Acta Zoologica Academiae Scientiarum Hungaricae 63(1), pp. 137–142, 2017
DOI: 10.17109/AZH.63.1.137.2017

SIMPLE COMMAND-LINE OPEN-SOURCE SOFTWARE TO ANALYSE BEHAVIOURAL OBSERVATION VIDEO RECORDINGS

Miklós Bán*, Mihály Földvári, Melinda Babits and Zoltán Barta

Department of Evolutionary Zoology and Human Biology and
MTA-DE "Lendület" Behavioural Ecology Research Group, University of Debrecen
*H-4020 Debrecen, Egyetem tér 1, Hungary; *E-mail: banm@vocs.unideb.hu*

Video techniques are broadly used in the field of behaviour sciences, thus there is an increasing interest in softwares that simplify the processing of recordings. One of the difficulties is that observed objects can exhibit a wide variety of behaviours that cannot usually be recognized and recorded automatically, therefore the video has to be watched and analysed manually by humans.

We have developed a simple, open-source and free command-line video analysis software, that helps the processing of observation sessions from recordings and helps to verify the processed events.

Keywords: video event recording, behaviour, experiment, data analysis.

INTRODUCTION

Scientific research relies on observation and the analysis of the data acquired through these observations. To produce exact and repeatable results scientists often use video techniques for recording the events of an experiment. Annotations are needed to back up claims based on the events recorded from the videos. The main goal of event recorder softwares is to enable the transcription of events directly with a keyboard in order to make storing and analysing video data easier. Although video recording techniques are highly developed, the analysis of videos is not necessarily simple. The available tools improved greatly in the last decades (Godden & Graham 1983, Kabra *et al.* 2013) thanks to the rapid development of computational techniques. In contrast to the simplest error-proof and slowest frame-to-frame video analysis methods (Barta *et al.* 2004, Mónus & Barta 2008) real-time and automated tools have recently been developed (Crispim *et al.* 2012, Kabra *et al.* 2013, Madan & Spetch 2014).

Despite these recent technical developments video analyses can still generate difficulties to scientists, especially in the field of animal behaviour. The wide variety of behavioural actions animals can perform and the non-standard environments these videos are recorded in prevents the development of a general solution to a simple and automatic video recording analysis. In such cases human input is necessary to recognize and categorize the events due to the high complexity or the low quality of the video records. The level of necessary human effort, however, depends heavily on the features of the applied analysis tool. Although there are numerous different computer applications to handle video recordings with dozens of functionalities, the crucial requirement of the accurate analysis is the ability to measure time: to quantify the elapsed time of particular events. Precise time measurement is possible if the frame rate of the video and the frame number of the event are known. In case of real time video analyses a software tool is needed to read the video time at the moment of the recorded event. These kinds of video analyses are very common in scientific research, when the observer watches a video recording and wants to extract information related to particular events (Godden & Graham 1983, Lind *et al.* 2005, Noldus *et al.* 2002).

Some of the scientific video event analyser softwares use their own embedded video player (e.g. Observer: Noldus *et al.* 2000) while others use an external video player, which is integrated in their graphical interface (e.g. BORIS: Friard & Gamba 2016, S&W and Vcode: Hagedorn *et al.* 2008). Softwares using external video player are more flexible regarding to video file formats and playing parameters, which can be advantageous in cases when there is a wide variety of input file types to be analysed.

All scientific video event recorder softwares that we know have graphical user interface. Although, graphical user interfaces are professedly comfortable thanks to the many functions that are available on the screen, these buttons and input areas occupy valuable area on the screen. By using a command-line application the reduction of screen area useful for watching the recordings can be avoided. Watching videos in full screen is especially useful when one tries to analyse high-resolution recordings showing small details: screen size is essential to improve the quality of the data. During this process users need a tool that allows focusing on the events of the video recording without paying much attention to the helper application.

Considering these needs we have developed a simple command-line driven scientific event recorder software (MWrap), which uses an external video player application (MPlayer) for displaying video files.

GENERAL OVERVIEW

We have focused on the simple event recording instead of providing dozens of easily accessible features. As a result, using our command line interface is not complicated; many of the functionalities found in other softwares are available in MWrap by using key-combinations or through initial settings in configuration files. MWrap is a simple command-line tool, which provides the user interface for event recording only, and delegates the handling of the video file to the external viewer.

