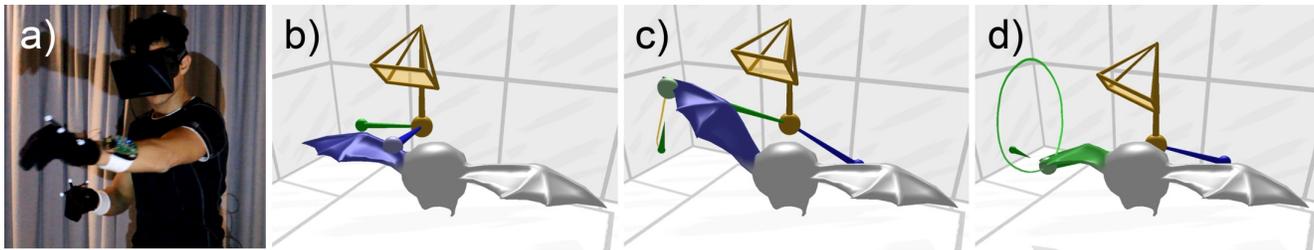


# Creature Teacher: A Performance-Based Animation System for Creating Cyclic Movements

Andreas Fender  
Department of Computer  
Science  
Aarhus University, Aarhus,  
Denmark  
andreasfender@cs.au.dk

Jörg Müller  
Department of Computer  
Science  
Aarhus University, Aarhus,  
Denmark  
joerg.mueller@acm.org

David Lindlbauer  
TU Berlin  
Berlin, Germany  
david.lindlbauer@tu-  
berlin.de



**Figure 1:** With *Creature Teacher* users can quickly create organic movements by successively selecting and manipulating parts of a virtual creature. (a) The user is wearing an HMD and pinch gloves. (b) Body parts are selected using the left hand. (c) Selected parts are manipulated using the right hand. (d) Periodic movements are detected and played back repeatedly.

## ABSTRACT

We present *Creature Teacher*, a performance-based animation system for creating cyclic movements. Users directly manipulate body parts of a virtual character by using their hands. *Creature Teacher*'s generic approach makes it possible to animate rigged 3D models with nearly arbitrary topology (e.g., non-humanoid) without requiring specialized user-to-character mappings or predefined movements. We use a bimanual interaction paradigm, allowing users to select parts of the model with one hand and manipulate them with the other hand. Cyclic movements of body parts during manipulation are detected and repeatedly played back - also while animating other body parts. Our approach of taking cyclic movements as an input makes mode switching between recording and playback obsolete and allows for fast and seamless creation of animations. We show that novice users with no animation background were able to create expressive cyclic animations for initially static virtual 3D creatures.

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. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

SUI'15, August 8–9, 2015, Los Angeles, CA, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3703-8/15/08 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2788940.2788944>.

## Categories and Subject Descriptors

H.5.m. [Information Interfaces and Presentation (e.g. HCI)]: Miscellaneous

## Keywords

Animation; 3D User Interface; Performance-Based; Virtual Reality; 3D Interaction and Graphics

## 1. INTRODUCTION

Animation is a powerful tool used in movies and games to give life to fictional characters and creatures. Those characters can convey emotions, motivations and intentions to viewers just by the way they are moving through space. This can be achieved by *cycle animations* which are repeatedly played back to make a character's locomotion look active. For example, when a game character is moving on the ground, a walking, sneaking or running cycle, depending on the current speed of locomotion, is played back repeatedly. Cycle animations are not only used for locomotion. Movements like breathing, chewing, dancing etc. are examples of cycle animations to be played back even when the virtual character is not changing location.

In general, 3D computer animation is implemented with virtual *articulated figures*, that is, hierarchical *limbs* are defined within the 3D model to deform it locally. With *joints*, we refer to the ends of the limbs or the connections of two limbs. The combination of all limbs yield the *skeleton* of the articulated figure. All limb transformations together yield the *posture*.

Creating convincing and expressive animations is challenging for various reasons. Depending on the number of limbs, a posture can have a large number of degrees of freedom (DOFs). Furthermore, animation introduces a time domain. Much knowledge and experience is required to create animations with state-of-the-art modeling and animation tools like *Blender* and *Autodesk Maya*.

In the recent years, an increasing number of 3D games include content creation by players as a core aspect. Many games enable players to easily personalize the look of their avatar or even to build whole worlds, thus those games are essentially heavily simplified and specialized modeling tools. Current content creation by players, however, is mostly limited to the specification of the appearance of game assets and does not include animation. As noted by Walther et al. [35], games generally do not provide expressive tools for players to create convincing animations. By now, animations of avatars and creatures inhabiting the game worlds are typically predefined or generated. We believe that animation can become a very important and expressive part of content creation by players. In our work, we want to take a step in this direction and allow users to playfully create animation without requiring expert knowledge.

An additional challenge is that due to the fictional nature of games, the virtual game characters can be of arbitrary morphology. Therefore we refer to them as creatures. If a game offers tools for the player to create such creatures, then their morphology might even be unknown at programming time. This makes the creation of specialized animation tools (unlike with specialized creation tools) even more challenging.

We developed Creature Teacher, an immersive virtual reality (VR) system to rapidly and easily create cyclic movement animations for virtual creatures. The main focus is the creation of expressive animations to be played back during locomotion, i.e., to specify how a creature moves through space. The system does not rely on preprocessing steps, specialized mappings or predefined animations. Equipped with a head-mounted display (HMD) and pinch gloves, users directly manipulate a virtual creature to easily create individual and expressive cyclic animations. Previous systems, while allowing for easy creation of smooth movements, often lack flexibility. We contribute by presenting a bimanual selection and manipulation approach to address the problem of providing expressive real-time control over different virtual creatures with arbitrary limb structure. We further present an approach to create cyclic animations by leveraging users' ability to perform repetitive motions and use those as an input method.

