Central European Multimedia and Virtual Reality Conference (2005)
J. Zara, J.Sloup (Editors)

The Drum Set Tutorial System by Means of Inverse Kinematics

Martin Linda, Vladimír Štěpán, and Adam J. Sporka

Department of Computer Science and Engineering, Czech Technical University, Prague, Czech Republic

Abstract

In this paper we describe a process of rhythm visualization by means of the conversion of note description into a virtual human animation. An assignment of hands to strokes will be described as well as drumstick trajectory inference. We focus on using inverse kinematics to animate a virtual drummer in order to adapt animation to arbitrary drum set. We discuss specific problems of applying this technique to our application (such as the problem of insufficient joint constraints) that aims at a high level of realism and flexibility.

Categories and Subject Descriptors (according to ACM CCS) I.3.7 [Computer Graphics]: Virtual Reality; J.5 [Arts and Humanities]: Music, Performing arts (e.g., dance, music).

1. Introduction

The drum set is a common instrument in modern music styles such as pop, rock, and jazz. To play this instrument is a specific and complex musical discipline. Its complexity lies in the necessity to use all four limbs independently and perform the right motion at the right moment with each limb. The mobility of the limbs are naturally interconnected and therefore a certain amount of ambidexterity is developed in the process of learning to play, which takes substantial training. Another specific feature of the drum set as a musical instrument is that there is no mandatory standard configuration. Thus various sets may differ in the count, type and spatial layout of the components, according to the drummer's personal preference.

Our work concerns the following basic aspects of drumming: First, the drummers must decide what particular note to play and when. Second, they must schedule their motions to play the note in the correct time. This may be done once they understand the structure of the rhythm. In practice, we have observed that drummers learn, analyse and understand the rhythm more easily if they can actually see the motions necessary to perform the rhythm. Beginners are especially challenged by the problem of analysing the hits composing the rhythm and choosing the appropriate limb for the hit. A comprehensive visualization would be helpful. Our goal was to create such a visualization considering user-defined drum sets and user-defined rhythms.

To our knowledge, there is very little work done in this particular field. The only work dealing with the visualization of drum play we were able to find was the project called A Virtual Drummer in Java 3D [KNZ01]. This work presents a system that processes music stored in PCM signals, converts it into the MIDI events, and pre-calculates the drummer's motions using the key-frame animation technique for a fixed drum set configuration. The principal disadvantage of this approach is its inability to adapt in real time to changes in the configuration of the drum set. To change the configuration, it is necessary to define new key-frames for each addition or reposition of a drum in an external tool. The requirements set by our goal call for entirely different approach to the visualization.

2. Drum Play Visualization

We may define the rhythm as a sequence of hits along a continuous time axis. The majority of modern music styles have a fixed and regular metrum, i.e. the time axis may be divided into discrete intervals with a certain resolution. The rhythm of the drum play may then be described as a table of events in which the columns represent discrete points in time, and the rows would be the possible targets of the drumsticks (various drums of the current configuration). The elements of the event table stand for the particular events of the rhythm.

The drumming is affected by the configuration of the set. The type and location of the drums determine whether the drummer tends to use left or right hand on a particular

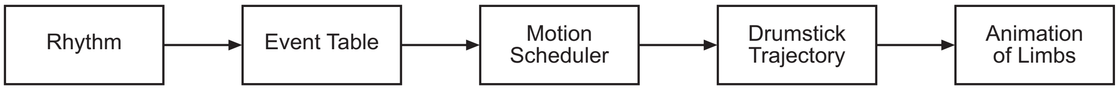

Figure 1: *Block diagram of the rhythm visualization process.*

drum. An incorrect alternation of hands may render some rhythms unplayable. As the particular choice of a configuration is usually done according to personal preference, the drummer should be given a chance to customize the drum set model used in the visualization. Figure 1 outlines the pipeline of the visualization considered in our work.

2.1. Motion Scheduler

The rhythm playback is driven by the traversal of the event table column-by-column. The events from the table determine the action to take in a given moment, such as the particular drumstick striking the drum or the foot pressing the pedal. It is necessary to ensure that no more that one event is assigned to a limb at a time, otherwise it would not be possible to play the rhythm. Scheduling the events to feet is straightforward as we make a simplifying assumption that a drum set has two pedals only. Therefore the scheduling task is reduced to hands. Obviously, for each event there are two possibilities for which hand to assign. The purpose of the motion scheduler is to make the decisions of which hand to assign to each event.

