

Article

The virtual lover: variable and easily guided 3D fish animations as an innovative tool in mate-choice experiments with sailfin mollies-I. Design and implementation

Klaus MÜLLER^{a,*}, Ievgen SMIELIK^a, Jan-Marco HÜTWOHL^a,
Stefanie GIERSEWSKI^b, Klaudia WITTE^b, and Klaus-Dieter KUHNERT^a

^aDepartment of Electrical Engineering & Computer Science, Institute of Real-Time Learning Systems, University of Siegen, Hölderlinstraße 3, Siegen, 57076, Germany and ^bResearch Group of Ecology and Behavioral Biology, Institute of Biology, University of Siegen, Adolf-Reichwein-Straße 2, Siegen, 57068, Germany

*Address correspondence to Klaus Müller. E-mail: klaus.mueller@uni-siegen.de.

Received on 30 March 2016; accepted on 19 October 2016

Abstract

Animal behavior researchers often face problems regarding standardization and reproducibility of their experiments. This has led to the partial substitution of live animals with artificial virtual stimuli. In addition to standardization and reproducibility, virtual stimuli open new options for researchers since they are easily changeable in morphology and appearance, and their behavior can be defined. In this article, a novel toolchain to conduct behavior experiments with fish is presented by a case study in sailfin mollies *Poecilia latipinna*. As the toolchain holds many different and novel features, it offers new possibilities for studies in behavioral animal research and promotes the standardization of experiments. The presented method includes options to design, animate, and present virtual stimuli to live fish. The designing tool offers an easy and user-friendly way to define size, coloration, and morphology of stimuli and moreover it is able to configure virtual stimuli randomly without any user influence. Furthermore, the toolchain brings a novel method to animate stimuli in a semiautomatic way with the help of a game controller. These created swimming paths can be applied to different stimuli in real time. A presentation tool combines models and swimming paths regarding formerly defined playlists, and presents the stimuli onto 2 screens. Experiments with live sailfin mollies validated the usage of the created virtual 3D fish models in mate-choice experiments.

Key words: computer animation, fish behavior, mate-choice experiment, research tool, virtual fish model.

Introduction

For around 80 years artificial visual stimuli have been used more and more in animal behavior experiments to investigate the importance of visual information in animal communication. One big advantage of using artificial visual stimuli instead of live stimuli is that morphology, coloration, and movement of artificial stimuli can be well defined and are independent of individual specific traits and

behavior. Additionally, the use of artificial visual stimuli helps to standardize stimuli used in behavioral studies and to “reduce” and “replace” stimulus animals regarding the 3Rs (Richmond 2010). The first experiments using artificial animal stimuli in fish behavior studies were conducted with dummies created from dead fish or wooden schematic models (e.g., Pelkewijk and Tinbergen 1937). With technical progress methods changed and improved over the years and researchers advanced and applied different techniques.

Baube et al. (1995), for example, used an electrical motor to move an artificial three-spined stickleback *Gasterosteus aculeatus* with a defined speed through an aquarium to imitate its natural movement during trials. The next step in “dummy evolution” were robotic fish. In contrast to previous dummies, robotic fish were able to swim along a programmed route (see e.g., Faria et al. 2010). Newer versions of robotic fish can also interact with the live fish (see e.g., Butail et al. 2013; Landgraf et al. 2014, 2016).

Screen-based stimuli

In contrast to robotic fish stimuli, screen-based stimuli (→ *screen-based stimulus* [for a better understanding of the technical parts of this article, we provide definitions of technical terms in the glossary (Table 2)]) provide only visual cues to focal fish and no tactile information (e.g., movement of water waves) or any chemical cues (e.g., material of the robotic fish) and cause fewer side effects such as noise of the mechanical drive system of the robot stimulus. The first screen-based stimulus experiments in fish used video playbacks presented on cathode ray tube (CRT) monitors positioned adjacent to a test aquarium. In this way, it was possible to show each focal fish the identical movements and behavior of the artificial stimulus fish (see e.g., Rowland et al. 1995; Rosenthal et al. 1996).

To get more control on stimulus' morphology and appearance, several researchers used video editing tools (→ *video editing tool*) that can also change the coloration (McDonald et al. 1995; McKinnon 1995) or fin size (Allen and Nicoletto 1997) of the stimulus. With such video editing tools, however, it is only possible to modify stimulus' coloration, morphology, and movement in a limited way. The 2D computer animations (→ *2D/3D computer animation*) offer a broader range of possibilities to manipulate stimulus characteristics. Here, a digital picture of a live fish is modified with the help of an image editing tool (e.g., GIMP [www.gimp.org] or Adobe Photoshop [www.adobe.com]; → *image editing tool*). With these tools it is possible to remove the background of the picture, make fins transparent, or change the coloration of the fish image. Once the fish image is complete, presentation software (e.g., Microsoft PowerPoint [www.microsoft.com]; → *presentation software*) is used to move the stimulus across the screen (Baldauf et al. 2009; Thünken et al. 2011; Fischer et al. 2014). This simple technique requires no special skills for animation, but is very limited in scope of rotating or bending of the stimulus. The use of 3D computer animations represents an even more powerful method. Pushed by the computer game and film industry, the development of professional and easy-to-use 3D animation tools boomed over the last decades and made them available for everyone, at a cost or even free to use. Researchers showed that 3D computer-animated stimuli can be used successfully to investigate different aspects of fish behavior in several species (see e.g., Chouinard-Thuly et al. 2017).

Computer animations—state-of-the-art

Since our presented novel toolchain includes 3D computer animation of fish stimuli, we first want to give an overview of the design and development from the past until today; subsequently we focus on the technical background used in these former studies.

Computer animations have been used in animal behavior research for more than 20 years since McKinnon and McPhail (1996) created a 3D computer-animated fish for the first time. They changed the coloration of throat and gill covers of virtual male

three-spined sticklebacks and analyzed the aggression level of live males in relation to stimulus coloration. Künzler and Bakker (1998) used a computer-animated male stickleback to test mating preference in live test females. They showed that their results obtained from experiments with animated stimuli corresponded to those of studies using live fish and video stimuli. A modified version of their initial stickleback animation was also used successfully in several follow-up studies (Bakker et al. 1999; Künzler and Bakker 2001; Mazzi et al. 2003; Mazzi et al. 2004; Mehliis et al. 2008). Morris et al. (2003) created a 3D computer-animated male swordtail fish *Xiphophorus continens* to test female preference for male coloration. The same computer animation was also used in (Morris et al. (2005) to test the relevance of male body size in female mate choice and Wong and Rosenthal (2006) used a swordtail fish stimulus to test the relevance of a male's sword for sexual selection.

