a model of a female gymnast wearing a leotard that was built up from data obtained using the Loughborough Anthropometric Shadow Scanner (LASS) body scanner West (1987). The points from the scanned body were filtered and then used to construct control points to generate a Non-Uniform Rational B-Spline (NURBS) surface model. This produced a reasonably high quality output but was slow to render on older hardware, so we also have a much faster model of the same gymnast with the scanned data replaced by sets of ellipsoids. Similar graphics models exist for male gymnasts, as well as models for specialised activities (for example the aerials event in freestyle skiing).

As the simulation of the human movement relies on the mechanics of moving limbs to control rotations, the system needs to account for the different body segment sizes that different gymnasts have. Such anthropometric data can be provided to the graphics renderer as well, which will then automatically scale the OpenInventor body segments (be they based on NURBS or ellipsoids). This allows us to provide rendered output with appropriately sized body segments for the gymnast being simulated.

The graphics modelling system allows the user to also include OpenInventor models of equipment and rooms, so that the rendered version of the simulated movement can be seen in the context of a normal gymnastics environment. Models have been constructed for several pieces of equipment including trampoline, high bar, rings and diving pool. Equipment models also have associated Perl scripts that allow them to dynamically respond to the data held in the angle files, ensuring that the rings, for example, are positioned where the gymnast's hands are.

2.2.3 Determining Head Orientation

To determine the head orientation of the gymnast, the group already had an Ascension 3D-Bird. This is a commercial three degrees of freedom inertial tracking unit, that is small and light enough to be easily mounted on the gymnast's head without unduly affecting range of movement.

The unit connects to a PC via a 4.5m long serial cable and comes with drivers for Windows. In the past we had used this sensor for other tracking applications and had ported the Ascension driver code to run under Linux. The drivers are capable of providing data as rotation matrices, Euler angles or quaternions, with up to 160 samples per second.

2.2.4 Head Mounted Display

Whilst initial development and testing of the training system could be performed using normal LCD computer monitors, we wanted to provide a more immersive experience for the gymnast using a Head Mounted Display (HMD). Fortunately we had access to a Trivisio 3scope HMD, which comprises a pair of miniature 800x600 24 bit colour displays mounted on a head band. Each display can be driven independently if required, although in prior testing we found that for many applications we could quite adequately use a single video signal split and sent to both displays simultaneously.

After gaining some experience with the 3scope, we then upgraded to an nVisor SX HMD, which offers stereo 1280x1024 full colour displays with a wider viewing angle (see Figure 1). This display improved the quality of the overall system, as the nVisor excluded peripheral vision of the real environment that the user was sitting in and also was less likely to move relative to the user's head when the wearer was making sharp turns or suddenly looking up or down quickly.

2.3 System Design

We wanted to produce a system that would be flexible and allow us to use it as a platform for future research. When we started working on the system, it wasn't yet clear that a single PC could run the simulation models for human movement, handle the data streaming in from the 3D-Bird and generate and display a real time virtual scene from the OpenInventor models. We therefore decided to split

Figure 1. The nVisor SX Head Mounted Display used in the system being worn by the gymnast.

the system into two halves as a client-server design to allow us to spread the processing load if necessary.

The back end server is responsible for running the simulation model of the human movement. Rather than running a simulation to completion, it runs a single time step in response to a query over the network socket from the client. When the time step is finished, it returns the angle data and pelvis position back to the client for rendering.

The client side of the system is where the interaction with the gymnast takes place, accepting data from the 3D-Bird head tracking sensor, requesting a new simulation frame from the server and then displaying the resulting virtual environment. The client application also provides the models for the virtual room that the virtual gymnast is moving in, including models of equipment such as trampoline.

As the 3D-Bird is providing head positions in real time, the client is designed so that any failure of the server to return angle data does not stop the 3D rendering process from taking place, using the existing angle data and the most recent head orientation data from the 3D-Bird.

This client-server model provides a number of benefits. Firstly it allows simulations models to be completely replaced, without unduly affecting the front end client. The client code does not need to know what movement is being simulated - it could be twisting somersaults on a trampoline or giant circles on a high bar. The user obviously will need that information, and will probably also want to include appropriate equipment models in the client invocation for the simulation model being rendered.

The modularisation also helps insulate the server side simulation models from changes in the front end client. This might include new rendering views, additional sensor inputs or even porting to new platforms.