MWrap offers different levels of sophistication for analysing video recordings, so the simplest recording technique can be used that is suitable for the particular task in hand. Recording of momentary events (behavioural patterns with short duration, for example biting or touching) and ranged events or states (activity with relatively long duration, such as grooming) are both possible. Behavioural sampling – in which case all behavioural events displayed by each individual on the video are recorded – or scan sampling (Martin & Bateson 2007) – when the behaviour of one or more individuals is documented at given intervals – are both possible by using MWrap.

IMPLEMENTATION

Using MWrap

Since MWrap uses an external video player and it has no own graphical interface, the event recorder and the video stream appear in two separate windows (Digital supplement 1). Both windows can be set to use the entire screen, and can be moved to another monitor.

Every hit on the keyboard initiates a bidirectional communication between the event recorder and the video player and is written into an output file. Read times are seconds elapsed from the first frame of the video record, which is independent from frame rate, seeking positions or the playing speed. These time values can easily be transformed into real times either in case of normal or time-lapse (concatenated consecutive images) videos.

The video player can be controlled by using initial parameters like video resolution or sound output and it is possible to use the program's own control keys to manage the video player itself. MWrap reads initial parameter settings from a configuration file. This is a simple text file and it is created (and placed in the working directory) automatically with default values when it is missing in the working directory. This parameter configuration solution allows setting up and using different settings for every video file if it is necessary. During event recording, the video player can be controlled (e.g. changing play speed) by the video player's own control keys. Some of its functionalities (like stop, pause-play) are available by using MWrap's special keys.

MWrap differentiates between event and control (special) keys. Any of the letter and number keys can be chosen as an event key, some other keys are assigned to specific control functions (e.g. jump to time position, delete last event, mark specimen). During the analyses usage of event keys that are not pre-assigned to any events and thus to any labels is also permitted (Digital supplement 2). In such cases an event is recorded without a label (Digital supplement 2).

As output of an event recording session MWrap creates simple text files. These can easily be imported in a wide variety of data analysing softwares. All separate sessions have a separate output file. All videos can have multiple event recording sessions. Each recorded event forms a row in the output file containing fields separated by user-defined characters. The fields are: automatically created event ID, key name, event label, and video time in seconds, specimen identifier and event duration.

To verify the recorded events MWrap provides an additional application (mcode.pl), which can create subtitles for the analysed video recording and retrieves the frames of every marked event as an image file thus creating a picture collection of the events. The application mcode.pl automatically writes the event labels on the images. Thus, after the event recording phase the researcher either can re-watch the video itself with labels of the actual behavioural events as subtitles, or can have a look at only those frames that contain recorded events and their labels. These possibilities ease the verification of the data recording process.

MWrap application was written in Perl (Script language functional under various operating systems, Christiansen *et al.* 2012) and only contains few dependencies, however, one important component for the synchronous bidirectional communication (handling named pipes) is currently only available on Linux and other UNIX compatible operating systems. For users without access to Linux we have created a live Linux system with MWrap. This bootable image file is available from the project's portal.

Special features

Calculating elapsed times between events: During event recording it is possible to mark any event as "terminal" event and if there is a previous event with the same label the elapsed time between the two events will be appended to the event row in the output file.

Specimen marking: Many times it is feasible that researchers are able to group events, like when recording the behaviour of several individuals from the same video footage. MWrap allows this functionality by enabling the attachment of a specimen identifier to an event during an event recording session. The attached identifier is appended to event rows in the output file.

Restart event recording: Any interrupted event recording can be continued later from the beginning or the last recorded event's position.

Delete last event: During an event recording the last event can be deleted, and these events are marked in the output file.

Seeking to time positions: By using the control keys different types of seekings (like jump forward or backward by predefined time-steps or seek back to the first position of specified marks) are available.

USAGE EXAMPLES

In our research group MWrap was used to analyse the behaviour of several species belonging to different taxa (firebugs, flies, spiders and birds) in various research projects. For example we used MWrap to analyse video recordings of open-field arena tests. As measurements of exploratory behaviour we recorded the time when an individual fire bug *(Pyrrhocoris apterus)* reached the wall of the arena, and the timing and frequencies of the individuals touching novel-objects (Gyuris *et al.* 2011). In another project we tested larval behaviour of stalk-eyed flies *(Teleopsis dalmanni)* by measuring behavioural traits such as boldness, exploration and activity (unpubl. data).

AVAILABILITY AND REQUIREMENTS

Project name: MWrap

Project home page: http://zoology.unideb.hu/mwrap/

Source code repository: https://github.com/EvoZooDeb/mwrap

Operating system(s): Unix compatible OS, tested on Debian and Ubuntu LINUX. MWrap Live can run on any kind of PC with an USB port and MWrap 2 (development version) can run on Microsoft Windows.