To create and to apply 3D virtual animal stimuli requires detailed technical knowledge of software used for modeling and animating 3D objects. Therefore, in most previous studies computer animation experts were recruited to support experiments in behavioral research. This causes financial strain that is not affordable for every research group. Therefore, Veen et al. (2013) published a free and user-friendly version of a 3D fish animation tool called *anyFish* (<http://swordtail.tamu.edu/anyfish>). Ingleby et al. (2015) extended the first version to *anyFish 2.0* with new features to create fish animations including the opportunity to exchange fish stimuli and animations with other research groups (see also Chouinard-Thuly et al. 2017). Additionally, they published a manual, detailed video tutorials and provided the complete software to create a fish stimulus animation for free. Culumber and Rosenthal (2013) were the first who used *anyFish* to investigate a possible influence of mating preference on tailspot polymorphism in *Xiphophorus variatus*.

Butkowski et al. (2011) went a step further and extended the idea of animated stimuli by creating a virtual swordtail fish that could interact with live female swordtail fish *Xiphophorus birchmanni*. Thereby they overcame one of the biggest limitations of screen-based stimuli (no interaction with the live fish) and opened up many new opportunities for using this method in behavioral studies. They conducted mate-choice tests in which a live female could choose between courting conspecific and heterospecific males. During the trials, cameras tracked the horizontal position of the female and enabled the virtual males to follow her position on screen.

Process of animation

In general, the process of creating the animation and conducting the experiment can be divided into 3 steps: (1) creation of the virtual animal stimulus, (2) animation of the stimulus, and (3) presentation of the animation to the live animal test group on screen.

Creation of the stimulus

There are several different ways by which researchers can create a virtual stimulus. With the support of a professional animator McKinnon and McPhail (1996) created a virtual 3D male stickleback using morphological measurements and the texture (→ *texture*) of the stimulus extracted from a photograph. Künzler and Bakker (1998) used a different method. They fixated the body of a dead male stickleback in epoxy resin and cut it into 23 thin slices of 1 mm thickness each. They scanned each slice to create a digital image and

arranged the digital images in the correct order along the length axis to get the exact body shape of the fish model. They wrapped a texture around the model, which they had extracted from an image of the fish, and used digitized scans of fins for fin textures. A more widely used technique is to digitally remodel size, shape, and placements of fins based on images taken from live fish (Morris et al. 2003; Butkowski et al. 2011). In contrast to *anyFish*, Veen et al. (2013) did not create a single unmodifiable virtual fish stimulus, but rather created a generic model (→ *generic stimulus model*), which could be customized within *anyFish* based on morphometrical data (Ingleby et al. 2015). The current version of *anyFish*, however, is limited to only some fish species as it includes only generic models of sticklebacks, poeciliids, and zebrafish. To create the final fish model, 56 landmarks, which describe morphological points on the virtual fish, have to be mapped to a lateral image of a live fish. *anyFish* then automatically creates the virtual fish model according to this information, representing shape and texture.

Animation of the stimulus

In addition to morphology and appearance, the natural movement of the stimulus is very important to be recognized as, for example, a “conspecific” by live test fish (e.g., Nakayasu and Watanabe 2014). In previous studies, researchers used different methods to create movements as realistic as possible. Künzler and Bakker (1998) derived the movement from a video, showing the courtship display of a live male stickleback, by using rotoscoping (→ *rotoscoping*). Morris et al. (2003) used a similar approach. They studied the movement of male swordtail fish in over 2700 video frames (→ *frame (video)*). They determined the duration of a certain behavioral pattern by counting the number of frames showing the behavior and then animated the same pattern with the virtual stimulus by using the same number of frames. Veen et al. (2013) and Ingleby et al. (2015) included different options for animating the stimulus in their software tool *anyFish*. The first option is to open a pre-existing path, which can be modified and customized by the user. The second option is the so-called key-frame animation (→ *key-frame animation*). In *anyFish*, this method can also be combined with rotoscoping. The third option is to apply motion capture data (→ *motion capture*) generated by third party software.

Since the virtual fish stimuli of Butkowski et al. (2011) had to be rendered (→ *rendering*) in real time according to the tracking data (→ *tracking*) of the live fish, the former described methods were not suitable for their animation. Instead, they created a library of 24 key movement animations, which, for example, included forward swimming or turning. Key movement animations could be combined to generate a complete movement sequence of the stimulus. The animation system was then connected to a 2D tracking system that tracked the position of a live fish swimming in an aquarium positioned between 2 screens. Depending on the tracking information, the animation was rendered in a way that the virtual fish followed the live fish in real time.

Presentation of the animated stimulus

In the last step, the animation is presented to the live test fish. In most cases the animation is prerendered (→ *rendering*) into a video file format that can easily be played back on video recorders (McKinnon and McPhail 1996) or using software, for example, VLC media player (<http://www.videolan.org>), as in Veen et al. (2013) and Ingleby et al. (2015). In contrast, Butkowski et al. (2011) did not render their animation to a video file. Due to the real time

interaction of the virtual stimulus with live fish it was not possible to use a prerendered animation. As the animation had to react in response to the live fish’s movement, it had to be directly rendered onto the screen. A most recent review on detailed conceptual and technical considerations for design, animation, and presentation of animated animal stimuli can be found by Chouinard-Thuly et al. (2017).

A novel toolchain for fish animations—beyond the state of the art

Here we introduce and describe in detail a novel toolchain (→ *tool-chain*) to conduct behavioral experiments with realistic 3D fish stimuli in a very user-friendly way (Gierszewski et al. 2017). Our software is based on a robot operation system called ROS (*v. fuerte*, www.ros.org; → *Robot Operation System*), which shows a high degree of modularity and flexibility. Additionally, it can easily be extended by other modules like fish tracking that is necessary for interactive animations. Our software consists of 3 different tools: (1) *FishCreator* for stimulus design, (2) *FishSteering* for animation, and (3) *FishPlayer* for presentation. Additionally, we developed (4) *FishSim* for the visualization (→ *visualization*) of the stimuli (Figure 1a–e).

FishCreator is a tool to design virtual 3D fish stimuli. With *FishCreator* it is possible to change size, morphology, and texture of a body and its fins in a very easy way. It is also possible to generate random-textured stimuli to avoid human bias. To generate a random-textured stimulus, *FishCreator* chooses a texture image for the body and one for each fin randomly from a library of previously prepared textures.

