The Simple Video Coder: A free tool for efficiently coding social video data

Daniel Barto¹ · Clark W. Bird¹ · Derek A. Hamilton¹ · Brandi C. Fink²

Published online: 8 August 2016
© Psychonomic Society, Inc. 2016

Abstract Videotaping of experimental sessions is a common practice across many disciplines of psychology, ranging from clinical therapy, to developmental science, to animal research. Audio–visual data are a rich source of information that can be easily recorded; however, analysis of the recordings presents a major obstacle to project completion. Coding behavior is time-consuming and often requires ad-hoc training of a student coder. In addition, existing software is either prohibitively expensive or cumbersome, which leaves researchers with inadequate tools to quickly process video data. We offer the Simple Video Coder—free, open-source software for behavior coding that is flexible in accommodating different experimental designs, is intuitive for students to use, and produces outcome measures of event timing, frequency, and duration. Finally, the software also offers extraction tools to splice video into coded segments suitable for training future human coders or for use as input for pattern classification algorithms.

Keywords Behavior coding · Open-source software · Video analysis

For psychologists, the analysis of video data allows the researcher to observe a participant's behavior as it occurred. Video is able to capture both the individual's moment-to-moment behavior and the immediate context in which the

behavior occurred. This enables researchers to quantify complex behaviors such as speech, facial gestures, and social interactions. Much work has been done on establishing ecologically valid methods for coding these complex behaviors from video records. For example, in clinical science, different laboratories have identified what constitutes “change talk” or how to evaluate therapeutic sessions (Miller, Yahne, Moyers, Martinez, & Pirritano, 2004). Developmental psychologists have investigated dyadic engagement with regard to personality development (Farroni, Csibra, Simion, & Johnson, 2002), and the development of the motor system through exploration of the environment (Adolph & Berger, 2006). Other psychologists have coded discussions of disagreements for the prediction of relationship dissolution and divorce (Gottman & Levenson, 2000). Animal researchers have utilized video analysis to examine the development of behavior sequences in novel environments, (Eilam & Golani, 1988), orofacial measures as a result of genetic manipulation (Pinel, Jones, & Whishaw, 1992), and how drug administration affects social behavior (Hamilton et al., 2014). All of these fields of study use video analysis to investigate a variety of complex behaviors.

Despite the diversity of these research fields, the common problem exists of quickly processing video data. The coding of a behavior video is a time-consuming process involving repeated watching of a video, with frequent pausing and rewinding. Reliability between coders is often a concern, which necessitates much time spent in training and coaching new researchers before behavioral coding can even begin (Hallgren, 2012). All of these issues are inherent to the field of behavioral coding and are unlikely to be fixed by new software. Many improvements can be made, however, in allowing a software program to more efficiently facilitate video coding, including establishing a database for future coding training.

✉ Daniel Barto
dbarto@unm.edu

¹ Department of Psychology, University of New Mexico, MSC03 2220, 1, Albuquerque, NM 87131, USA

² Department of Psychiatry and Behavioral Sciences, Clinical and Translational Science Center, University of New Mexico, Albuquerque, NM, USA

The Simple Video Coder (SVC) software that we propose was created to increase the efficiency of researchers in processing video data. Although other coding software exists that offers a multitude of options, the usefulness of these solutions is limited by expense and complexity. For example, the Noldus Observer has a rich set of data visualization tools but requires an expensive license (Noldus, Trienes, Hendriksen, Jansen, & Jansen, 2000). SVC is not the only open-source option, since other freeware programs also exist for behavior coding. Datavyu, a free open-source software application, contains a powerful scripting utility to enable researchers to automate data analysis and extend functionality (Datavyu Team, 2014), but this complexity comes with a steeper learning curve, because the interface has more options. Cowlog (Hänninen & Pastell, 2009) and the Observational Data Coding System (Maclin & Maclin, 2005) are both open-source coding programs; however, the coding output requires an additional parser or experience with the R programming language to obtain the data. The goal of SVC, from installation to use to output, is to create a simplified experience for users to code videos and generate immediately useful output.

SVC is intended to provide an efficient way to code videos in a straightforward, flexible, and approachable manner for the user. With SVC the processes of software installation, behavior coding, and data analysis have been streamlined for the user to begin coding immediately without requiring external knowledge sources to operate the software. The components of SVC, Python (the scripting language) and MPlayer (the video player), were chosen to offer the user a stable environment compatible across all versions of Windows, while also allowing for a wide range of video file types to be played. This allows the coder to focus more on applying the coding scheme than learning how to operate and troubleshoot the software or manage video file incompatibility. In addition, the SVC interface is designed to facilitate the training of new coders while providing the tools necessary for expert coders to quickly and effectively code videos.

