

The Animation Loop Station: Near Real-Time Animation Production

Benjamin Walther-Franks, Florian Biermann,
Nikolaas Steenbergen, and Rainer Malaka

Research Group Digital Media
TZI, University of Bremen, Germany

Abstract. We present the animation loop station, a near real-time animation production system. It brings the concept of loop stations or loopers commonly used for live music performances to the animation domain. Our system allows a single animator to build an animation layer by layer by capturing his own movement and building up complex motion through several passes or takes. In order not to interfere with capture and not break the performer’s flow, control commands are handled by a speech interface.

Keywords: computer animation, computer puppetry, performance animation, layered animation, loop stations, performing arts.

1 Introduction

Motion capture is a method for fast, cost-efficient computer animation. It is widely used in the film, television and computer game industries. The increasing availability of low-cost full-body and facial capturing devices such as Microsoft’s Kinect even moves motion capture in the reach of novices and small companies [2]. But while this approach already speeds up production times in comparison to offline animation practices such as keyframe animation, the motion capture workflow is still complex, requiring animators to clean, edit and apply capture data to rigged characters. Thus, while mocap includes real-time elements, the entire workflow production is still far from real-time.

In the music industry, changing consumer behavior has shifted the emphasis from selling studio productions toward live events. Simple devices, loop stations, enable small bands or even single person acts to create high fidelity songs on stage without pre-recorded material. By recording samples on stage, looping them, and cleverly layering multiple sample loops a single artist can create complex pieces entirely on his own, live, in front of an audience. The realtime music production becomes as much part of the experience as the “resulting” piece.

Looper workflows enable a rapid production style with a “live” quality that other areas of media production, such as animation, can benefit from. All that is lacking are the right concepts and interfaces. As a first step in this direction, we present the animation loop station, a near real-time animation production system. It enables a single artist to create complex animations just by acting

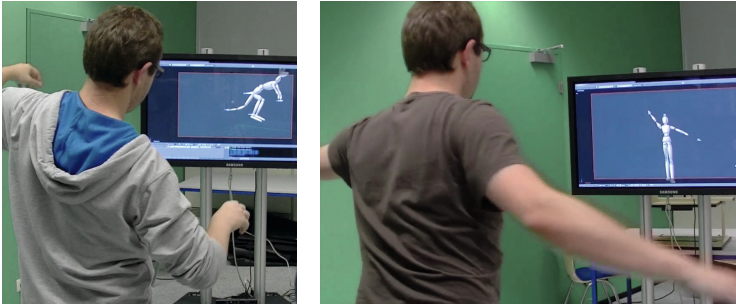


Fig. 1. The animation loop station in *Puppeteer* (virtual reality-style remote hand interaction; left) and *Performer* (body-to-body mapping; right) control modes

out and layering the animation, much like a musician constructs a song on stage by layering tracks. Building on recent developments in the area of computer puppetry [13], we have developed an environment that truly moves computer animation into the domain of real-time production.

2 The Animation Loop Station

Loop stations, loopers, or live looping samplers, are hardware or software devices that enable easy recording, playback and layering (overdubbing) of samples. They usually take the form of small hardware boxes, with pedals for foot activation, such as the *DigiTech JamMan* series [1]. Software loopers, such as *Sooper-Looper* [2] (which is in turn based on the *Echoplex Digital Pro LoopIII and LoopIV* hardware devices) run on common operating systems and only require the appropriate audio I/O to produce the same functionality.

These devices enable recording and editing in the production domain (audio), while control (record, play, undo, stop, clear) is administered through hardware or software pedals and buttons, essentially low-fidelity spatial input. For animation, we need to record spatial input to bring virtual characters to life, occupying spatial input for the production domain (3D motion). In our animation looping system, we thus switch the modalities to aural input for control and gestural input for content creation. Since only 3D body motion is recorded, the aural channel is free for recording, layering and playback control.