2.4 System Implementation

The server side application is formed from a main routine written in C that deals with network communication using standard socket libraries, another routine written in C that acts as a harness to the actual simulation model, and lastly the simulation model routines themselves, usually written in FORTRAN. The latter have been extracted from the original FORTRAN 77 based simulation code, tweaked slightly so that the routines simply generate a single time step rather than trying to run a simulation of the whole movement. The time needed to compute this time step needs to be less than $\frac{1}{50}$ of a second if the simulation is to run in real time. However the server can delay sending the result back if the user wishes to run the simulation at a slower pace during the early phases of training. The client can indicate the delay required in each time step request, so it is possible to speed up and/or slow down a simulation dynamically.

The server is stateful, as it needs to be able to maintain information about the current simulation for each client that connects to it. The TCP connection from the client is not broken when the server returns angle data. Instead the server simply returns to listening for the next request from the client to generate the next step in the simulation.

The reason for doing this is that it allows the potential for having the client pass angle data *into* the server during the request for the next step. The server can take this angle data and suitably adjust the stored state for the simulation before asking the FORTRAN routines to generate the next time step. This architecture therefore allows the input data for various body segment angles from the client to alter the simulation model in real time. These body segment input angles could come from keyboard or mouse generated data, or from other realtime sensor hardware besides the 3D-Bird.

The client side of the system is also written in multiple languages to allow easier integration of the existing software that we had available. As the OpenInventor libraries' main bindings are presented in C++, this is the language chosen for most of the code in the client. However the user space driver for the 3D-Bird uses a pre-existing set of C routines and there is also an embedded Perl interpreter that is used to construct and modify the OpenInventor data structures based on the existing image rendering systems we have.

As the client has a number of real time demands placed on it, the application uses multiple threads and non-blocking I/O techniques to allow the multiple aspects to run simultaneously within the normal main OpenInventor application loop. The 3D-Bird thread collects angle data from the sensor and makes it available in a set of global variables that other threads can read. An OpenInventor timer is set to go off every $\frac{1}{50}$ of a second, which then runs a callback routine to collect any data pending from the 3D-Bird, uses this to alter the head orientation angles in the OpenInventor model and to see if a block of valid angle data is returned from the server.

If the server has returned angle data, the OpenInventor data structures are updated to reflect this. Irrespective of whether data was obtained from the server or not, the routine then tells OpenInventor to render the views of the virtual environment again. This means that the 3D-Bird head orientation angles are always reflected in a model generated roughly every $\frac{1}{50}$ of a second, even if the back end simulation server has not managed to return data in that time frame.

Meanwhile another thread is constantly looping to generate the network requests to the server and gather the response data. It is this routine that marks the data as valid - this allows the timed callback routine above to run even if the server is only part way through returning its data.

Figure 2. The system in use by the experienced artistic gymnast.

There are also OpenInventor callback routines set up for keyboard and mouse events. These are used to provide control input such as the speed up or slow down requests, kill the network request thread, quit the application, etc. These callbacks are also one way to allow body angle data to be altered on the fly, permitting dynamic modification to the current simulation model state as detailed above.

Despite the design of the system into separate client and server to allow an expected inability of a PC to handle all requirements, we have found that even a relatively old and slow machine is capable of running both the client and our back end simulation code. Our main machine during development, testing and the case study below was a Dell n-series workstation with 1GB of RAM, a single 2.8GHz Intel Xeon processor and an Nvidia GeForce 4 graphics card with dual DVI outputs driving two 800x600 displays (LCD screens and/or the HMD unit described above). This machine was running Linux (initially Fedora Core, replaced later with Debian Etch), with the Nvidia proprietary graphics drivers.

3. CASE STUDY

Once the system described above was implemented, we tested it with a real gymnast. This individual was an experienced artistic gymnast who was about to learn a new skill - a double somersault dismount from high bar with one twist in the second somersault (a back-in full-out). He did not normally attempt to view the landing whilst performing a high bar dismount.

We introduced him to the system and allowed him to practise viewing the movement for some time. He used the HMD while a normal LCD screen showed his coach and the research team an "external" view so that they could provide timing advice on changing head orientation (Figure 2).

(a) Original test

(b) Retest 12 months later

Figure 3. Viewing during a back-in full-out dismount from high bar.

The gymnast made good progress in learning the appropriate head movements during a single session using the HMD which showed a gymnast's view during a simulated back-in full-out on trampoline. Subsequently in the gymnasium he was then taken through a progression of twisting somersault skills culminating in the target dismount from high bar. In the flight phase of this dismount it can be seen that the gymnast's head position is such that he can view the landing area throughout (Figure 3(a)). Afterwards he was questioned as to whether he felt the simulation training had helped him learn what to expect during the new skill and whether it had helped him view the landing area. The result was positive, and he was supportive of the use of the system in such training. A performance of the back-in full-out dismount from high bar was recorded 12 months later to check that the gymnast had retained his viewing skills (Figure 3(b)).

