Masked-piper: Masking personal identities in visual recordings while preserving multimodal information

Babajide Owoyele¹, James Trujillo²,³, Gerard de Melo¹, & Wim Pouw²,³*

¹ Hasso Plattner Institute, University of Potsdam
² Donders Center for Brain, Cognition, and Behaviour, Radboud University, Nijmegen
³ Max Planck Institute for Psycholinguistics, Nijmegen
*Corresponding author: wim.pouw@donders.ru.nl

Abstract

In this increasingly data-rich world, visual recordings of human behavior are often unable to be shared due to concerns about privacy. Consequently, data sharing in behavioral science, multimodal communication, and human movement research is often limited. In addition, in legal and other non-scientific contexts, privacy-related concerns may preclude the sharing of video recordings and thus remove the rich multimodal context that humans recruit to communicate. Minimizing the risk of identity exposure while preserving critical behavioral information would maximize utility of public resources (e.g., research grants) and time invested in audio-visual research. Here we present an open-source computer vision tool that masks the identities of humans while maintaining rich information about communicative body movements. Furthermore, this masking tool can be easily applied to many videos, leveraging computational tools to augment the reproducibility and accessibility of behavioral research. The tool is designed for researchers and practitioners engaged in kinematic and affective research. Application areas include teaching/education, communication and human movement research, CCTV, and legal contexts.

Keywords

- Multimodal communication
- Kinematic research
- Data Privacy
- Open Science
- Masking
- Research Reproducibility

Metadata

<table>
<thead>
<tr>
<th>Code metadata description</th>
<th>Please fill in this column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current code version</td>
<td>v1</td>
</tr>
<tr>
<td>Permanent link to code/repository used for this code version</td>
<td><a href="https://github.com/WimPouw/TowardsMultimodalOpenScience">https://github.com/WimPouw/TowardsMultimodalOpenScience</a></td>
</tr>
<tr>
<td>Permanent link to reproducible capsule</td>
<td><a href="https://wimpouw.github.io/TowardsMultimodalOpenScience/Index">https://wimpouw.github.io/TowardsMultimodalOpenScience/Index</a></td>
</tr>
</tbody>
</table>
1. Introduction

In this increasingly data-rich world, visual recordings of human behavior are often unable to be shared due to concerns about protecting privacy (Narayanan et al, 2016). Consequently, data sharing in behavioral science, multimodal communication, and human movement research is often limited. Nevertheless, data sharing is crucial in scientific contexts, as it allows for analyses on said data to be computationally reproducible (Buchanan et al., 2021; Gilmore et al., 2018). For sensitive, quantitative data, such as medical records, new methods have been developed to share data without exposing personally identifiable information by creating synthetic data that preserve some statistical aspects of the original data (Abay et al., 2019). For sensitive visual data of human behavior, there are currently no widely available comparable solutions (Joel et al., 2018).

Due to this lack of technical solutions, it is still common in research on multimodal communication and other kinematic research areas to forgo sharing the original video recordings (e.g., Gawne et al., 2019) or to conditionally share them upon request as often required by journal policies (Meyer, 2018; Wilkinson et al., 2016). Unfortunately, researchers seldom ratify sharing requests, as previous research shows (Savage & Vickers, 2009). One way of circumventing certain privacy concerns is to share only the quantified kinematic data. While this may allow other researchers to perform statistical analyses, it obscures the data source and makes it impossible to observe the original recordings. In contrast, the ability to inspect the original video and audio can help researchers understand what is happening and how quantified results relate to further real-world phenomena observable in the data. It also enables third parties to assess the quality of the recordings, the requirements for which may differ depending on the specific research questions, e.g., see Pouw and colleagues (2020, 2021), as well as the particular level of multimodal communication one is interested in (Rasenberg et al., 2020). Furthermore, presenting the actual video recordings rather than just plots and graphs is important for optimally communicating research to peers, such as an academic conference or public lecture. It is, therefore, crucial to find an effective middle ground that allows data to be as anonymous as possible while maintaining pertinent visual information about the context in relation to dynamic human behavior.

Building on advancements in computer vision and deep learning techniques to track full-body kinematic information instantiated in MediaPipe (Lugaresi et al., 2019), we present Masked-Piper as a tool to address the above anonymity-related challenges. Our tool makes use of MediaPipe’s convenient light-weight CPU-based processing pipeline, which we leverage to:

- track hand, body, and facial kinematics, and store the quantitative information as a frame-by-frame time series,
- distinguish a human body from background information,
- mask the human body in the original video while retaining background information,
- project kinematics onto the masked video.
Figures 1 and 2 provide two examples of input frames and the resulting outputs. Researchers can apply Masked-Piper (code here) on a large number of videos by placing the original folders into a processing folder. Masked-Piper will iteratively process all videos storing masked and kinematic videos. The kinematic time series files produced contain time stamp information (based on the frame rate of the original video) next to the available key points.