With *FishSteering* we developed a novel method to animate the created 3D virtual stimuli. Here, the user can steer the stimulus with the help of a video game controller (→ *video-game controller*) in real time in contrast to *anyFish* that is not currently able to render animations in real time (Veen et al. 2013; Ingleby et al. 2015). Swimming and turning movements are automatically calculated by a self-designed algorithm based on the analysis of video material showing live sailfin mollies swimming in a test tank (Smelik et al. 2015). With *FishSteering* it is possible to animate 1, 2, and even more stimuli simultaneously, and the recorded movement patterns can be applied to all already existing stimulus models, created with our *FishCreator*. Thus, our software is not limited to one or a few fish species.

The *FishPlayer* tool is designed to present previously recorded animations on 2 screens that are positioned adjacent to a test tank during experiments, for example, in a typical binary choice situation (Gierszewski et al. 2017). *FishPlayer* can present all recorded movements with all stimuli in any combination and sequence, which is a novel feature compared to previous animation studies. Our toolchain itself is currently not released to the public but it is planned to provide a free version in the near future.

The Sailfin Molly Case Study

In this section, we describe our novel toolchain in more detail. Additionally, we give a general overview of the communication between the different tools and show how these are connected to each other. We present the toolchain on the basis of a case study with sailfin mollies *Poecilia latipinna* but it can be applied to other fish species as well. Our method was already successfully used and

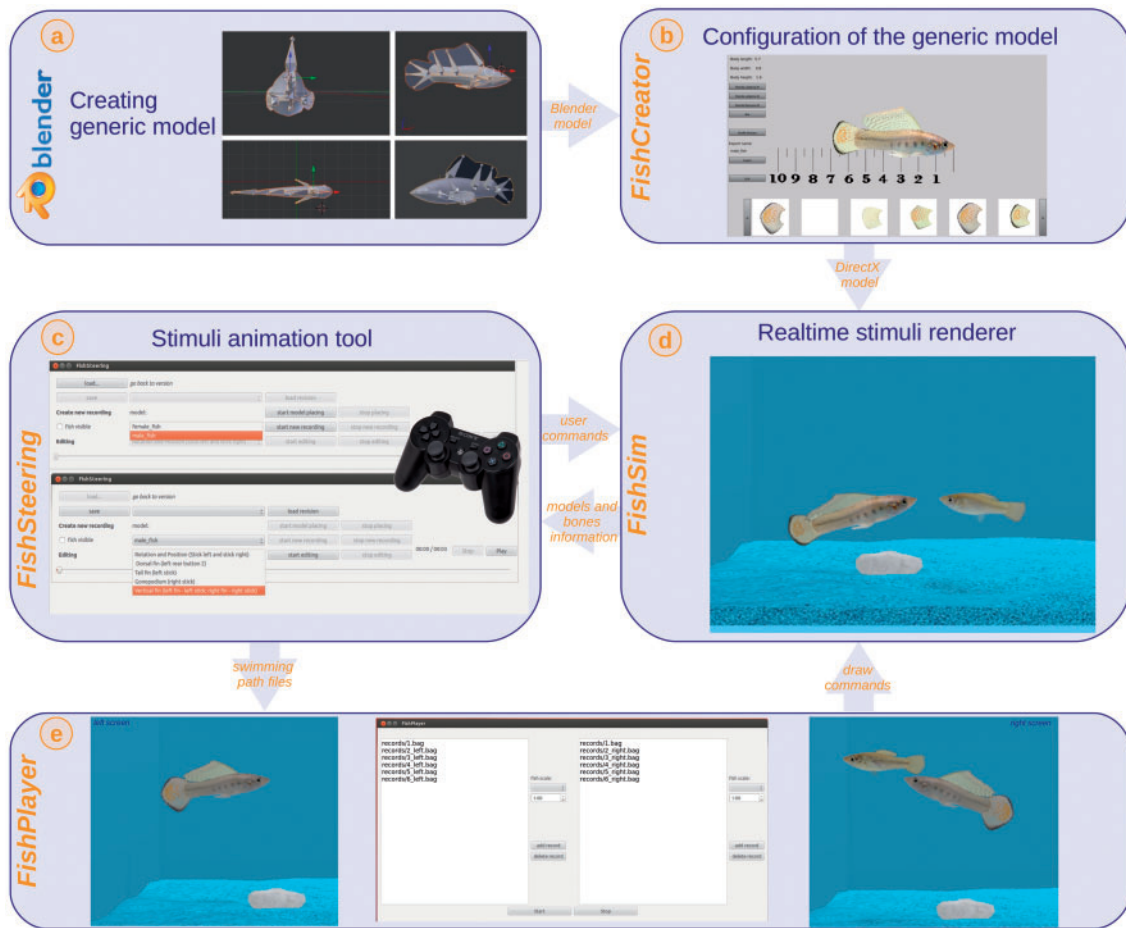


Figure 1. Overview of the toolchain. (a) Design of the generic model was done in Blender. (b) Configuration of the generic model with *FishCreator*. The tool offers different textures for fins and the body. (c) Stimulus animation with *FishSteering*, in which the user is able to steer the stimulus with the help of a game controller. (d) Real-time stimuli renderer *FishSim*. *FishSim* presents the stimuli on screen. The virtual tank environment in *FishSim* was designed according to the real test tank. (e) Presentation of the stimuli with *FishPlayer*. *FishPlayer* has individual playlists for each screen (left and right).

validated in mate-choice experiments with live sailfin molly test fish (Gierszewski et al. 2017).

General system overview

Our system is based on the robot operation system ROS. A robot system consists of sensors (\rightarrow *sensor*), manipulators (\rightarrow *manipulator* (*robotics*)), controllers, and human machine interfaces (\rightarrow *human machine interface*). All components have to work together, even if they are distributed over several computers, and thus have to communicate which is provided by a middleware (\rightarrow *middleware*) like ROS. As modern stimulus animation systems have requirements similar to robot systems, we decided to use ROS as the middleware for our toolchain. In addition to the stimulus animation, which should be shown on multiple screens, a tracking system for automatic measurement of behavior and interactive animations (as used by Butkowski et al. 2011), an information screen for the experimenter, and controllers for steering the stimulus can be part of such systems (\rightarrow *interactive computer animation*).

All components presented here are modular and can run as 1 system distributed over several computers. *FishSim* is the central tool to visualize the animation. To bring the animation to screen, the *FishSim* program has to be started for every screen separately.

Several *FishSim* programs can run on the same computer or even on different computers connected via network.