4. DISCUSSION AND FURTHER WORK

The system described in this paper has the potential for enabling faster and safer learning of the correct head movement and viewing strategy for maintaining the correct orientation during aerial gymnastic movements.

The client/server design of the system has proved to be a success. As well as the initial aerial movement model that the system was initially designed for, another server back end has been developed that combines two separate pre-existing simulation models together. In this case we have taken an initial simulation of high bar giant circles and then allowed the gymnast to indicate the point at which he wishes to dismount. At this time, the back end server simulation switches to the somersault aerial simulation, with the new simulation state being generated from the last state that the giant circle simulation was in.

As well as the coaching applications demonstrated in the case study, the system has a number of other potential uses. It can take existing simulation code from previous research and, with only a small amount of adjustment, turn it into a back end server, which will allow researchers to more easily visualise their simulation output. This is in itself a valuable application of the system, as it can provide immediate feedback to the researcher who is developing new biomechanical simulation models. Until now such researchers have had to rely on rendering still images and making MPEG movies from their simulations, which is a somewhat more time consuming task.

The system can also be used in teaching biomechanics to undergraduates. An early version of the system that allowed body angles to be changed using keyboard input was demonstrated to Masters students. They found it to be a fun and useful way of investigating how arm movements can initiate twisting in somersaults, demonstrating first hand how the mechanics they were learning could be applied. In fact it was actually quite difficult to prise the machine away from some of them at the end of the session!

We are constantly developing the system at this point and have a number of avenues for further investigation. The first of these is the use of additional sensors to provide a full real time simulation driven by real limb movements. We have done some promising initial tests using a Polhemus Fastrak six degrees of freedom electro-magnetic sensor system, using four sensors to determine orientations of individual upper arms and also a combined "leg" (as the legs are often kept together during aerial movements).

We are also planning a more thorough evaluation of the system. This will involve trialing it with more gymnasts and also comparing on board gymnast's view video during real gymnastic movements with the simulated view that we generate. We are also interested in determining how immersive the simulation needs to be in order to be effective - i.e. when an elite gymnast views the moving scene is it just the head that moves or do the eyes move as well?

References

Beier KP. 2002. The Virtual Football Trainer. Fathom . Available online at http://www.fathom.com/feature/122244/

Fulk G. 2005. Locomotor training and virtual reality-based balance training for an individual with multiple sclerosis: A case report.. Journal of Neurologic Physical Therapy. .

Hamilton R. 2004. 'Pipe Dream' becomes reality as sub training enters information age. Sea Power .

Hiley M, Yeadon M. 2001. Swinging around the high bar. Physics Education 36:14–17.

Kelly A, Hubbard M. 2000. Design and construction of a bobsled driver training simulator. Sports Engineering pp. 13–24.

King M, Yeadon M. 2005. Factors influencing performance in the Hecht vault and implications for modelling. Journal of Biomechanics 38:145–151.

Macchiarella N, Brady T, Arban P. 2005. High fidelity flight training devices in the training of ab initio flight students.In: *24th Digital Avionics Systems Conference, 30th October - 3rd November 2005* (. ; vol. 1). pp. 5.B.2–5.1–9.

Salas E, Burke C. 2002. Simulation for training is effective when.... Quality & Safety in Health Care 11:119–120.

Tichon J, G. W, Mildred T. 2006. Virtual Training Environments to Improve Train Drivers' Crisis Decision Making. Simulation Industry Association of Australia .

West G. 1987. Loughborough Anthropometric Shadow Scanner (LASS) [master's thesis]. : Loughborough University of Technology.

Yang S, Smith B, Graham G. 2007. Healthy Video Gaming: Oxymoron or Possibility. Innovate Journal of Online Education 4. Available online at http://www.innovateonline.info/print.php?view=pdf&id=186

Yeadon MR, Atha J, Hales FD. 1990. The simulation of aerial movement - IV. A computer simulation model. Journal of Biomechanics 23:85–89.

Yeadon M, Brewin M. 2003. Optimised performance of the backward longswing on rings. Journal of Biomechanics 36:545–552.

Yeadon M, King M. 2002. Evaluation of a torque-driven simulation model of tumbling. Journal of Applied Biomechanics 18:195–206.

Yeadon M, Kong P, King M. 2006. Parameter determination for computer simulation model of a diver and a springboard. Journal of Applied Biomechanics 22:167–176.