For the body pose information, 33 key points are available with 3D position coordinates and additional visibility variables, which helps judge the reliability of position estimates. For hand kinematics, 42 position key points are tracked in 3D; for facial kinematics, 3D position coordinates are provided for a face-mesh, containing 478 key points corresponding to designated areas of the face mesh (for more information see https://google.github.io/mediapipe/solutions/holistic.html). Masked-Piper utilizes MediaPipe’s kinematics drawing module to project the kinematic information on top of the masked video. Information about the face mesh coordinates, body pose, and hand kinematics is maintained in the video.

In addition, MediaPipe’s drawing module renders a full-body pose in a 2D space that aligns with the original video at each frame. Thus, Masked-Piper masks original visual information and reinstates key bodily information in the video in de-identified form.

The current tool has applications beyond the behavioral sciences. Firstly, it can solve the problem of collecting unnecessarily superfluous information about human behavior. Consider, for example, that many (audio)visual surveillance systems might not require recording a person’s identity but still record such information. Using our masking tool, such systems might be employed to monitor certain activities (e.g., running) or levels of activities (e.g., amount of people), which do not require amassing identifiable information. The current tool could resolve this issue of superfluous information that inflates privacy risks by only maintaining relevant information about human behavior while mitigating privacy risks. We envision

---

many other applications of the current tool, such as in legal contexts, where hearings with witnesses can be recorded to reduce the risk of identity exposure while maximizing the embodied communicative information inherent to human communication.

2. Implementation

The tool is currently implemented in Python, with all required code provided in the supplementary material. Users need to install the required modules, copy the files to be processed into a local directory, and run the provided notebook. The open nature of the code provides additional transparency into how the tool works and allows customizability on the user’s part. Nonetheless, the notebook is fully functioning in its current form and thus does not require much technical expertise.

As described above, Masked-Piper carries out several processing steps for each video provided. First, MediaPipe motion tracking is applied to the video, such that hand, body, and facial key points are located on each frame, providing time series for each of the key points, in x,y coordinates (given in pixels, local to the still frame). These are collected and provided as output in CSV format for further processing, sharing, etc. In parallel to collecting the key point-based tracking data, the holistic module from MediaPipe automatically detects a silhouette of the person in the video frame and extracts it from the background. This silhouette subsequently serves as the mask.

To accomplish this, Masked-Piper draws this silhouette in black on top of the current video frame. The key point positions are then drawn onto the same silhouetted frame using the MediaPipe drawing module. Finally, the silhouetted (i.e., masked) marked frames are saved as a new video file using OpenCV. This new video file is the pseudo-anonymized output video, which preserves the holistic context of the video (e.g., background, human subject within the background) and provides more fine-grained information about the position of the limbs and fingers, facial expression, mouth movement, etc.

---

Figure 2 Another example of input frame (left) and masked frame (right) for a non-adult speaker (Source: TedX²)

² https://www.youtube.com/watch?v=OMbNoo4mCcl
2.1 Choice of framework

The MediaPipe framework affords analysis of complex and dynamic bodily behaviors and is more easily installed than more heavy-duty GPU-based approaches to tracking human poses such as FrankMocap (Rong et al., 2020). The value of using MediaPipe is (1) a good balance of resource consumption, (2) incremental and iterative processing, and (3) a broad library of supporting toolkits/libraries for developers and researchers to select from and customize (Lugaresi et al., 2019).

2.2 Modification of underlying tool

Our modification of the MediaPipe tool is to consider the body silhouette to distinguish the background from the body contained in the video, then tracks the body, and finally, creates a new video that only retains the background, masks the body, and overlays the kinematics back onto the mask. We further modify the original code so that time series data provide all kinematic information per frame over time. This tool is thus convenient for researchers to mask videos and extract kinematic time series for their research using a next-generation body tracker going beyond slower 2D tracking systems such as OpenPose (Cao et al., 2017).

3. Discussion

Any tool that risks exposing personally identifiable information needs to be used carefully. Several paths may lead to the exposure of someone's identity. Since we preserve information about communicative body movements and speech content, it is clear that any identifiable audio information remains unmasked for audio-visual recordings. Further, we can imagine cases where what is said or how one moves can still be sufficient to retrieve someone's identity (Runeson et al., 1983) in cases where you know the person in question well, and know their potentially unique ways of communicating. The masking tool should therefore be seen as considerably reducing risks of identity exposure. In behavioral science, identity exposure risks due to familiarity are not generally applicable though, as the researchers using the video recordings often have no connection with the study sample. Indeed, body movement data are not considered identifiable by any legal standard (e.g., GDPR guidelines). Another limitation of the current tool is that only one body per frame can be detected. Thus, further iterations of the masking tool will need to be developed to mask multiple persons in one video. Finally, any automated computer vision-based tracking may be insufficiently precise depending on your research questions. Fortunately, researchers can easily verify the quality of the videos and tracking performance produced by Masked-Piper.