3D stimulus design

FishCreator is based on the free and open-source (\rightarrow *open-source*) 3D computer graphics software *Blender* (v. 1.71, www.blender.org) and on its game engine (\rightarrow *game engine*). Similar to Veen et al. (2013) and Ingley et al. (2015) it is necessary to predefine a generic model of the fish that will later be customized in *FishCreator*. Until now we have implemented 4 different generic models: a male and female sailfin molly, a male Atlantic molly *Poecilia mexicana* and a geometrical box model. If needed, the model selection in *FishCreator* can be complemented with additional generic models of various fish species.

Generic 3D model design

In the following section, we describe the creation process for a generic model on the basis of sailfin mollies (Figure 1a), but all steps can be performed accordingly for every other fish species as well and even for other animals. Rosenthal (2000) defined 2 different types of stimulus design: the “exemplar-based animation” and the “parameter-based animation.” Our generic model is a combination of both. We used a lateral image of a live fish for each sex to outline

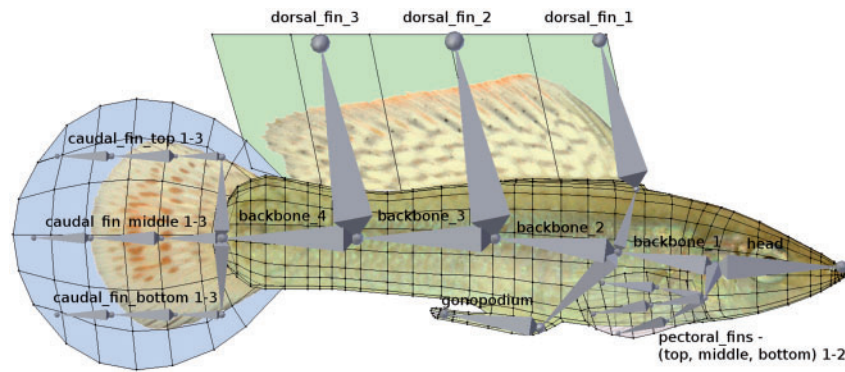


Figure 2. Generic mesh model of a male sailfin molly with textures and bones. The mesh represents the morphology of the fish body. A single gray cone describes a bone of the virtual skeleton. Body mesh and fin planes (blue—caudal fin, green—dorsal fin) are wrapped in textures.

the 2D shape of the model. To get a 3D model mesh (\rightarrow *mesh (polygon)*) we used parameters like thickness, length, and height from exact measurements of the live fish.

Each fin (dorsal, pectoral, and caudal) consisted of a plane (mesh) that holds the texture. The planes were larger than the standard size of a fin to offer enough space for even larger fin textures (Figure 2), for example, to create stimuli outside the natural range. Parts of these planes that are not covered by a texture are not considered by the render system and hence not visible in the final animation. We positioned the fin planes according to the picture of the lateral fish to the body of the model (Figure 2). We designed the eyes of the model separately by adding 2 hemispheres with a high transparency to make them look like real lenses.

For animating the model, we added a standardized virtual skeleton with 36 bones (\rightarrow *bones*), the minimum number of bones that was needed to provide natural swimming movement of the model (Figure 2). For the sailfin molly, we specifically added 2 additional bone groups: 3 bones for raising and lowering the dorsal fin, which is an important part of the courtship display in sailfin molly males, and 1 bone for moving the gonopodium, a copulatory organ in males in Poeciliids. In general, it is possible to add any kind of bone to the virtual skeleton to design movement characteristics of other species as realistically as possible.

Each bone is connected to the surrounding area of the model mesh and hence able to displace this area when the bone is moved. For example, the dorsal fin bones were connected to the surrounding dorsal fin plane (Figure 2). This allows movements of the dorsal fin by moving the dorsal fin bones. In *Blender*, this connecting process is called weight painting (\rightarrow *weight painting*) and can be done either manually or automatically. Following this procedure, we also designed a generic female model and a heterospecific *Poecilia mexicana* model. Morphology of the mesh as well as textures were model specific but the skeleton was identical, except for the bone moving the gonopodium, that was deactivated in the generic female model. We also designed a 3D box model that had no skeleton (could not be bent while moving) but could also be equipped with a texture. The generic models were used as basis for the customizing process in *FishCreator*, as described below.

Generation of skin and fin textures

To color the skin of the 3D models, we created so-called textures of body and fins by cutting out the respective parts of fish images and mapped them onto the 3D model by UV mapping (\rightarrow *UV mapping*). For UV mapping, we exported UV map templates from the modeling

software *Blender*, which represent the outlines of the respective model parts (body and single fins) that have to be textured.

Body and fin texture design was done using lateral fish images, with fins spread. Fish were photographed (Canon EOS 600D, F/5.6, 1/250 s, ISO 200, focal distance 55 mm) in a small tank 40 cm \times 40 cm \times 12 cm; water level 10 cm height). Illumination was provided by 2 neon tubes above the tank and an external camera flash (Nissin Speedlite). To adjust the white balance in a standardized way, it is recommended to add a reference color object, like a graycard, to the scene. Pictures were then imported as RAW format using *UFRaw* for the image editor *GIMP* (v. 2.8) to adjust for white balance, illumination and contrast to get the best quality textures. We added a white plastic board inside the tank, positioned in short distance parallel to the front, so that fish were able to swim freely but were forced to align laterally.

We then cut bodies and fins out of these images by using *GIMP* and fitted the textures to the former created UV map templates. Textures could be fit perfectly onto the UV map templates using several *GIMP* functions such as rotate, scale, and cage-transform (\rightarrow *cage-transform*). As real fins are transparent, fin textures were also modified to look transparent on the 3D model. In general, the user has multiple options given by common image editing tools (e.g., *GIMP* or *Adobe Photoshop*) to change coloration of textures, add additional patterns to textures such as dots or stripes, or manipulate the texture in any other way. Generated textures were saved as PNG (Portable Network Graphics) files. *FishCreator* automatically detects these textures and includes them to its library as described in the following section.