## 2. RELATED WORK

The two dominant approaches for computer animation are *keyframing* and *performance animation*, which are discussed in this section. Other techniques like inverse kinematics or behavioural animation exist, aiming to make the animation more realistic and simplify the creation process. However, those techniques are outside the scope of this paper since most of them are orthogonal (or potential additions) to the actual interaction techniques employed in our system.

Table 1 is an overview of previous animation interfaces for articulated figures, which we categorize by four dimensions. The first dimension specifies the input dimension (i.e., *WIMP*, *Sketch*, or *Tangible*), including tracking as input.

For tracking, we distinguish between *Tracking: Mimicking* (acting out movements) and *Tracking: Guiding* (controlling from a third person view). The second dimension specifies the handling of the animation time domain. This includes *keyframing* as well as different types of performance animation systems. *One-take systems* only need one recording take to create the complete animation, thus the animation time passes only once. With *multi-pass systems*, the animation time passes multiple times and users control different DOFs in each take. The other two levels specify if a system is immersive and the level of control. Level of control defines the granularity with which users can edit postures and movements. On one side, users have fine grained control over every single limb (i.e., *low-level*). On the other side, only pre-defined movements are remapped or movements are created automatically (i.e., *high-level*). Note that the table only contains animation systems for articulated figures and hence does not contain interfaces that only support rigid object transformations. Furthermore, the table does not account for pre- and post-processing steps.

### 2.1 Keyframing

With keyframing, only keyframe postures need to be created. The large number of DOFs of an articulated figure can be handled by users accurately one after another without animation time passing. This leads to the advantage that compared to the output, only a limited number of DOFs is necessary as input. Current state-of-the-art software contains sophisticated implementations of the keyframing technique. Those interfaces are very powerful, however, can be difficult to use for novices. In particular, making motions look organic is a very demanding task. Furthermore, keyframing is generally very time consuming, even for professional animators.

Besides established tools such as *Blender* and *Maya*, other user interfaces make use of the keyframing technique. The main focus of those interfaces is making the posing process easy and quick. The *HoloSketch* system is a stereoscopic sketching and animation interface with 3D user input [5]. Users can grasp virtual objects and apply transformations to pose and animate them. In contrast to our work, transformations can only be applied to whole objects rather than postures of articulated figures. With [20], [2] and [4], animations can be created by sketching the keyframe postures of the articulated figure by hand on a piece of paper. Eitsuka et al. [7] created an augmented reality based animation system, with which the user manipulates the key postures of a virtual character with his or her finger. Jacobson et al. [12] presented a modular input device, overcoming the limitations of specialized input devices. Single physical joints can be assembled and manipulated, making the system very flexible.

Even though those tools simplify the creation of postures, performance-animation tools often outperform keyframing approaches in terms of time required to create convincing animations.

### 2.2 Performance animation

In contrast to keyframing, performance animation systems rely on the user's physical timing and coordination skills. One well known technique is *motion capturing*: the performer's motions are mapped onto a virtual character. In most cases, a 1:1 mapping is used to achieve realistic

	<b>WIMP</b>	<b>Sketch</b>	<b>Tangible</b>	<b>Tracking: Mimicking</b>	<b>Tracking: Guiding</b>
<b>Keyframing</b>	<i>Blender Maya</i>	<i>MotionMaster</i> [20] <i>Chao</i> [2] <i>Davis</i> [4]	<i>Jacobson</i> [12] <i>Monkey</i> [8]		Eitsuka [7] <i>Osawa</i> [25]
<b>One-take</b>	Laszlo [18]	Motion Doodles [33]	Komura [15] Zhiqiang [21] Numaguchi [23] Kim [14] Krause [17] Mobile Animator [10]	Shin [28] Creature Features [27] Yamane [36] Vögele [34] KinÊtre [3] <b>ThatcherWorld</b> [30] Ishigaki [11]	SmurVEbox [1]
<b>Multi-pass</b>	Yamane [37] Kim [13]		<i>Oore</i> [24] Oshita [26]	Dontcheva [6]	Shiratori [29] Kostandov [16] <b>CAT</b> [22] <i>Creature Teacher</i>

**Table 1: Overview of interfaces for animating articulated characters. Bold entries are immersive. Italic entries allow for fine grained limb level control. Conventional interfaces can be found at *WIMP-Keyframing*.**

and natural human-like animations. The main drawback of motion capturing is the restriction to human or humanoid characters. Furthermore, only animations that are physically possible can be created directly. Digital puppetry [32], another performance-based approach, aims at overcoming these limitations by not using a 1:1 mapping.

### 2.2.1 Most similar systems

Shiratori et al. [29] created a digital puppetry plugin for *Maya*. Users can move and rotate a tracked device freely to manipulate the associated limbs of any articulated 3D figure. However, 3D input is only used for manipulation. Other tasks, like selection and setting up the mappings, is done through *Maya*. Therefore, users need a certain level of expertise, whereas our system is self-contained and does not need any setup of mappings. Kostandov et al. [16] presented a poster with the idea of an immersive VR character animation system with a direct manipulation interface. Using a 3D input device, users can grasp joints of an articulated figure and record the motions for subsequent playback. The system is suitable for non-humanoid characters, but only feasible for relatively simple limb structures. We build on their approach. However, our system is designed to also animate creatures with more complex limb structures. Martin et al. [22] presented a 3D input animation system called *CAT*, which lets users author animations by grasping and moving the limbs using a handheld controller. However, the system is specialized on quadruped creatures, relying on predefined and automated movements.