For this assignment, the following factors are considered: spatial configuration of the drum set, common practice, and personal preference of the drummer. E.g., the hi-hat cymbal is usually located on the left side of the set, but it is usually played by right hand. For each drum, a numerical priority is specified that determines which hand is more likely used to play it. The scheduling decision is also influenced by the history of previous decisions.

Properly selected alternation of hands can help avoid hand collisions. The scheduler decides with respect to the position of the drum, priority of the hand for the drum, and the history of the previous decisions.

2.2. Drumstick Trajectory Inference

A drum (or a cymbal) may be played by hitting different places on its surface (center, edge, rim) resulting in different timbres of the sound. To reflect this in our model, we introduce the concept of hit-points. A hit-point is a possible target of the drumstick. Multiple hit-points may be defined for each drum. In the previously defined event table, each row represents a hit-point.

The drumstick trajectory is to be constructed in the way that the end of the drumstick meets all scheduled hit-points. For a realistic effect, a mere knowledge of hit-points is not sufficient. In reality, a hit on a drum is performed by raising a drumstick above the drum followed by bringing it down on the drum. The drumstick then bounces back above the drum.

The Figure 2 shows a construction unit of the drumstick trajectory as modelled in our approach that represents a single hit of a drum. Its origin (point “0”) depends on where the previous unit has ended. The key moment of each unit is the hit-point itself (“2”). The points “1” and “3” were added to the model, simulating the pre-impact raise and the post-impact bounce of the drumstick respectively. Thus we simulate the typical whip of the drummer's wrist.

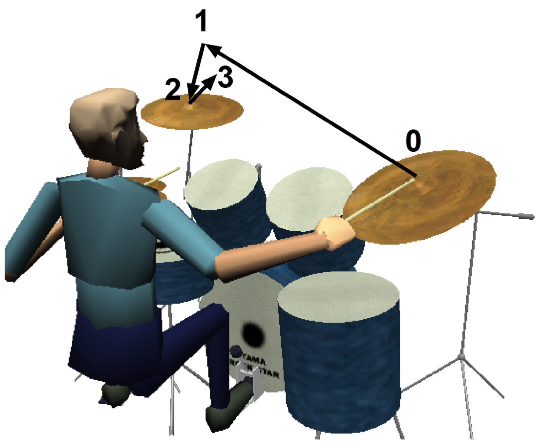

Fig. 2: *A trajectory of the end of the drumstick*

3. Inverse Kinematics Module

The purpose of the inverse kinematics (IK) module is to set up the joints of the virtual drummer so that the end of the drumstick follows the trajectory. The IK makes use of the Cyclic Coordinate Descent (CCD) iterative algorithm as described in [Wel93]. An arbitrary target location is given as input and CCD computes the joint rotations to move the end-effector (drumstick in our case) to this location.

The input to the IK module is the sequence of points interpolated between the four control points of the drumstick trajectory described above (see Figure 2). Each point is used as a target location to find the body posture (the rotations in joints) that is displayed at the appropriate time. Therefore a fixed drum set configuration and positioning of hit-points on the drums is not necessary. The animation can adapt to the drum set in real time. This aspect of our approach is important as it enables the user to visualize rhythm on arbitrary drum set.

Due to the nature of our task, we had to perform the animation with a limited amount of input data. particularly with the trajectory of the drumstick end without any addi-


Fig. 3: *Left – effect of insufficient joint constrains. Right – correct posture.*

tional knowledge of other joint positions. That imposed the following problems.

The IK algorithm brings the drumstick to its destination in the simplest possible way and the resulting effect can often look unnatural. Figure 3 shows an example: When a motion starts high (above a cymbal) and ends low (on snare drum), a real drummer would pull the hand back and strike the snare drum with an elbow low and close to body. The IK module would leave the elbow high and bend it down to reach the destination. The drumstick seems to thrust at the drum instead of beating it.

Using joint constrains (rotation limits and stiffness) that would reduce the space of possible solutions improves the realism to certain level but often not sufficiently.