The careful use of Masked-Piper has the potential to improve ethical research practices as well as maximize open science practices. It will help researchers to mask videos in large quantities, which they can then easily share with their peers. This will indirectly also improve the core of the scientific

3 https://google.github.io/mediapipe/solutions/holistic.html
process itself, because research reproducibility increases by allowing other researchers easy access to
the original research context as contained in the video recordings.

Funding
This research has been co-funded by a VENI grant (VI.Veni.201G.047) awarded by the Dutch Research
Council (NWO) to Wim Pouw (PI).

Acknowledgments
We wish to thank Linda Drijvers, Judith Holler, Asli Ozyurek and their affiliated research groups for their
helpful comments on the masking tool. We are also grateful to the multimodal community on Twitter that
couraged us to explore such a tool and its value. We acknowledge continued support from Max Planck
Institute Donders Institute, the Hasso Plattner Design Thinking Research Program, particularly Jonathan
Edelman and Joaquin Santuber. The collaboration between colleagues at the HPI AI lab from Victor
Omolaoye and funding from the Hasso Plattner Foundation is also acknowledged.

References
release using deep learning. Lecture Notes in Computer Science (Including Subseries Lecture Notes in
Artificial Intelligence and Lecture Notes in Bioinformatics), 11051 LNAI, 510–526.
https://doi.org/10.1007/978-3-030-10925-7_31

Buchanan, E. M., Crain, S. E., Cunningham, A. L., Johnson, H. R., Stash, H., Papadatou-Pastou, M., Isager, P. M.,
Data Set. Advances in Methods and Practices in Psychological Science, 4(1).
https://doi.org/10.1177/2515245920928007

fields. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017,
2017-January, 1302–1310. https://doi.org/10.1109/CVPR.2017.143

https://doi.org/10.1075/GEST.00034.GAW/CITE/REFWORKS

https://doi.org/10.1177/2515245917746500

Social Psychological Topics: Challenges, Tools, and Future Directions. Advances in Methods and Practices


Supplementary material / Additional Information:
Here we describe the steps to setup Masked-Piper and a demo usage

**Masked-Piper Setup and Usage**

Install all the packages in requirements.txt. Then move the videos that you want to mask into the input folder. Then run the code as shown on the GitHub Page with Examples (and shown below). Running this code will loop through all the videos in the input folder and save all the results in the output folders.
The following video will be processed for masking:
['sample.mp4', 'ted_kid.mp4']

Main Procedure of Masked-Piper

The following chunk of code loops through all the videos loaded into the input folder, assesses each frame for body poses, extracts kinematic info. Next, the code masks the body in a new frame that preserves the background, projecting
the kinematic information on the mask. In addition, the code stores the kinematic information for that frame into the time series .csv for the hand + body + face.

```python
# Make a loop
for vid in video_names:
    # Capture the video, and check video settings
    videoname = vid
    videoPath = "./Input_Videos/" + videoname
    capture = cv2.VideoCapture(videopath)  # Load the video file
    frameWidth = int(capture.get(cv2.CAP_PROP_FRAME_WIDTH))  # Get the frame width
    frameHeight = int(capture.get(cv2.CAP_PROP_FRAME_HEIGHT))  # Get the frame height
    fps = int(capture.get(cv2.CAP_PROP_FPS))  # Get the frames per second

    Make an empty video file where we capture the pose tracking on
    four = cv2.VideoWriter_fourcc("*AVC1")  # For different video formats you could use e.g., "*IVD1"
    out = cv2.VideoWriter("./Output_MaskedVideos/" + videoname + ".csv", four, 
                          fps, (int(frameWidth), int(frameHeight)))

    # For each scene in the video, find all objects, which start with name "personX"
    frames = videocapture.get(cv2.CAP_PROP_FRAME_COUNT)  # Get the total number of frames
    # Store the results of the kinematic pose tracking
    for frame in range(frames):
        # Read the frame from the video
        ret, frame = capture.read()
        # If the frame is not an image, skip it
        if not ret:
            continue

        # Process the frame and store the results of kinematic pose tracking
        x, y = frame.shape
        original_image = cv2.cvtColor(frame, cv2.COLOR_BGR2YCrCb)
        ycrcb = cv2.split(original_image)
        y = ycrcb[0]
        frame = cv2.merge([y, cv2.resize(frame, (x, y))])
        # Process the frame and store the results of kinematic pose tracking
        for frame in range(frames):
            # Read the frame from the video
            ret, frame = capture.read()
            # If the frame is not an image, skip it
            if not ret:
                continue

            # Process the frame and store the results of kinematic pose tracking
            x, y = frame.shape
            original_image = cv2.cvtColor(frame, cv2.COLOR_BGR2YCrCb)
            ycrcb = cv2.split(original_image)
            y = ycrcb[0]
            frame = cv2.merge([y, cv2.resize(frame, (x, y))])
            # Process the frame and store the results of kinematic pose tracking
```

Done with processing all folders; results are in your output folders!