### 2.2.2 Performance-based: One-take

*Creature Features* [27] and other systems (e.g., [34]) focus on non-humanoid creature animations. Users are tracked and their movements are mapped onto the creature’s movements. The systems rely on a set of predefined animations for each creature to puppeteer in order to blend between them according to the performance. The system of Kim et al. [14] uses a haptic input device and virtual physically simulated marionette strings, to create complex character

animations. With *KinÊtre* [3], the object to animate is deformed according to the user’s performance. Yamane et al. [36] use human motion capture data and map the motion to non-humanoid creatures. While one-take animation systems can be advantageous in terms of creation speed, several challenges emerge. For example, movements must be physically possible for the actor to perform, otherwise specialized mappings or predefined animations need to be employed.

### 2.2.3 Performance based: Multi-pass

Kim et al. [13] proposed a performance-based desktop system. Users drag joints of physically simulated 3D-creatures with the mouse. Oore et al. [24] used two bamboo tubes for a bimanual real-time control of the selected limbs. In [6], movements of a handheld device are automatically mapped onto the virtual character’s movements. The multi-pass approach tries to balance creation speed and flexibility. A larger number of DOFs can be handled subsequently by creating superimposed animations, often in an additive manner.

## 3. WALKTHROUGH

We now illustrate *Creature Teacher* in the context of animating a game asset (see Figure 2). Since our system is intended for games in which content creation is important, giving users the ability to express individuality in the animations is a key point. In the example of this walkthrough, a skull with wings shall be animated. More specifically, animation cycles for flying forward, flying backward and flying on the spot need to be created by the player to specify how the skull can move through the game world. The starting point (or input) for *Creature Teacher* is a rigged 3D model of a creature, containing the mapping from limb transformations to mesh deformations.

Figure 2 depicts the rapid creation of one possible flying-on-the-spot animation. The player can give individual details to the movements. The individualism also comes naturally, since *Creature Teacher* is performance-based. Therefore, variances in the players’ hand movements or even flawed and unwanted movements make the end result unique.

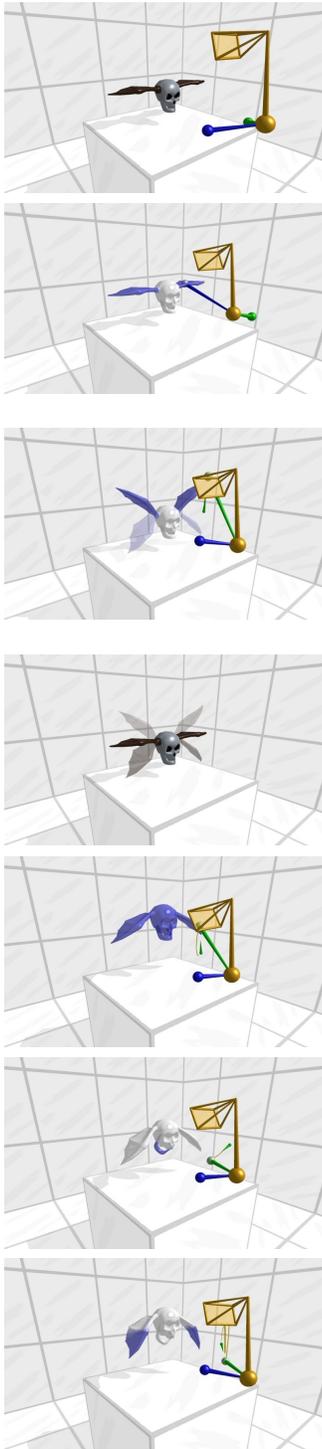


Figure 2: Walkthrough for the rapid creation of a flying-on-the-spot animation.

## 4. CREATURE TEACHER

This section provides details of interaction techniques, the workflow and some implementation details of Creature Teacher.

1. The user sees the virtual creature.

2. At first, the user selects both wings using the left hand (blue arm).

3. The wing's posture is manipulated using the right hand (green arm). The hand's motion is mapped onto the orientations of the wings in real-time. By moving the hand up and down, the user creates a wing-flap-motion.

4. After repeating the motion a couple of times, the motion is recorded. The skull repeats the motion from then on. Now other parts of the skull can be animated or the wing flap can be refined.

5. An up and down motion according to the wing flaps is given to the whole creature. The user has to select the whole creature and to move it up and down in the correct rhythm.

6. To add more life, a chewing-like motion is given to the mouth.

7. The user gives a delayed flap motion to the pinions to make the animation look more organic.

## 4.1 Overview

Users wear pinch gloves and move their hands in mid-air. The two hands are tracked and represented as two 3D volumetric cursors with different colors. Users see the cursors through an HMD and perceive the position at the actual hand positions. The two hands of the users have different roles, i.e., the non-dominant hand is mainly used for selection and the dominant hand is mainly used for manipulation (cf. [9]). For simplification, the right hand is considered as the dominant hand in this paper (however, Creature Teacher allows for configuring this).

## 4.2 Setup

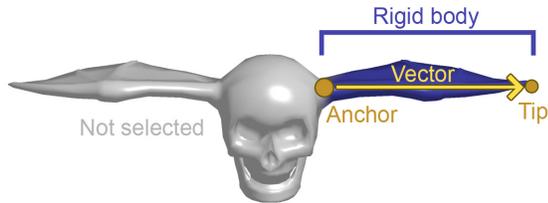
The main hardware components of the system are two custom made pinch gloves, an *Oculus Rift* (Development Kit 1) and an *OptiTrack* system. Markers are attached to each glove to be tracked by the *OptiTrack* system, which transmits position data to the Creature Teacher software. Further markers are attached to the *Oculus Rift* to get the camera position. For camera orientation, the build-in sensors of the *Oculus Rift* are used. The software is implemented using *Java*, 3D rendering is based on *OpenGL*.