Figure 4a (labels “A”, “B”, “C”, “D”) explains one trait of the CCD algorithm that affects the realism: Moving the end-effector away from the base straightens the manipulator and narrows the space of possible solutions. This ensures that the resulting posture would be realistic. Position B on Fig. 4a will be reached. In the opposite direction, back against the base, the space of solutions within the constraints would be extending, and CCD algorithm would most likely choose position “D” in Fig. 4a instead of “C”, which is desired. The less realistic postures tend to appear when the required motion of the end-effector aims against the limb and towards the body.

We have observed that this problem relates to the quasiperiodic character of drummer’s motions. A sequence of hits is often performed repeatedly. The first instance of such sequence may be animated realistically, but the others tend to be more and more deformed. Although the targets are the same, the motion is always computed from different starting conditions.

The problems mentioned above were eliminated using a very simple method. The new body posture is not computed continuously from the posture found for the previous target point. Instead, for each new end-effector target point (Fig. 4b, labels “b” and “c”) we use the same default starting body posture (Fig. 4b, label “a”) with hands bent close to a body. During the animated motion the arms never get bent more. Thus the end-effector never pushes against the limb and towards the body. The resulting posture is more realistic. The identical starting conditions also solves the problem of deformations during repeated motions. The cost for increased realism is a slightly longer computation, as it takes more iteration to compute the posture.

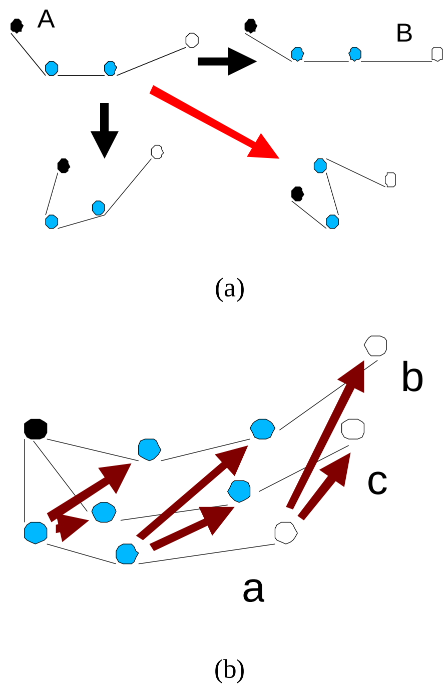

Figure 4: *(a) Issue of insufficient realism (b) Our solution (“a” – starting body posture; “b”, “c” – target postures.)*

4. Implementation

The visualization system has been implemented in MS VC++ 6.0 programming language [Mic98]. The visualization of the drum set and the humanoid has been realized using OpenGL [NDW03] and GLUT libraries. The system has been designed as a tool allowing its users to learn how to play specified rhythms on an arbitrary custom drum set. The drum set may be customized in an embedded drum set editor in which the hierarchy of the drums, their spatial relations, hit-points, and hand assignment priorities are specified. The rhythms to be visualized may be defined and later modified in a built-in pattern editor. The pattern editor automatically adapts to the current drum set configuration. The camera position and orientation in the visualization window may be arbitrarily modified so that the user may find the best view of the visualized rhythm. The drum set and the drummer figure are placed into a simple virtual environment. The geometry of the drummer figure is defined in an external file in VRML H-Anim format [Han04].

5. User Study

To evaluate the results of our work, especially with respect to its educational value, a simple user study has been carried out. Five persons, whose demographic data are shown in Table 1, participated in the experiment. Their experience was described by grades 1 (best) to 5 (worst). The participants were only acquainted with the basic controls of the application. They were asked to carry out the following tasks:

1. Observe the demonstration of the rhythm played on the pre-defined drum set and assess the quality and realism of the animation of the drummer with respect to aesthetics and common practice of drumming.
2. Compose your own rhythms using the pattern editor and decide whether the rhythm visualization meets your requirements or not.
3. Set up your own drum set using the embedded editor. Make a feasibility assessment. Then repeat step 2.
4. Judge whether the visual preview of the rhythm helped you to understand the rhythm and learn to play it.