Customization of the generic model with *FishCreator*

FishCreator is a tool for generic model customization that runs in the *Blender game engine* (Figure 1b). The user can easily design the virtual fish in a user-friendly graphical user interface (GUI) (\rightarrow *graphical user interface*) in which the generic model species can be selected. In the current version of *FishCreator*, 4 different generic models can be selected: male and female sailfin molly, Atlantic molly male, and a 3D box. The model is loaded with its default dimensions (cm) for length, height, and thickness, but the user can change all these dimensions. In the next step, the user can assign textures onto the model by clicking on the body part to be changed like the body, the dorsal fin, the pectoral fin, and the caudal fin. Available textures, that were previously created and saved in the *FishCreator* folder, are listed at the bottom of the GUI. In addition to this option for user-based model design, we also added a function to texture the

Table 1. Editing options for the animation of different body parts

Animation of/correction of	PS3 controller button/joystick	Possible range of animation
Dorsal fin	Left rear button (infinitely variable)	0° to 90° up
Tail and caudal fin	Left joystick	-30° to +30° to the side
Gonopodium	Right joystick	-170° to +170° to the side
Pectoral fin	Left joystick left fin, right joystick right fin	0° to 90° to the side

model randomly to get an independently designed model to avoid human bias in behavioral studies. This function selects textures for body and fins randomly from the texture folder and maps these to the model. The last step of the customization process is to export the final model that is then ready for presentation with *FishSim*.

Visualization and the virtual environment—*FishSim*

FishSim is the central component in the animation toolchain (Figure 1d). It visualizes stimuli and their movements and is based on the free and open-source computer game engine *Irrlicht* (v. 1.81, <http://irrlicht.sourceforge.net/>). When starting the program, it automatically loads all available models from the model folder including the complete skeleton structure of the stimuli. *FishSim* internally stores the bone structure and can rotate the bones around x-, y-, and z-axes. We integrated *FishSim* to the infrastructure of ROS. In this infrastructure, every program is referred to as a *node* (→ *node*). These *nodes* can receive messages from and send messages to other nodes. In this case, the *FishSim* node receives messages, which include position and orientation of each stimulus, and renders these accordingly. In the current version, *FishSim* renders the stimuli 60 times per second (fps) to the screen, a frame rate (→ *frame rate (fps)*) that is sufficient for the visual system of most fish (Fleishman and Endler 2000; Oliveira et al. 2000). But frame rate can easily be adapted to more sensitive visual systems of other animals and to requirements for different screen types (see also Chouinard-Thuly et al. 2017).

For presentation during experiments, the fish stimuli were simulated to swim in a virtual fish tank environment (Gierszewski et al. 2017). To make the animation more realistic, the dimensions of the virtual tank, the wall color as well as the ground texture were adjusted to be analogous to the real experimental tank containing the live test fish. For illumination of the virtual aquarium we used the game engine's feature to include virtual light source, which was mounted "above" the virtual fish tank to resemble illumination of the real test tank and to keep illumination differences as small as possible. To increase the realism of the animation we also used several methods to simulate depth. In addition to shadow effects (Veen et al. 2013) and occlusion cues for the stimuli (Zeil 2000; Veen et al. 2013), we also added objects (in our case stones) as a perspective reference (Mehlis et al. 2008).

Stimulus animation—*FishSteering*

FishSteering (Figure 1c) is the tool to animate the stimuli. It is also developed as a ROS *node*. To make the animation process as easy as possible we developed a semi-automated steering (→ *semi-automated steering*) mode controlled with a Sony PlayStation controller (*DualShock 3*, www.playstation.com). With *FishSteering* it is possible to steer a stimulus in *FishSim* directly in real time or to record the movement commands to a file for later presentation. *FishSteering* communicates with *FishSim* automatically and gets a list of all stimuli and their bones currently loaded in *FishSim*. All stimuli are animated successively. Before the steering process starts, the user can place the stimuli in any starting position in the virtual

tank. The swimming path of the first stimulus is then created by steering it within the virtual tank with the game controller. After the first swimming path of the first stimulus is done, one can jump back to the beginning of the animation, choose the next stimulus and start to animate this. During the animation of the second stimulus, the former animated stimulus and its swimming path are replayed and shown in *FishSim*, so that the user always has an overview of all animated stimuli and their respective swimming paths. Thus, it is possible to coordinate movements of 2 or more stimuli to simulate different behavior, for example, following, courtship or avoidance. There are nearly no limitations in the number of animated stimuli that can be steered and presented in 1 animation.

To animate swimming movements of fish stimuli, we adopted an algorithm described by Smielik et al. (2015), who analyzed videos of swimming sailfin mollies with computer vision algorithms (→ *computer vision*) and extracted the swimming movement of the fish body. They estimated the extracted movements of the fish with polynomial spline interpolation (→ *polynomial spline interpolation*) and created a special function. This function continuously calculates the angles of the backbones during the swimming process according to speed and turning angle. For our generic sailfin molly male model shown in Figure 2, it calculates the angles between all *caudal_fin* bones, all *backbone* bones, and the *head* bone.

In general, *FishSteering* supports 3 different swimming animation modes: (1) forward swimming, (2) curved forward swimming, and (3) on-spot turning. All these modes can be easily controlled with the controller. The swimming speed can be continuously varied from 0 to 40 cm/s. The turning speed for curved swimming and on-spot turning can be selected from 0°/s to 200°/s and the turning speed for up and down swimming from 0°/s to 30°/s. These swimming and turning speeds were defined based on video analysis and personal behavioral observations, and specified for the use of sailfin molly models. In general, the swimming speed can be adjusted individually to any species.

After the swimming path of the stimuli is created, the user has the option to animate some specific parts like dorsal fin, gonopodium or pectoral fin or to improve the formerly created animation in a second cycle, for example, to simulate courtship behavior at certain points in the animation. Table 1 shows the currently implemented options for editing the sailfin molly models. The joystick/button positions are directly mapped to the bone angles of the respective part of the stimulus (e.g., angle of dorsal fin to raise or lower it during courtship).

Presentation—*FishPlayer*

FishPlayer is the tool to visually present created animations during experiments. It was designed for stimulus presentation in dichotomous test situations, for example, mate-choice tests, typically used in animal behavior experiments. *FishPlayer* was developed to replay 2 animations on 2 different screens at the same time, either 2 identical animation sequences on both screens, or different animations on each side. Nevertheless, *FishPlayer* can also be used with only 1 monitor if needed, for example, for sequential choice tests.