## 4.3 Selection

The selection process starts as soon as the left hand is close to the creature. All previously animated movements of the creature are paused. The creature becomes semi-transparent and its texture is removed. This way, users can concentrate on animating the actual creature while gaining some insights into the underlying structure. We found this to be a good balance between low visual complexity and enough information to create animations. The body part closest to the left hand is highlighted. When pinching, the highlighted part is selected, which is visualized by a change in color of the part. Users can hold the pinch and move the cursor through the creature to select multiple parts in a row. Typically, we observed that users first coarsely reach into the figure to pause the movement, move the hand to the body part to select until the respective body part is highlighted and eventually start pinching. Users can freely select multiple body parts or extend the selection by pinching again with the left hand. Deselection is achieved by starting to pinch within already selected body parts. Users can pinch shortly outside of the range of the creature for complete deselection.

To enable quick selection of large structures as well as accurate selection of small parts, the effective volume of the selection cursor is proportional to the velocity of the hand movement. Slow movements allow for fine grained selection of single parts, since the cursor volume is small. When moving the hand very fast, large parts or even the whole creature can be selected very quickly without having to move through every single body part. This technique is inspired by the *Harpoon Selection* proposed by Leitner et al. [19], but applied to a 3D cursor instead of 2D pen input.

Selected body parts are shaded in blue. We refer to connected selected body parts as *rigid bodies*. The *anchor* of a rigid body is a point at the body part that has non-selected neighboring body parts. The *rigid body vector* is generated as following and depicted in Figure 3. The direction points from the anchor to the joints' center of mass within the rigid body. The magnitude equals the distance of the joint or the joints which are furthest apart from the anchor. Adding the rigid body vector to the anchor yields the *tip*.



**Figure 3:** Example for a rigid body: The whole wing is selected, indicated by blue shading. The rigid body’s anchor is at the transition between the head and the wing. Its tip is at the pinion.

#### 4.4 Manipulation

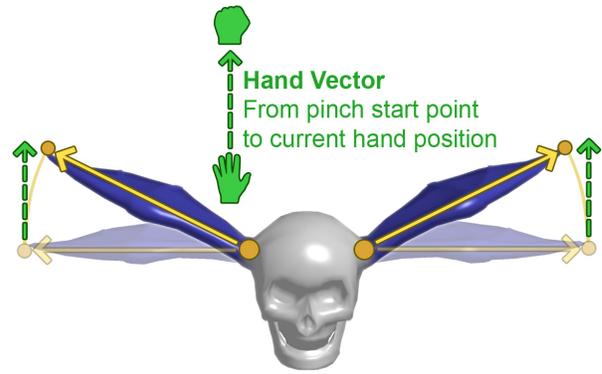
After selection, users can manipulate the created rigid bodies. The manipulation technique is related to the *Handle Bar Metaphor* [31]: Song et al. proposed using a virtual handle which is controlled bimanually to apply transformations on selected objects. Users shift two ends of a virtual handle to rotate, translate and scale it. The resulting transformation of the handle is applied to the selected object. The main motivation of the *Handle Bar Metaphor* is coping with low precision tracking, i.e., with technical inaccuracies. The *OptiTrack* system we use is rather accurate, but nevertheless the orientation control is much more fine grained, when using this technique instead of, e.g., hand rotation. In our system, every rigid body vector can be thought of being such a handle bar, whereas one end is at the rigid body anchor and the other end is controlled by the right hand when pinching. The result transformation is applied to all limbs of the respective rigid body.

To start the manipulation, users pinch with the right hand. Moving the right hand moves the tip of every rigid body, which changes the orientations of the rigid body vectors. All limbs of a rigid body are transformed accordingly. By default, the magnitudes of the rigid body vectors remain constant, therefore, even if users describe a straight line, the tip moves along an arc. Thus, only the orientation is changed. The basic principle is depicted in Figure 4.

If users start the manipulation at a rigid body tip, then the manipulation is collocated. However, from our experiences, it is more important for users to have a good overview of the creature than collocated manipulation. Therefore, users can find the best spot to start the manipulation and not necessarily have to start at the tip. Every rigid body tip is shifted according to the movement of the right hand, i.e., delta movements of the hand are applied to each rigid body tip. This behavior is especially important when multiple body parts are selected (see Figure 4). Furthermore, by not enforcing collocation, user fatigue is potentially reduced, because users can perform manipulations while being in a convenient posture instead of e.g., having to lean to the respective body part during manipulation.

In general, only 2 DOFs of the orientation are manipulated with the right hand, i.e., only yaw and pitch are changed. To enable roll, users have to additionally pinch with their left hand during manipulation. Moving the left hand around the right hand applies a roll transformation to all rigid bodies around their vectors.

If scaling is enabled (toggled through a 3D menu), then not only the orientation, but also the magnitude of the rigid



**Figure 4:** Manipulation with two rigid bodies: In this example, the user started to pinch with the right hand at an arbitrary point in the air and moved the hand straight upwards. The rigid body tips are both moved upwards according to the hand vector. The distance of the tip to the anchor, i.e., the magnitude of the rigid body vector, remains constant. Non-selected body parts are not influenced at all.

body vectors can be manipulated. By moving a rigid body tip towards the rotation anchor, the rigid body and its body parts become smaller. By moving it away, it gets bigger. In our current implementation, scaling is uniform.

If a rigid body has no anchor, e.g., when the whole creature is selected, then the movement of the right hand is applied to the translation of the rigid body, instead of the orientation. To rotate such rigid bodies, users need to pinch with the left hand during manipulation and to move the left hand around the right hand (cf. [31]).