The software we describe here differs from its predecessors in three ways: It (1) offers adjustable coding and key-binding assignments, (2) provides an intuitive user interface that is designed to be used by student coders, and (3) provides methods for video segmentation based on codes used for training humans or for use in pattern classification algorithms.

System technical overview

The SVC software is written in the Python language. It comprises four libraries: wxPython, a graphical user interface (www.wxpython.org/); MPlayerCtrl, a Python wrapper for playing movies (<https://pythonhosted.org/MplayerCtrl/>); and the Matplotlib, and Numpy libraries for plotting functions (Hunter, 2007; Van der Walt, Colbert, & Varoquaux, 2011).

The SVC software includes two standalone open-source programs: MPlayer, which is used to control video playback, and Mencoder, which is used to segment the video into coded segments (www.mplayerhq.hu/).

The SVC software supports a large variety of standard video file formats through MPlayer. Therefore, users typically do not need to convert video files for use with SVC. There is no limit to the length of files; however, large files (>1 gigabyte) may affect performance on slower computers. Video files are played at the native frame rate of the file. The SVC software can run only on a Windows-based operating system, ranging from Windows XP to Windows 10. The minimum hardware requirements for the SVC include: 1 GB of hard drive space, 512 MB RAM, and a 1-GHz processor. SVC utilizes the time code of the video file for logging events and is precise to a 0.2-s scale. The time codes are translated into cumulative seconds for all data analysis.

Behavioral coding design

Many different labs use coding schemes that are not necessarily translatable between labs, or even between different projects (Coan & Gottman, 2007). Critically, researchers need to have the ability to change codes, but they also must retain code consistency within a project, without the need to reenter the same codes for each separate video. SVC enables the researcher to create up to 18 unique codes. These codes are stored in a text file that is read into the program once it starts. The researcher may change the codes by changing the entry in the configuration text file. There are no restrictions regarding special characters or spaces; any code stored in the text file is generated in the program and outputted into the data analysis. Codes also do not need to be represented with numbers or single letters, which eliminates confusion while identifying which groups of codes occur within a video. Within the configuration file, the researcher is also able to change the “hotkey” or the keyboard key that is used to log behaviors during a program. Thus, the researcher is able to decide which keys to use for each code, such as using the “S” key to be used to represent the code “Smiling.” The configuration file containing the codes relevant to the project can be easily distributed between coders, and ensures all coders are using the same set of code definitions.

User interface

The SVC interface is designed specifically to be intuitive to researchers who do not have experience with complicated coding software. The interface is designed to facilitate attention to the video being coded itself, rather than the coding software. Behaviors are coded by setting an onset and offset,

and can be logged in two equivalent ways: by using the mouse pointer to check a box located at the bottom of the display, or by using a hotkey on the keyboard. Codes can be either state (a single point in time) or event-based (e.g., a duration; Martin & Bateson, 2007). Event-based codes are states that overlap; that is, one state can occur within the time span of another distinct state. Each code requires the user to define an onset and an offset, and the program will notify the user if there is an error in logging the code. The user is also able to annotate notes to individual codes to provide a qualitative comment on any behavior.

Precise control of playback is important for coding video data. Typically researchers identify a coded behavior only after its onset, and then must rewind the video. The SVC software provides a traditional slider to mark the progress of the video, for more precise rewinding. The ability to precisely control playback allows the user to capture quick events such as facial gestures. The SVC software also has buttons to control variable rewind. Using the variable-rewind buttons, the user is able to jump back either 2, 10, or 20 s, or to step forward one frame at a time. In addition to these playback controls, SVC offers variable playback, to slow down or speed up playback from -50% to $+200\%$ of the frame rate of the video.

The SVC software was also designed to facilitate the training of novice coders. The emphasis during the development of the program was on accessibility and instructive value for novice coders to learn the coding schemes of specific projects. As such, instructive notes can be appended to the coded

behavior to annotate a certain point or movement while the video is being reviewed. We believe that this feature streamlines teaching, for learners are able to view the behavior occurring in context, and not just as an isolated video clip. Figure 1 depicts the user interface for SVC and the Specific Affect Coding System (SPAFF; Coan & Gottman, 2007).

The main display of the interface is the loaded video display. Alongside this display is a panel with the possible codes and information that indicates when a behavior code is on or off. Displaying the possible codes allows the coder to decide which code to use without referring to the options listed in the coding manual. The far right panel logs the code onsets and offsets. Codes can be logged through a keypress or a mouse click, as previously described. Additionally, the user can edit or modify any code or occurrence time by right-clicking on an entry in the coding window. Under the display interface is the control panel, which provides the playback controls for the user. These buttons are similar in style to those of other video players and provide a familiar environment to the user. Similar to operating the behavior codes, these buttons can be controlled through either mouse clicks or keyboard shortcuts. The top menu bar contains the file operations to load videos, save/load data files, and splice video segments. Hovering the mouse over any of these file options or the playback control buttons generates a short description of the option's function in the bottom menu of the display. The visual placement of these elements on the screen is arranged to be similar to placement in other multimedia programs that most end users will have experienced. The goal of such familiarity is to provide

Fig. 1 The SVC user interface, depicting a training video of a research assistant. (A) The video display. (B) The coding options for the session; this panel also provides visual feedback when a code is on. (C) The codes

logged, with their onsets and offsets in seconds. (D) Controls for precise playback, and options for setting the code onsets with a mouse

the user with an intuitive sense of how to operate the software, without requiring extensive training or referring to software documentation (Galitz, 2007).