Table 2. Glossary: detailed description of the technical terms used in the text

2D/3D computer animation	Computer animation describes the process of generating animated images (frames), concatenated to videos. In contrast to 3D computer animations, 2D animations are based on a 2D geometry and do not take depth perception into account. Typical examples of 2D computer animations are cartoons. The 3D animations are more complex and more realistic and based on a 3D geometry. With these it is possible to create depth perception for the observer.
Bones (animation)	Within the context of computer animation a bone is a structure inside a model mesh similar to a bone inside a human body. It is connected to the surrounded 3D model mesh. All transformations and rotations of the bone directly affect the connected polygon mesh regarding the predefined weight (\rightarrow <i>weight painting</i>). With these it is possible to move whole mesh groups like an arm of a human by just moving 1 bone.
Cage transform	Special editing tool in <i>GIMP</i> . With this, the user can select an area of an image and can push or pull the borders of this selection to deform it.
Computer vision	Computer vision covers the field of automatic processing, analyzing, and understanding camera images with help of a computer.
Frame (video)	In context of video and animation, a frame is a single image within the sequence of moving images of a video or animation.
Frame rate (fps)	Indicates the number of displayed images (\rightarrow <i>frames</i>) within a sequence of images of a video or animation per second.
Game engine	A game engine is a software framework for computer game development. In general, it provides tools for game design, development, and execution. During execution it renders (draws) the 3D virtual scene on screen.
Generic stimulus model	A generic stimulus model is a generalization of a fish stimulus. It can be configured regarding size, morphology, and texture to generate a unique new stimulus of the used species.
Graphical user interface (GUI)	A GUI is a screen-based control panel to interact with the computer. The information on the computer is visualized with graphical elements. The user can select these elements to control the computer. A classical GUI is, for example, controlled with the help of a computer mouse.
Human machine interface (HMI)	A HMI enables human to interact with computers. With this, a human operator can control machines, whereas the machine sends feedback back to the operator. Examples of HMI are GUI command line interfaces or web-based interfaces.
Image editing tool	With the help of image editing tools the user can manipulate digital images. For example, it enables the user to resize, recolor, crop, or retouch images. For a useful list, see Chouinard-Thuly et al. (2017) .
Interactive computer animation	In contrast to classical computer animations, interactive computer animations are generated in real time and can react or respond to input signals.
Key-frame animation	A method to animate objects or models. The user defines object parameters like position or orientation at a certain time and the computer calculates and generates the frames between these “key-frames”. This method reduces the effort of the animator as he do not have to define every frame of an animation but just significant positions.
Manipulator (robotics)	Mechanical device that enables robots to interact physically with the environment.
Mesh (polygon)	A polygon mesh represents the surface of a 3D object. It consists of edges and vertices, connected to a kind of net.
Middleware	A middleware is a software that is located between the computer operation system and the application or service. It provides several tools, which simplifies the development process of applications.
Motion capture	Process of analyzing and recording the movement of objects, animals, or humans.
Node	A node is a software application within the middleware <i>ROS</i> (\rightarrow <i>middleware</i> , \rightarrow <i>ROS</i>).
Open source (software)	The programming source code of open-source software is free available and can be modified by the user. In contrast, closed software can just be used but not reprogrammed.
Polynomial spline interpolation	Mathematical approach to approximate a mathematical function with the help of piecewise polynoms (splines) out of predefined points.
Presentation software	Presentation software is used to create graphical presentations and to present these presentations to the audience.
Rendering	Rendering is the process of converting virtual 3D models or whole 3D scenes to a 2D image. This is necessary to display the object or scene on a 2D screen. <i>Prerendering</i> : Since the rendering process can be very computationally intensive and consequently very time-consuming, the images are rendered time-independent and stored in a video file. The final video can be played faster so that the animation runs fluently. <i>Real-time rendering</i> : In contrast to prerendering, real-time rendering has the possibility to modify the animation during runtime. It is used for computer games or applications where the animation depends on input parameters. In general, real-time rendered animations are not that complex, so that it can be rendered faster.
ROS	Middleware (\rightarrow <i>middleware</i>) for robotic purpose. It provides several tools and features, helping developers to program and execute software for robots. In addition to its application in the field of robotics it also has a lot of benefits for all areas of application, where sensors (e.g., cameras) are included or software applications are distributed to multiple computers and a network communication is necessary.
Rotoscoping (animation)	Method to animate objects or models. It describes the process of deriving the movement of an object from a video by adjusting the virtual counterpart manually frame by frame so that the overlay of the animation is congruent with the object video.

(continued)

Screen-based stimulus	A stimulus, which is shown on screen.
Semi-automated steering	Describes a steering mode, which is mostly automated but still needs user input. This mode lightens the user's workload.
Sensor	Device, which detect or analyzes some type of input from the physical environment. The input can be, for example, light, sound, or temperature.
Texture (animation)	In animation context, texture means an image, which is mapped to the 3D object or model surface.
Toolchain (software)	A toolchain is a set of software tools to perform a specific task. Every single tool of the toolchain can be used to solve a specific problem. The output of 1 tool is normally the input of the following tool, so that the tools are functionally connected like a chain.
Tracking (video)	Video tracking is a method to locate a moving object over time (frame by frame) in a video sequence. This can be done in 2D and 3D space. In case of 3D tracking, the tracking system calculates the 3D position of the object and—if required—the 3D orientation.
UV mapping	UV mapping describes the process of mapping a texture to a surface of a 3D object. To place a 2D texture to a 3D surface, the 3D object surface gets unfolded to a 2D plan. Since x , y , and z are already used to describe the axes of the 3D object space U and V are introduced to describe the 2D coordinates on the unfolded object surface. The texture position regarding the surface plan is defined in U and V coordinates.
Video editing tool	With the help of a video editing tool the user can manipulate video sequences. For example, it enables the user to cut videos, change coloration, or add subtitles.
Video game controller	A hand-held input device to control, for example, video games.
Weight painting (animation)	Describes the process of connecting the object mesh with the object bones. Since not all parts of the mesh should move regarding a specific object bone in the same way, the designer can use the weight painting function to define the relative bone influence to the related polygon mesh.

FishPlayer automatically starts a *FishSim* node for each screen. The user can choose the exact position (in pixels) of the animation on each screen. *FishPlayer* offers a separate playlist for both *FishSim* nodes for both screens (Figure 1e). Every entry of the playlist defines the location of a file folder, which includes a swimming path file of either none, 1, or several model stimuli and the used model stimuli themselves. All stimuli are automatically loaded to the relevant *FishSim*. If the folder does not contain any stimulus model, the virtual tank is empty (structure of a tank background but no stimulus).