Every created animation is additive, i.e., if the body part to manipulate is already animated, then the new motion is added to the prior motion. Therefore, users can first define a coarse motion of a large body part and then add fine grained motions to sub parts or vice versa.

#### 4.5 Loops

Movements are recorded only when users perform cyclic movements, thus no record button exists and no explicit mode-switching is required. Users can simply perform cyclic movements until the result looks satisfactory, which allows for fast trial and error.

During manipulation, movements of body parts are saved as soon as users perform a cyclic movement during manipulation. We refer to these cyclic movements as *loops*. As soon as the system detects a loop, the respective body parts are shaded green during manipulation. Users can continue the manipulation to improve the movement. Loops are confirmed by stopping to pinch with the right hand, or cancelled by shortly pinching with the left hand.

Only the most recent cyclic movement is saved. Previous loops are discarded whenever a new cycle is detected. We decided to not average the movement over previous valid cycles, because we believe that the recent loop has the highest likelihood to be the desired one. This way, users can release whenever they are satisfied with the result and the subsequent playback will look exactly like the last motion they have done during manipulation. The cyclic movements

of the selected body parts are then played back repeatedly, also while animating other body parts.

#### 4.5.1 Loop algorithm

Since robustly finding motion loops is not the focus of this paper, we will only coarsely describe how we detect loops in Creature Teacher. However, Creature Teacher’s system architecture allows for the implementation of different algorithms, including more sophisticated methods such as spectral analysis or dynamic time warping. With our current approach, finding loops is reduced to the task of finding the last movement the user did at least twice in a row. This simplifies the problem considerably, since there is no need to detect all cycles and patterns that have been created before. More specifically, if there was a recent repetitive movement, then it is not important if the same movement was done earlier during the manipulation.

When pinching with the right hand, a spatiotemporal curve is created to describe the motion of the right hand. The hand motion is sampled with a constant time interval and saved as an array of 3D points. Releasing and pinching again creates a new curve. No previous curves are taken into account.

Creature Teacher applies a continuous online loop detection during manipulation. For every input point, our algorithm searches for the closest of the previous input points. The closest point must have a distance smaller or equal than 3.5 inches. Furthermore, there must be points within the interval between the point pair with a larger distance. For example, two successive points of the curve have a close distance, but do not count as a pair, since there are no points with a larger distance in between. As soon as a point pair is found, Creature Teacher analyzes the curve between the found and the current point. Curves are marked as loops if two successive curve parts match each other in terms of duration. Additionally, every sampling point of one curve part within a loop must have a matching counterpart (i.e., a point within a distance of 3.5 inches) on the other curve part.

After a loop is detected, the motion of the selected body parts within the found time interval is copied to the resulting animation. Smoothing is performed to make the end point of the loop curve fit with the start point, thus allowing for seamless playback of the loop.

## 4.6 Posing

If users stop pinching before a loop was detected, the selected body parts remain in the manipulated position. Therefore users can seamlessly adjust postures and create cyclic movements without a mode switch. For example, users can first set the creature’s posture to then create a cyclic movement based on the posture. Body parts can still be adjusted after they are already animated, that is, a constant orientation can be added to cyclically moving parts.

## 5. DESIGN PROCESS

This section describes the design process of the system including the rationales of design decisions and some dead ends during development.

### 5.1 Immersive virtual reality

Because of the rise of immersive virtual reality and tracking devices in the consumer market especially in gaming, we

have chosen to utilize respective 3D interaction concepts to make 3D animation feasible. Using an HMD can be seen as the chosen navigation technique: The immersive setup frees the user from having to explicitly control the virtual camera. No mode switch or extra button is needed and thus users can fully concentrate on the interactions with the creature. The camera is controlled very naturally by moving and rotating the head while having the hands free. To avoid cybersickness, a 1:1 mapping from the physical movements to the virtual movements is used.

Besides the navigation, there are other advantages. For example, users get a better idea of the size and proportions of the virtual creature. The selection is collocated to utilize the proprioception abilities of users.

### 5.2 Visualization

We believe that it is very important that users seamlessly see the effects and results when editing the animation. If users are idle, the cycle animation keeps running and the creature is rendered in its original appearance with its texture and no additional visualizations. Many animation user interfaces render the skeleton within the mesh as an abstraction of the posture and to visualize the topology. In contrast, we do not visualize the skeleton to reduce the visual complexity and the cognitive load. Users are supposed to get a feeling of directly moving body parts instead of manipulating an underlying skeleton. Furthermore, a skeleton would possibly distract from local deformations of the creature’s body parts during manipulation. To visualize the topology, body parts near the left hand are highlighted as areas with blue shading during selection. This way, users can explore the topology with the left hand and also predict the selection. The selection is generally related to the *Silkcursor* of Zhai et al. [38], since it makes use of partial occlusions in a similar way. One difference to the *Silkcursor* is the fact that not the cursor, but the creature is semi-transparent. This way, users can see through the creature and reach parts which would normally be occluded.

### 5.3 Bimanual interaction

Instead of using a conventional input like a mouse, we decided for 3D input. The main reason is that users should have the possibility to freely describe 3D trajectories, instead of e.g., only trajectories on a plane. We decided against using specialized input devices to be flexible regarding topology of the creature and motions that can be created.

The main aspects that define the quality of an animation are timing and weight. Novices using a keyframe animation interface in general have trouble in giving the right timing and making the body parts look like they have weight. The resulting animations often appear unnatural or robotic. Using the hands however makes use of organic hand movements and inherent human skills. Users have a sense of timing and rhythm which they can express through the hands. Furthermore, users have a feeling of how objects with different weights move and can approximately simulate the trajectories with their hands.