After the initial coding session is complete, the coded behavior must be evaluated. The SVC software allows for loading in coded datasets and quickly jumping to examples of a coded behavior within the video. By clicking on the coded entry, video playback will begin just 1 s before the specific coded event, to allow users to quickly move through specific codes and verify the accuracy of the behavior codes. For determining coder reliability, the coded data sets may also be stripped of their labels to allow for second-pass coding. As we describe in greater detail below, the comma-separated files that are generated by the program may then be exported to statistical programs for reliability analysis.

Data generation

The SVC software produces human-readable output data that allows for easy export to statistical programs and for rapid statistical analysis. During a coding session, SVC continually autosaves the codes in a separate data file as backup. Once the researcher completes the coding session, two different output files are generated, as depicted in Fig. 2. These files are by default named according to the participant, but they can be named on the basis of the researcher's preference. SVC includes overwrite protection, prompting the user when a file with the same name already exists.

The first file is a reference file that contains the codes generated by the researcher. This file contains the file name and file location of the specific video used, the date the video was coded, the participant identification number, and all the codes in sequential order. The second file outputs demographic data of the codes observed, including frequency, total duration, and time to first onset. The data output from SVC was designed to allow researchers immediate access to frequency and duration information for the coded behaviors, without the need for additional operations such as copying and pasting or performing external calculations in different software. The timing output of SVC, meanwhile, allows the researcher to access the timing onsets and offsets in a columnar format. Both of these files output immediately useful data to the researcher without the need of file conversions or proprietary software. For more complex statistical analysis, these .csv files can be opened in a variety of statistical programs, including Microsoft Excel, SPSS, R, and MATLAB.

Splicing video

SVC is not only an intuitive interface for coding videos, but is also able to translate the original video into separate video

Fig. 2 Overview of SVC data processing. A researcher picks the coding scheme and hotkeys to be used for the project. SVC integrates the codes, and the user loads the video. Once the user codes the video, two output comma-separated value (.csv) files are generated. The first data file (.csv) contains the onsets of the codes and can be used to segment the original video, or can be reloaded into SVC to return to a previous coding session. The second output file contains the frequencies, total duration, and time to first onset of each behavior code

segments labeled by behavior. Coding behavior is an arduous process that reduces the content of the video into dependent measures of frequency and duration. Although this data reduction simplifies analysis, it is important to preserve this coded content for use in more complex data analysis and experimental methodologies. For example, producing coded video segments could allow researchers to use specific clips in an experiment, to train future coders in what constitutes a specific behavior, or to use clips as inputs for pattern classification algorithms.

Pattern classification is a rapidly growing field within machine learning that involves analyzing data sets to find reoccurring patterns and regularities (Kotsiantis, 2007). Pattern classification has been used in many different domains, including facial recognition and vocal inflection (Girard & Cohn, 2015). General pattern classification, in its supervised-learning form, requires the creation of a “training set” in which the algorithm is provided material that is known to contain the pattern. Although there have been impressive

advances in detecting patterns, training sets critically still need to be rated by humans to verify and inform the performance of classifiers (Marszalek, Laptev, & Schmid, 2009). Behavioral coding in general, and by SVC specifically, with its emphasis on coding and data generation that is both immediate and accessible, is able to produce rich training sets that would be valuable for pattern classification algorithms that use supervised learning.

SVC does not perform pattern classification directly; rather, it is a tool for translating the coding that researchers are already doing into formats that are applicable for different research methodologies. Once the video is coded, SVC can produce new segments based on the onsets and offsets of the codes. These new segments are labeled by participant and behavior code. The researcher thus is not required to splice video as soon as coding is finished. As long as the text files and original videos are saved, splicing can be done on demand. Thus, over time, SVC could be used as a tool to produce a database of coded behavior.

Conclusion

SVC is novel software that was developed specifically for researchers to code behavioral data. It can accommodate different coding schemes and is designed to be intuitive for users with limited computer experience. By minimizing more complex data visualization and reducing the number of superfluous options, the interface is streamlined to allow for easy use. Finally, the data output of SVC enables researchers to preserve their hours of work coding behavior, and it offers the ability to generate coded video segments on demand. In conclusion, SVC offers the researcher the opportunity to enhance data collection practices already in use by increasing the efficiency of user coding, archiving the coded behavior in relation to the timestamps of the video, and offering the possibility of splicing the original video into coded segments for future use.