The swimming path file and the model stimuli inside the folder are not linked to each other. As a consequence, the models inside the folder can be replaced by any other models without changing the animated movement, thus animations are exactly the same for different models. This allows highly standardized conditions in different trials within an experiment, in which the behavior of stimuli is consistent. All different stimuli of a trial can move and behave in the same way defined by a single swimming path file. For example, Gierszewski et al. (2017) combined a colored 3D box with the same swimming path as the other stimulus, a 3D sailfin molly male. The box either swam through the virtual tank or did not move. The box was then presented together with a 3D fish on the second screen, either swimming the same pathway as the box or not swimming, to test the relevance of stimulus shape and movement in mate-choice experiments. Gierszewski et al. (2017) showed that a moving stimulus (both box and fish) elicited a stronger response in live sailfin molly females than an immobile stimulus, but a swimming fish was even better than a swimming box.

For the experiment, the user defines 2 playlists for the whole trial. *FishPlayer* automatically plays entry after entry, changes the stimuli in each instance of *FishSim*, and sends the movement commands of the swimming path files stored in the entry folder. For each playlist the chronological order and number of entries can be chosen individually by the user. This simplifies and automatizes the process of using animated visual stimuli in experiments and reduces the workload.

Discussion

We developed and introduced a novel toolchain offering a user-friendly way to design virtual fish stimuli by behavioral biologists.

Stimuli can easily be animated (by using a game controller) and can then be presented on screen during experiments in a highly standardized manner. We especially wanted to highlight the special features of our novel toolchain, which differ from previous methods for stimulus design, animation, and presentation. First, *FishCreator* provides a user-friendly option to design stimuli with high degree of variation in morphology and coloration. *FishCreator* offers to a folder containing various fish textures providing various combinations of different cues to prevent pseudoreplication in experiments as stimuli can be designed randomly and size measurements can be adjusted to represent mean values of populations, as proposed by Rosenthal (2000). Second, with *FishSteering* we introduced a novel method to animate stimuli with the help of a game controller. This gives researchers the possibility to animate stimuli without practice and knowledge on how to technically develop animations in an easy and fast way. Third, the produced swimming path files can be applied to any model, even to arbitrary models (e.g., box; Gierszewski et al. 2017), and can be rendered in real-time onto different screens, which is in contrast to Ingleby et al. (2015). Furthermore, *FishPlayer* highly simplifies the way of conducting dual-screen-based stimuli experiments. Finally, our toolchain is based on a robot middleware, which makes it possible to extend the system easily by other components like tracking systems. These features make the toolchain very flexible and applicable to a huge variety of experiments in animal behavior research like experiments in mate choice, shoaling decisions, cognitive abilities like perception and discrimination abilities, communication and even those regarding the evolution of sexual traits. With our novel toolchain it is possible to conduct experiments with stimuli showing a high variability in morphology and behavior in a highly standardized and controlled manner, which is impossible with live test animals. Additionally, it helps to reduce the number of experimental animals concerning animal welfare as defined in the 3Rs by Richmond (2010).

Gierszewski et al. (2017) used our presented method in several experiments. During a thorough validation with live sailfin mollies, the acceptance and viability of the created virtual 3D fish models was confirmed. It could be shown that live test females preferred a virtual 3D male over an empty virtual 3D tank when presented in a

2-choice situation. Moreover, live females spent the same amount of time in front of the virtual 3D male, in front of a live male, or in front of a video of a live male when presented with an empty tank as alternative stimulus, hence showing that live test fish preferred to spent time with the presented fish stimulus irrespective of the method used. This result underlines that our animation is equally as attractive as a live male and a video of a male. During experiments using a virtual 3D male and a virtual 3D box that were either presented moving or static, it could be shown that live test females spent more time in front of a moving stimulus even if it was the artificial box. This result confirmed the usability and acceptance of our new approach to animate a stimulus with the help of a game controller. Additionally, tests revealed that the “fish” shape of a virtual stimulus was preferred over an animated “swimming” box, indicating that live females were able to distinguish between virtual stimuli. Moreover, live male test fish could discriminate between a virtual 3D male and a virtual 3D female. These results showed that our approach of generating a 3D virtual stimulus enables the design of realistic virtual sailfin molly models that are recognized by live test fish and also provide sufficient information about sex-specific characteristics. Hence, the validation performed by Gierszewski et al. (2017) indicated the use of our virtual fish simulation to be a powerful tool for mate-choice studies in sailfin mollies. Moreover, our toolchain is applicable in studies with other fish species as well.

We have now extended the system with a 3D tracking system (Müller et al. 2014, 2016) to overcome one of the major limitations of virtual stimuli: the lack of interaction between the live test fish and its virtual counterpart. This will open up new horizons for future studies using computer animation in animal behavior research.

Acknowledgment

We thank Kathryn Dorhout for proofreading the manuscript.

Funding

The presented work was developed within the scope of the interdisciplinary, DFG (Deutsche Forschungsgemeinschaft)-funded project “virtual fish” (KU 689/11-1 and Wi 1531/12-1) of the Institute of Real-Time Learning Systems (EZLS) and the Research Group of Ecology and Behavioral Biology at the University of Siegen.