Users continuously alternate between selection and manipulation, i.e., the left and the right hand are alternating between resting and being active. With this we aim at reducing user fatigue.

## 5.4 Cyclic movements

The initial idea was to intuitively teach movements to the creature. One way of teaching is repetition which led to cyclic motions as input methodology. With cyclic motions as input, users can simply perform motions repeatedly and refine them until the result is satisfactory. Most of previous work need explicit mode-switching for animation recording.

## 5.5 Iterations

Throughout the development, we constantly invited users to test our system. We observed interaction with the system and used a think aloud protocol. We also collected their feedback in open ended interviews.

The first prototype version of the system was a straightforward naive approach: All joints could be pinched individually with the left or with the right hand, that is, there was no prior selection. Therefore, no more than two joints could be controlled directly. When moving joints, the whole model reacted physically very similar to [16]. However, users found it to be very difficult to control, even with very simple skeletons. As the number of joints increased, interaction quickly became unfeasible with this technique. Therefore a technique for improved, simultaneous control over multiple joints was needed. With the selection technique, body parts are selected and when manipulating, the number of joints within the selection can be nearly arbitrary, since they all undergo the same transformations. Summarizing, the selection simplified the interaction significantly and made animating feasible.

Not all users preferred to walk around the figure to see it from different angles. Some even wanted to sit down. Therefore it is also possible to adjust the creature's orientation. Users can either stand and walk around the creature or sit down at a table and still reach all body parts by adjusting the orientation. A table potentially reduces fatigue since the non-active hand can rest on it.

## 6. USER STUDY

In order to explore the usability of Creature Teacher, we conducted a user study with 13 participants. The user study was done at the end of the iterative design process. The goal of the study was to get insights into the usage of the system and to evaluate the interactions and algorithms specific to our system, i.e., selection and manipulation of models and the creation of cyclic movements.

### 6.1 Participants

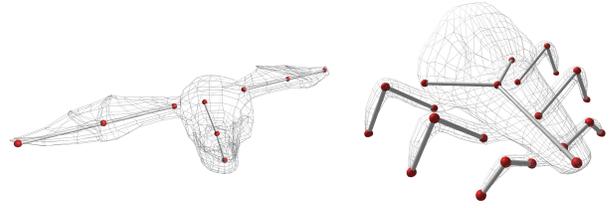
We recruited 13 participants (5 female), between 23 and 29 years old ( $Mdn = 26$ ) to test our system. All participants were right-handed (based on self-reports) and all of them had very little or no experience with using animation software. All participants had very little or no experience using immersive virtual reality systems and none of them had used Creature Teacher before.

### 6.2 Apparatus

The study was conducted in a quiet experimental room (room size approximately 4 by 3 meters). We used the apparatus as described in the Implementation section.

### 6.3 Procedure

Participants were introduced to the setup, followed by a 5 minute tutorial guided by the experimenter. Within this tu-



**Figure 5:** Users had to animate two models. The first one consists of 6 limbs and the second one consists of 15 limbs.



**Figure 6:** Sample user animations after using the system for the first time.

torial, participants selected and manipulated a simple pendulum (consisting of one limb) and were guided through the process of creating an oscillating movement of the weight of the pendulum around its pivot.

After the tutorial, participants created animations with two provided 3D models. The first model was a skull with wings (6 limbs, see Figure 5 left) for which participants were asked to create a flying-on-the-spot animation. Secondly, participants animated a walk-cycle with a roach-like creature (15 limbs, see Figure 5 right). No further instructions or templates on how the final animations could look like were given to the participants. Users created both animations multiple times and experimented with the system.

Participants were allowed to ask questions to the experimenter at any time. After completion, we conducted a semi-structured interview with questions on ease of use for each interaction technique (e.g., selection, manipulation, loops), rated on a 5-point Likert scale from 1 (strongly disagree) to 5 (strongly agree). Additionally, we encouraged participants to suggest improvements and missing features of Creature Teacher. Each session took approximately 60 minutes.

## 6.4 Results and Discussion

In general, participants quickly got used to the system and described the animation process as playful and enjoyable. In the following, we will describe feedback on the different interaction techniques included in Creature Teacher as well as results from the questionnaires.

### 6.4.1 Learning

After the guided tutorial, participants became familiar with Creature Teacher within approximately 5 to 10 minutes. The creation of the skull's flying-on-the-spot animation usually took between one and two minutes. This learn-

ing curve was expected and we observed this during our iterative design process with several other naive users. Participants became increasingly confident and were able to complete animations without guidance or help of the experimenter. Users were able to rapidly create their desired animation ( $Mdn = 4$ , "I was satisfied with the result animations."). They were able to fully focus on the outcome of the creation process and not the system itself. Users reported that they were working efficiently ( $Mdn = 5$ , "I had the feeling that I was working efficiently").

#### 6.4.2 Selection and Manipulation

In the beginning, most participants constantly mixed up the handedness of selection and manipulation. This changed after a short time and users felt that they understood the interactions and possibilities ( $Mdn = 5$ , "I understood how to interact with the system."). Some users first tried to select by pointing towards a body part from the distance, i.e., they expected a raycast selection. However, users quickly got used to the selection technique, which is reflected in the subjective ratings ( $Mdn = 4$ , "It was easy to select body parts."). Users first started with exploring the models using the left hand to highlight the selectable body parts. During manipulation, users explored the effects of the right hand to the selected body part to then create the actual movement. After some time of exploration, users stated that the movements got more and more purposeful. Again, this is reflected in high ratings of acceptance for the proposed manipulation techniques ( $Mdn = 4$ , "I was able to purposefully create movements.").