2.1 Capture

The animation loop station offers two control modes, based on distinct animation metaphors. For “classical” full body motion capture we have the *Performer* mode, which maps the movements of the loop station user’s skeleton onto a

¹ <http://www.digitech.com/>

² <http://essey.net/sooperlooper/>

humanoid-like character’s skeleton. This is how most motion capture works and it enables a user with a certain capability of bodily expression to create a wide variety of character motion. For animation of non-humanoid characters or other scene elements we have the *Puppeteer* mode that employs manipulation techniques from virtual reality and computer puppetry. Our current implementation focuses on 6DOF capture for 3D animation, although the loop station concept could be applied to other input devices such as multi-touch interactive screens.

2.2 Control

For control via the aural modality we use spoken commands that are detected by state-of-the-art speech detection technology. These are simple one or two word commands that change the mode of the loop station. We implemented commands for basic recording and playback control much like audio loopers offer: “Record”, “Play”, “Stop”, “Undo”, and “Clear”.

3 Live Animation Techniques

For both capture modes, we employ the principle of animation layering [1]. In each pass, certain features will be controlled live by captured input. If these features already have motion, there are three modes of layering:

- *Destructive* layering replaces the existing motion with the new motion.
- *Additive* layering adds the new motion to the existing motion (cf. [1]).
- *Blended* layering combines new and existing motion with a given weighting.

our system also includes timing modes that use spatiotemporal input to only drive the timing of a previously defined motion, without changing its component. Say the animator is happy with a trajectory, but wishes to speed up parts of the motion while slowing down others. The animation loop station offers *Drag-imation*-style performance timing [4] that allows the animator to drag an object through time along its trajectory at the desired pace, and use the new pacing as the new timing.

4 Implementation

Our current implementation of the animation loop station is written in python as an add-on to the open source 3D modeling and animation software Blender³. We chose to write an extension for Blender as it offers a powerful python API and a full classical animation pipeline. We use a Microsoft Kinect sensor for full-body motion capture and audio input. A server application translates skeleton data and detected speech to Open Sound Control⁴ commands which are sent to the python add-on client. For speech recognition we use the Microsoft Speech Platform SDK.

³ <http://www.blender.org/>

⁴ <http://opensoundcontrol.org/>

5 Use Cases

As a main use case for the animation loop station we see hobbyist and semi-professional animation production. At little cost, even a single artists can achieve animation results in short time spans. Professional studios could also make use of the technology to quickly create animated storyboards or initial sketches that can then be refined by other animation methods. Combined with high precision tracking devices, it is conceivable that the loop station approach will find its way into professional motion capture workflows.

Another interesting aspect is that by moving animation production closer to realtime it becomes more of an art form that can be consumed for its own sake. Rather than only watching the resulting animation as part of a motion picture or computer game, audiences might find satisfaction in watching the act of production itself. Artists could use the animation loop station as a new means of expression, creating entire animated sequences on the fly in real-time, based on audience feedback and intuition.

6 Challenges

The key idea of the animation loop station is to enable fast meta-control for capture and layering modes via a different modality than actual capture input so that a single user can do both without having to leave the “flow”. This approach does pose problems in certain scenarios. One such scenario is facial motion capture, where the same muscles for producing speech commands are used for creating capture input. Yet the overlap is minimal: After initiating capture with a speech command, facial motion capture can proceed unhindered. The spoken command for ending capture can be recognized and the relevant segment of the capture edited out automatically.

References

1. Dontcheva, M., Yngve, G., Popović, Z.: Layered acting for character animation. *ACM Trans. Graph.* 22(3), 409–416 (2003)
2. Leite, L., Orvalho, V.: Anim-actor: understanding interaction with digital puppetry using low-cost motion capture. In: *Proc. ACE 2011*. ACM, New York (2011)
3. Sturman, D.J.: *Computer Puppetry*. Computer Graphics in Entertainment (1998)
4. Walther-Franks, B., Herrlich, M., Karrer, T., Wittenhagen, M., Schröder-Kroll, R., Malaka, R., Borchers, J.: DRAGIMATION: Direct Manipulation Keyframe Timing for Performance-based Animation. In: *Proc. GI 2012* (2012)