References

- Allen JM, Nicoletto PF, 1997. Response of *Betta splendens* to computer animations of males with fins of different length. *Copeia* 1:195–199.
- Bakker TCM, Künzler R, Mazzi D, 1999. Condition-related mate choice in sticklebacks. *Nature* 401:234–234.
- Baldauf SA, Kullmann H, Thünken T, Winter S, Bakker TCM, 2009. Computer animation as a tool to study preferences in the cichlid *Pelvicachromis taeniatus*. *J Fish Biol* 75:738–746.
- Baube C, Rowland W, Fowler J, 1995. The mechanisms of colour-based mate choice in female threespine sticklebacks: hue, contrast and configurational cues. *Behaviour* 132:979–996.
- Butail S, Bartolini T, Porfiri M, 2013. Collective response of zebrafish shoals to a free-swimming robotic fish. *PLoS ONE* 8:e76123.
- Butkowski T, Yan W, Gray AM, Cui R, Verzijden MN et al., 2011. Automated interactive video playback for studies of animal communication. *J Vis Exp* 48:e2374.
- Chouinard-Thuly L, Gierszewski S, Rosenthal GG, Reader SM, Rieucan G et al., 2017. Technical and conceptual considerations for using animated stimuli in studies of animal behavior. *Curr Zool* 63:5–19.
- Culumber ZW, Rosenthal GG, 2013. Mating preferences do not maintain the tailspot polymorphism in the platyfish *Xiphophorus variatus*. *Behav Ecol* 24:1286–1291.
- Faria JJ, Dyer JR, Clément RO, Couzin ID, Holt N et al., 2010. A novel method for investigating the collective behaviour of fish: introducing ‘Robofish’. *Behav Ecol Sociobiol* 64:1211–1218.
- Fischer S, Taborsky B, Burlaud R, Fernandez AA, Hess S et al., 2014. Animated images as a tool to study visual communication: a case study in a cooperatively breeding cichlid. *Behaviour* 151:1921–1942.
- Fleishman LJ, Endler JA, 2000. Some comments on visual perception and the use of video playback in animal behavior studies. *Acta Ethol* 3:15–27.
- Gierszewski S, Müller K, Smielik I, Hütwohl J-M, Kuhnert K-D et al., 2017. The virtual lover: variable and easily guided 3D fish animations as an innovative tool in mate-choice experiments with sailfin mollies-II. Validation. *Curr Zool* 63:65–74.
- Ingleby SJ, Asl MR, Wu C, Cui R, Gadelhak M et al., 2015. anyFish 2.0: an open-source software platform to generate and share animated fish models to study behavior. *SoftwareX* 3–4:13–21.
- Künzler R, Bakker TC, 1998. Computer animations as a tool in the study of mating preferences. *Behaviour* 135:1137–1159.
- Künzler R, Bakker TC, 2001. Female preferences for single and combined traits in computer animated stickleback males. *Behav Ecol* 12:681–685.
- Landgraf T, Bierbach D, Nguyen H, Muggelberg N, Romanczuk P et al., 2016. RoboFish: increased acceptance of interactive robotic fish with realistic eyes and natural motion patterns by live Trinidadian guppies. *Bioinspir Biomim* 11:015001.
- Landgraf T, Nguyen H, Schröder J, Szengel A, Clément RJG et al., 2014. Blending in with the Shoal: robotic fish swarms for investigating strategies of group formation in guppies. In: Duff A, Lepora NF, Mura A, Prescott TJ, Verschure PFMJ, editors. *Biomimetic and Biobrybrid Systems*. Switzerland: Springer International Publishing, 178–189.
- Mazzi D, Künzler R, Bakker TC, 2003. Female preference for symmetry in computer-animated three-spined sticklebacks *Gasterosteus aculeatus*. *Behav Ecol Sociobiol* 54:156–161.
- Mazzi D, Künzler R, Largiadèr CR, Bakker TC, 2004. Inbreeding affects female preference for symmetry in computer-animated sticklebacks. *Behav Genet* 34:417–424.
- McDonald C, Reimchen T, Hawryshyn C, 1995. Nuptial colour loss and signal masking in *Gasterosteus*: an analysis using video imaging. *Behaviour* 132:963–977.
- McKinnon JS, 1995. Video mate preferences of female three-spined sticklebacks from populations with divergent male coloration. *Anim Behav* 50:1645–1655.
- McKinnon JS, McPhail J, 1996. Male aggression and colour in divergent populations of the threespine stickleback: experiments with animations. *Can J Zool* 74:1727–1733.
- Mehlis M, Bakker TC, Frommen JG, 2008. Smells like sib spirit: kin recognition in three-spined sticklebacks *Gasterosteus aculeatus* is mediated by olfactory cues. *Anim Cogn* 11:643–650.
- Morris MR, Moretz JA, Farley K, Nicoletto P, 2005. The role of sexual selection in the loss of sexually selected traits in the swordtail fish *Xiphophorus continens*. *Anim Behav* 69:1415–1424.
- Morris MR, Nicoletto PF, Hesselman E, 2003. A polymorphism in female preference for a polymorphic male trait in the swordtail fish *Xiphophorus cortezi*. *Anim Behav* 65:45–52.
- Müller K, Gierszewski S, Witte K, Kuhnert KD. 2016 (Forthcoming). Where is my mate? - Real-time 3-D fish tracking for interactive mate-choice experiments. *ICPR 2016—23rd International Conference on Pattern Recognition; VAIB 2016—Visual observation and analysis of Vertebrate and Insect Behavior Workshop Proceedings*. Cancun, Mexico, 4 December, 2016.
- Müller K, Schlemper J, Kuhnert L, Kuhnert KD, 2014. Calibration and 3D ground truth data generation with orthogonal camera-setup and refraction compensation for aquaria in real-time. In: Battiatto S, Braz J, editor. *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, Volume 3. Lisbon, Portugal, 5-8 January, 2014. SciTePress 2014, 626-634.

- Nakayasu T, Watanabe E, 2014. Biological motion stimuli are attractive to medaka fish. *Anim Cogn* 17:559–575.
- Oliveira RF, Rosenthal GG, Schlupp I, McGregor PK, Cuthill IC et al., 2000. Considerations on the use of video playbacks as visual stimuli: the Lisbon workshop consensus. *Acta Ethol* 3:61–65.
- Pelkwijk JT, Tinbergen N, 1937. Eine reizbiologische Analyse einiger Verhaltensweisen von *Gasterosteus aculeatus* L. *Z Tierpsychol* 1:193–200.
- Richmond J, 2010. The three Rs. In: Hubrecht R, Kirkwood J, editors. *The UFAW Handbook on the Care and Management of Laboratory and Other Research Animals*. Oxford: Wiley-Blackwell, 5–22.
- Rosenthal GG, 2000. Design considerations and techniques for constructing video stimuli. *Acta Ethol* 3:49–54.
- Rosenthal GG, Evans CS, Miller WL, 1996. Female preference for dynamic traits in the green swordtail *Xiphophorus helleri*. *Anim Behav* 51:811–820.
- Rowland WJ, Bolyard KJ, Jenkins JJ, Fowler J, 1995. Video playback experiments on stickleback mate choice: female motivation and attentiveness to male colour cues. *Anim Behav* 49:1559–1567.
- Smielik I, Müller K, Kuhnert KD, 2015. Fish motion simulation. In: Al-Akaidi M, Ayesch A, editors. *ESM-European Simulation and Modelling Conference 2015*. Leicester, United Kingdom, 26–28 October, 2015. EUROSIS, 392–396.
- Thünken T, Baldauf SA, Kullmann H, Schulz J, Hesse S et al., 2011. Size-related inbreeding preference and competitiveness in male *Pelvicachromis taeniatus* (Cichlidae). *Behav Ecol* 22:58–362.
- Veen T, Ingleby SJ, Cui R, Simpson J, Asl MR et al., 2013. *anyFish*: an open-source software to generate animated fish models for behavioural studies. *Evol Ecol Res* 15:361–375.
- Wong BBM, Rosenthal GG, 2006. Female disdain for swords in a swordtail fish. *Am Nat* 167:136–140.
- Zeil J, 2000. Depth cues, behavioural context, and natural illumination: some potential limitations of video playback techniques. *Acta Ethol* 3:39–48.