Programming language: Perl (Christiansen *et al.* 2012, https://www.perl.org/)

PERL requirements (all available from CPAN.org): IO::Handle, Term::ReadKey, File::Basename, File::Find, File::Copy, POSIX

Optional Perl libraries (all available from CPAN.org): Image::Magick, Statistics::R

License: GNU GPLv3

Software versions: MWrap: 2016.06.22
Perl: 5.20.2
MPlayer: SVN-r37375

*

Authors' contributions – M. Bán developed MWrap and mcode.pl applications. Z. Barta developed processor scripts for data analyses. M. Földvári and M. Babits tested the application and prepared the user manual. M. Bán and M. Földvári drafted the manuscript and Z. Barta helped in manuscript revision. All authors read and approved the final manuscript.

REFERENCES

Barta, Z., Liker, A. & Mónus, F. (2004): The effects of predation risk on the use of social foraging tactics. *Animal Behaviour* **67**: 301–308. https://doi.org/10.1016/j.anbehav.2003.06.012

Crispim Junior, C. F., Pederiva, C. N., Bose, R. C., Garcia, V. A., Lino-de-Oliveira, C. & Marino-Neto, J. (2012): ETHOWATCHER: validation of a tool for behavioral and video-tracking analysis in laboratory animals. *Computers in Biology and Medicine* **42**: 257–264. https://doi.org/10.1016/j.compbiomed.2011.12.002

Christiansen, T., Foy, B. D., Wall, L. & Orwant J. (2012): *Programming Perl*. 4th ed. O'Reilly Media, Sebastopol, USA, 1184 pp.

Friard, O. & Gamba, M. (2016): BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations. *Methods in Ecology Evolution* **7**(11): 1325–1330. https://doi.org/10.1111/2041-210X.12584

Godden, D. H. & Graham, D. (1983): 'Instant' analysis of movement. *Journal of Experimental Biology* **107**: 505–508.

Gyuris, E., Feró, O., Tartally, A. & Barta, Z. (2011): Individual behaviour in firebugs (Pyrrhocoris apterus). *Proceedings of the Royal Society of London B: Biological Sciences* **278**: 628–633. https://doi.org/10.1098/rspb.2010.1326

Hagedorn, J., Hailpern, J. & Karahalios, K. G. (2008): VCode and VData: illustrating a new framework for supporting the video annotation workflow. Pp. 317–321. *In:* Bottoni, P. & Levialdi, S. (eds): *Proceedings of the Working Conference on Advanced Visual Interfaces.* Association for Computing Machinery Inc., New York. https://doi.org/10.1145/1385569.1385622

Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S. & Branson, K. (2013): JAABA: interactive machine learning for automatic annotation of animal behavior. *Nature Methods* **10**: 64–67. https://doi.org/10.1038/nmeth.2281

Lind, N. M., Vinther, M., Hemmingsen, R. P. & Hansen, A. K. (2005): Validation of a digital video tracking system for recording pig locomotor behaviour. *Journal of Neuroscience Methods* **143**(2): 123–132. https://doi.org/10.1016/j.jneumeth.2004.09.019

Madan, C. R. & Spetch, M. L. (2014): Visualizing and quantifying movement from prerecorded videos: The spectral time-lapse (STL) algorithm. *F1000Research* **3**: 19. https://doi.org/10.12688/f1000research.3-19.v1

Martin, P. & Bateson, P. (2007): *Measuring behaviour: An introductory guide.* 3rd ed. Cambridge University Press, Cambridge, 223 pp. https://doi.org/10.1017/CBO9780511810893

Mónus, F. & Barta, Z. (2008): The effect of within-flock spatial position on the use of social foraging tactics in free-living tree sparrows. *Ethology* **114**: 215–222. https://doi.org/10.1111/j.1439-0310.2007.01472.x

Noldus, L. P. J. J., Trienes, R. J. H., Hendriksen, A. H. M., Jansen, H. & Jansen, R. G. (2000): The Observer Video-Pro: New software for the collection, management, and presentation of time-structured data from videotapes and digital media files. *Behavior Research Methods, Instruments, & Computers* **32**: 197–206. https://doi.org/10.3758/BF03200802

Noldus, L. P. J. J., Spink, A. J., & Tegelenbosch, R. A. J. (2002): Computerised video tracking, movement analysis and behaviour recognition in insects. *Computers and Electronics in Agriculture* **35**(2): 201–227. https://doi.org/10.1016/S0168-1699(02)00019-4

Received February 1, 2016, accepted October 10, 2016, published March 3, 2017