#### 6.4.3 Loops

Users reported that the loop detection was working well and they were able to achieve their desired results. Cyclic movements were in most cases detected when they were intended ( $Mdn = 4$ ). However, some users reported that small cyclic movements were not detected at all. We are working on this issue for future versions of Creature Teacher. Additionally, we adjusted the thresholds of our algorithms according to the results and our observations.

Finally, participants reported having no difficulties in repeating similar hand movements in order to create loops ( $Mdn = 4$ , "I was able to perform repeated hand movements for loop creation.").

#### 6.4.4 Problems

Most of the problems were related to technical issues. One major problem was, that due to the tracking inaccuracies, many users always felt an offset between the cursors and the hands in the immersive setup. This made coordination difficult in the beginning and needed some time of adaption. Especially one user, an amateur musical conductor, reported even small differences as interfering with his intents. Furthermore, users felt an offset and a slight delay between the real and virtual head movements.

#### 6.4.5 Conclusion

The goal of the user study was to give an initial validation of the system's quality. To us it was especially important to know if novice users are able to quickly create satisfying animations. One main aspect we wanted to gain insights into was if users are able to purposely do cyclic movements to create motions. The overall ratings and user opinions are

very promising. This was the very first time these users interacted with the system. Therefore we only provided creatures with a simple limb structure to let users concentrate on the interaction. More complex models were only tested informally during development. Advanced users will be the subjects of our future investigations.

## 7. LIMITATIONS AND FUTURE WORK

The system in its current state concentrates on rotating limbs, since in most cases this is sufficient for posing and animating. Further effort is required for the design and implementation to also fully support individual scaling along each axis and translation of individual limbs. However, depending on the application, it might not be necessary to support transformations other than rotation. Many other animation interfaces for novices are also only supporting changes in orientation.

Creature Teacher uses few input DOFs considering that it is performance-based, i.e., only 12 DOFs consisting of 6 DOFs for the head and 3 DOFs per hand are used. Only position tracking is needed for the hands, since the interaction does not rely on the hand orientations. This makes the system also realizable with simpler 3D input devices than the *OptiTrack* system. The limiting factors are the accuracy and the latency of the input device. However, tracking devices which become available in the consumer market are likely to soon allow for a cheaper solution than the *OptiTrack* system. Furthermore, to also be open to non-immersive setups, other output devices than an HMD could be used without major changes to the basic interaction. Therefore, to reduce the technological overhead, we would like to try out simpler input and output devices. However, one advantage of our current setup is the possibility to naturally walk around the virtual creature.

During the user study, we informally asked some of the participants to also try out Creature Teacher with a regular display positioned in front of them. While a formal comparison of immersive and non-immersive user experience is outside the scope of this work, comments hint the benefits of using an immersive virtual reality system. Most users especially pointed out the collocated selection when using the immersive setup.

With its current implementation, Creature Teacher only contains a very minimalist toolset to move the body parts. However, in general there are no restrictions in including common concepts used in many animation interfaces to make the creation process easier, quicker and the results more convincing. The most prominent example is inverse kinematics. By including inverse kinematics into Creature Teacher, certain animations would potentially be easier to create. One example would be the creation of a walking animation. With the current system, users first animate the upper leg, followed by the lower leg. With inverse kinematics, it would be possible to only grab the foot and guide it along an arc. The basic interaction techniques would remain the same, however, users would benefit from this addition. For the current system in its proof of concept state however, we did not include orthogonal techniques like these. We concentrated on the core interaction of the system for the sake of better validation.

By now, the target group of Creature Teacher are users with few or no animation background, like players of a 3D game. In general, the accuracy of the system as well as of

the users' hands is not high enough for professional animating. However, with the inclusion of animation techniques like mentioned above, the system might as well become interesting to professional animators, since they could quickly prototype animations. Different movement styles could be tried out quickly to use the best fitting one for subsequent post-processing steps.

## 8. CONCLUSION

We presented the immersive virtual reality animation system Creature Teacher. The system is used to rapidly create organic cyclic motions for non-humanoid characters without the need for predefined animations and specialized mappings. Our interplay of selection, manipulation and cyclic motion detection allows for fast and easy animation creation while still enabling high expressiveness. We have shown that users with little or no experience in animation were able to quickly create satisfying 3D cycle animations.

## 9. ACKNOWLEDGEMENTS

The authors would like to thank all the users who found many bugs and gave helpful feedback. This work has been supported by the ERC through grant ERC-2010-StG 259550 ("XSHAPE").