Installation

SVC is published under the GNU license and is free to use. It is open-source and free to be modified if the user chooses. The authors claim no ownership over any data generated with the software. If you use the software, we ask that you cite this article.

To download the free software, please visit the corresponding author's website: www.unm.edu/~dbarto/. The installation package includes the SVC executable, compiled with Python 2.7.8, and Python libraries with a manual for use.

Author note This research was supported in part by NIH/NIAAA Grant Number R01 AA019462 to D.A.H., and by NIH/NIAAA Grant

Number R21 AA022367 to B.C.F. B.C.F. is also supported by the National Center for Research Resources and the National Center for Advancing Translational Sciences of the National Institutes of Health, through Grant Numbers UL1 TR001449 and KL2 TR001448.

References

- Adolph, K. E., & Berger, S. E. (2006). Motor development. In D. Kuhn, R. S. Siegler, W. Dam, & R. M. Lerner (Eds.), *Handbook of child psychology* (6th ed., Vol. 2, pp. 161–213). Hoboken, NJ: Wiley. doi:10.1002/9780470147658.chpsy0204
- Coan, J. A., & Gottman, J. M. (2007). The Specific Affect Coding System (SPAFF). In J. A. Coan & J. J. B. Allen (Eds.), *Handbook of emotion elicitation and assessment* (pp. 267–285). Oxford, UK: Oxford University Press.
- Datavyu Team. (2014). Datavyu: A video coding tool. New York, NY: New York University, Databrary Project. Retrieved from datavyu.org
- Eilam, D., & Golani, I. (1988). The ontogeny of exploratory behavior in the house rat (*Rattus rattus*): The mobility gradient. *Developmental Psychobiology*, 21, 679–710. doi:10.1002/dev.420210707
- Farroni, T., Csibra, G., Simion, F., & Johnson, M. H. (2002). Eye contact detection in humans from birth. *Proceedings of the National Academy of Sciences*, 99, 9602–9605. doi:10.1073/pnas.152159999
- Galitz, W. O. (2007). *The essential guide to user interface design: An introduction to GUI design principles and techniques*. Hoboken, NJ: Wiley.
- Girard, J. M., & Cohn, J. F. (2015). Automated audiovisual depression analysis. *Current Opinion in Psychology*, 4, 75–79. doi:10.1016/j.copsyc.2014.12.010
- Gottman, J. M., & Levenson, R. W. (2000). The timing of divorce: Predicting when a couple will divorce over a 14-year period. *Journal of Marriage and Family*, 62, 737–745. doi:10.1111/j.1741-3737.2000.00737.x
- Hallgren, K. A. (2012). Computing inter-rater reliability for observational data: An overview and tutorial. *Tutorials in Quantitative Methods for Psychology*, 8, 23–34.
- Hamilton, D. A., Barto, D., Rodriguez, C. I., Magcalas, C. M., Fink, B. C., Rice, J. P., ... Savage, D. D. (2014). Effects of moderate prenatal ethanol exposure and age on social behavior, spatial response perseveration errors and motor behavior. *Behavioural Brain Research*, 269, 44–54. doi:10.1016/j.bbr.2014.04.029
- Hänninen, L., & Pastell, M. (2009). CowLog: Open-source software for coding behaviors from digital video. *Behavior Research Methods*, 41, 472–476. doi:10.3758/BRM.41.2.472
- Hunter, J. D. (2007). Matplotlib: A 2-D graphics environment. *Computing in Science and Engineering*, 9, 90–95.
- Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, 31, 249–268.
- Maclin, O. H., & Maclin, M. K. (2005). Coding observational data: A software solution. *Behavior Research Methods*, 37, 224–231. doi:10.3758/BF03192690
- Marszalek, M., Laptev, I., & Schmid, C. (2009). Actions in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009* (pp. 2929–2936). Piscataway, NJ: IEEE Press. doi:10.1109/CVPR.2009.5206557
- Martin, P., & Bateson, P. (2007). *Measuring behaviour* (3rd ed.). Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9780511810893
- Miller, W. R., Yahne, C. E., Moyers, T. B., Martinez, J., & Pirritano, M. (2004). A randomized trial of methods to help clinicians learn motivational interviewing. *Journal of Consulting and Clinical Psychology*, 72, 1050–1062. doi:10.1037/0022-006X.72.6.1050

- 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. doi:[10.3758/BF03200802](https://doi.org/10.3758/BF03200802)
- Pinel, J. P., Jones, C. H., & Whishaw, I. Q. (1992). Behavior from the ground up: Rat behavior from the ventral perspective. *Psychobiology*, 20, 185–188.
- Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, 13, 22–30.