One participant at a time was taking part in the experiment. The experimenters and the participants did not interact during execution of the tasks. The total duration of the experiment was not limited. Having completed the tasks, the participants were asked to make comments on each step in writing and assign grades to the following features of the system:

Fig. 5: *Pattern Editor*

partic.	age	ED	IT	OI	QA	RA	AU	H
A	25	4	1	no	2	1	1	yes
B	27	3	2	no	1	2	1	yes
C	24	2	4	no	2	2	1	yes
D	26	3	2	yes	3	2	1	yes
E	26	3	3	yes	1	1	1	yes

Table 1: *The user study data. ED – experience with drum play, IT – IT experience, OI – other instruments played; for other codes se list below.*

- Quality of drummer's animation (QA; 1 best, 5 worst)
- Realism of hand assignments (RA; 1 to 5)
- Adaptation to the user (AU; 1 to 5)
- Program helpful to analyse the rhythm (H; yes/no)

Despite the relatively small number of participants, their answers often described similar experience. Their feedback may be summarized as follows:

1. The users mostly find the animation realistic, except only for certain hits and drum set configurations in which they found the animation not being smooth enough. They all stated that their visual impression had been positively influenced by the drum animation.
2. The hand assignment is usually correct. However, in certain cases, it seems not to be the most efficient solution, such as the unnatural alteration of the hands playing the ride cymbal.
3. The drum set editor was found to be “user-friendly” both by its design and implementation. The automatic adaptation of the pattern editor to the drum set was also welcome.
4. The users all made the same conclusion that when slowed down, the animation helps the analysis and understanding of the rhythm.

6. Conclusion

In this document, a drum set tutorial system has been described. We have outlined the purpose of such system and the requirements for its implementation.

To implement the simulation and visualization of the drum player, we built a system that realized a pipeline consisting of motion planning, drumstick trajectory inference, and drummer's figure animation by means of inverse kinematics. We have identified a problem related to the instability of limb joints during the quasi-periodical motion. This can be avoided by always calculating the posture from the default starting body posture.

Finally, we have described our implementation of a tutorial system application that allows the following functionality:

- set-up of the custom drum sets
- editing of the rhythms for an arbitrary set
- visualization of rhythms in real time
- adaptation of the drum play technique to the particular drum set
- slow replay and the camera manipulation for an easy analysis of the rhythm

To evaluate the functionality, a simple user study has been carried out. It has shown that the aesthetic impression from the animation was positive. The application visualizes the rhythms correctly on the user-defined drum sets.

Our work may be followed by implementation of motion dynamics. This would improve the realism of the drummer's animation, as it would be possible to realistically simulate effects such as the bounce of the drumsticks on the drums or parametrize the motion for different strengths of the impact. Another improvement could be gained if the orientation of the end-effector would be taken into account while solving the IK problem.

Acknowledgment

We would like to thank to Bohuslav Franc for providing the code of his Java implementation of CCD algorithm and valuable help with rewriting it in C++.

References

[Gra99] Grace R. Hudba a zvuk na počítači (Music and Sound on Computer; in Czech) Grada, Praha, ISBN 80-7169-519-X, 1999.

[Eli04] Elias, H. Inverse Kinematics. Retrieved March 14, 2004 from: http://freespace.virgin.net/hugo.elias/models/m_ik.htm.

[KNZ01] Kragtwijk, M., Nijholt, A., Zwiers, J. An Animated Virtual Drummer. International Conference on Augmented, Virtual Environments and Three-Dimensional Imaging (ICAV3D), Mykonos, Greece, 2001.

[NDW03] Neider J., Davis T., Woo M.: The OpenGL Programming Guide, The Official Guide to Learning OpenGL, Version 1.4, Fourth Edition ("The Redbook"), Addison-Wesley, ISBN 03-2117-348-1, 2003.

[Han04] Specification for a Standard VRML Humanoid. Retrieved March 14, 2004 from: http://h-anim.org/Specifications/H-Anim1.0.

[Mic98] The Microsoft Corporation: MSDN Library Visual Studio 6.0 release, Software development environment documentation, 1998.

[Vrm97] VRML 97, international standard ISO/IEC 14772-1:1997.

[Wel93] Welman, C. (1993). Kinematics and Geometric Constraints for Articulated Figure Manipulation. Master's thesis, Simon Fraser University.