## 10. REFERENCES

- [1] R. Beimler, G. Bruder, and F. Steinicke. Smurvebox: A smart multi-user real-time virtual environment for generating character animations. In *Proc. VRIC*, 2013.
- [2] M.-W. Chao, C.-H. Lin, J. Assa, and T.-Y. Lee. Human motion retrieval from hand-drawn sketch. *IEEE TVCG*, 18(5):729–740, 2012.
- [3] J. Chen, S. Izadi, and A. Fitzgibbon. Kinêtre: Animating the world with the human body. In *Proc. UIST 2012*, pages 435–444, 2012.
- [4] J. Davis, M. Agrawala, E. Chuang, Z. Popović, and D. Salesin. A sketching interface for articulated figure animation. In *Proc. SCA 2003*, pages 320–328, 2003.
- [5] M. F. Deering. Holosketch: a virtual reality sketching/animation tool. *ACM Trans. Comput.-Hum. Interact.*, 2(3):220–238, 1995.
- [6] M. Dontcheva, G. Yngve, and Z. Popović. Layered acting for character animation. In *Proc. Siggraph 2003*, pages 409–416, 2003.
- [7] M. Eitsuka and M. Hiraoka. Authoring animations of virtual objects in augmented reality-based 3d space. In *Proc. IIAIAAI 2013*, pages 256–261, 2013.
- [8] C. Esposito, W. B. Paley, and J. Ong. Of mice and monkeys: a specialized input device for virtual body animation. In *Proc. I3D 1995*, 1995.
- [9] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior*, 19(4):486–517, 1987.
- [10] M. Gutierrez, F. Vexo, and D. Thalmann. The mobile animator: interactive character animation in collaborative virtual environments. In *Proc. IEEE VR 2004*, pages 125–284, 2004.
- [11] S. Ishigaki, T. White, V. B. Zordan, and C. K. Liu. Performance-based control interface for character animation. *ACM TOG*, 28(3):61:1–61:8, 2009.
- [12] A. Jacobson, D. Panozzo, O. Glauser, C. Pradalier, O. Hilliges, and O. Sorkine-Horning. Tangible and modular input device for character articulation. *ACM TOG*, 33(4), 2014.
- [13] J. Kim and N. S. Pollard. Direct control of simulated nonhuman characters. *IEEE Comput. Graph. Appl.*, 31(4):56–65, 2011.
- [14] S. Kim, X. Zhang, and Y. J. Kim. Haptic puppetry for interactive games. In *Proc. Edutainment*, pages 1292–1302, 2006.
- [15] T. Komura and W.-C. Lam. Real-time locomotion control by sensing gloves. *Computer Animation and Virtual Worlds*, 17(5):513–525, 2006.
- [16] M. Kostandov, R. Jianu, W. Zhou, and T. Moscovich. Interactive layered character animation in immersive virtual environment. In *Proc. Siggraph 2006 Research Posters*, 2006.
- [17] M. Krause, M. Herrlich, L. Schwarten, J. Teichert, and B. Walther-Franks. Multitouch Motion Capturing. In *IEEE Tabletops and Interactive Surfaces 2008*, 2008.
- [18] J. Laszlo, M. van de Panne, and E. Fiume. Interactive control for physically-based animation. In *Proc. Siggraph 2000*, pages 201–208, 2000.
- [19] J. Leitner and M. Haller. Harpoon selection: Efficient selections for ungrouped content on large pen-based surfaces. In *Proc. UIST 2011*, pages 593–602, 2011.
- [20] Q. L. Li, W. D. Geng, T. Yu, X. J. Shen, N. Lau, and G. Yu. Motionmaster: authoring and choreographing kung-fu motions by sketch drawings. In *Proc. SCA 2006*, pages 233–241, 2006.
- [21] Z. Luo, I.-M. Chen, S. H. Yeo, C.-C. Lin, and T.-Y. Li. Building hand motion-based character animation: The case of puppetry. In *Proc. CW 2010*, pages 46–52, 2010.
- [22] T. Martin and M. Neff. Interactive quadruped animation. In *Motion in Games*, volume 7660 of *Lecture Notes in Computer Science*, pages 208–219, 2012.
- [23] N. Numaguchi, A. Nakazawa, T. Shiratori, and J. K. Hodgins. A puppet interface for retrieval of motion capture data. In *Proc. SCA 2011*, pages 157–166, 2011.
- [24] S. Oore, D. Terzopoulos, and G. Hinton. A desktop input device and interface for interactive 3d character animation. In *In Proc. Graphics Interface 2002*, pages 133–140, 2002.
- [25] N. Osawa and K. Asai. An immersive path editor for keyframe animation using hand direct manipulation and 3d gearbox widgets. In *Proc. IV 2003*, pages 524–529, 2003.
- [26] M. Oshita. Multi-touch interface for character motion control using example-based posture synthesis. In *Proc. WSCG 2012*, pages 213–222, 2012.
- [27] Y. Seol, C. O'Sullivan, and J. Lee. Creature features: Online motion puppetry for non-human characters. In *Proc. SCA 2013*, pages 213–221, 2013.
- [28] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher. Computer puppetry: An importance-based approach. *ACM Trans. Graph.*, 20(2):67–94, 2001.
- [29] T. Shiratori, M. Mahler, W. Trezevant, and J. K. Hodgins. Expressing animated performances through puppeteering. In *Proc. 3DUI 2013*. IEEE, 2013.

- [30] M. Slater, M. Usoh, R. Geeas, and A. Steed. Creating animations using virtual reality thatcherworld: a case study. *Computer Animation*, 0:50, 1995.
- [31] P. Song, W. B. Goh, W. Hutama, C.-W. Fu, and X. Liu. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proc. CHI 2012*, pages 1297–1306, 2012.
- [32] D. J. Sturman. Computer puppetry. *IEEE Computer Graphics and Applications*, 18(1):38–45, 1998.
- [33] M. Thorne, D. Burke, and M. van de Panne. Motion doodles: an interface for sketching character motion. In *Proc. SIGGRAPH 2004*, 2004.
- [34] A. Vögele, M. Hermann, B. Krüger, and R. Klein. Interactive steering of mesh animations. In *Proc. SCA 2012*, pages 53–58, 2012.
- [35] B. Walther-Franks and R. Malaka. An interaction approach to computer animation. *Entertainment Computing*, 5(4):271–283, 2014.
- [36] K. Yamane, Y. Ariki, and J. Hodgins. Animating non-humanoid characters with human motion data. In *Proc. SCA 2010*, pages 169–178, 2010.
- [37] K. Yamane and Y. Nakamura. Natural motion animation through constraining and deconstraining at will. *Visualization and Computer Graphics, IEEE Transactions on*, 9(3):352–360, 2003.
- [38] S. Zhai, W. Buxton, and P. Milgram. The silk cursor: Investigating transparency for 3d target acquisition. In *Proc. CHI 2004*, pages 459–464, 2004.