

# SignCollect: A ‘Touchless’ Pipeline for Constructing Large-scale Sign Language Repositories

Gomèr Otterspeer<sup>✉</sup>, Ulrika Klomp<sup>✉</sup>, Floris Roelofsen<sup>✉</sup>

g.otterspeer@uva.nl, u.klomp@uva.nl, f.roelofsen@uva.nl

## Abstract

The projectteam of the Signbank project at the University of Amsterdam intends to substantially extend the NGT lexicon in Global Signbank within a limited timespan. To make this possible, the signCollect platform was developed to automate a major part of the workflow. The signCollect system includes a ‘touchless’ interface which enables a signer to control the system through simple gestures (recognized using computer vision) to (i) prompt the display of the next lexical entry, (ii) start a new recording, and (iii) approve/disapprove a recording. This capability allows a signer to record between 60 to 120 signs per hour, without the need for any assisting staff to be present. The approved recordings immediately become visible in the signCollect database, so that other members of the team can add metadata. With feedback from workshop participants we intend to further optimize the signCollect platform and make it available as an open-source tool for all sign language research teams.

**Keywords:** Sign Language, Data Collection, Automated Data Harvesting

## 1. Introduction

### 1.1. The Signbank Project

The Signbank project at the University of Amsterdam aims at extending the NGT (Sign Language of the Netherlands) dataset in Global Signbank. Global Signbank is a lexical database utilized to collect, store and display signs from various sign languages around the world. Various research institutes employ Signbank for the publication and collection of lexical entries and use it to conduct field research (Cassidy et al., 2018). The aim of the current Signbank project is to substantially extend the NGT dataset (see Klomp et al., 2024 for more details on the general project) within a relatively short timespan of 14 months. In this project, the team aims to identify, record, and describe thousands of lexical signs. To automate parts of this process, the first author of this paper – a Deaf programmer and native signer of NGT – developed the signCollect platform.

### 1.2. Overview of the signCollect Platform

signCollect has three components:

1. signCollect Studio for recording;
2. signCollect Dashboard to support collaborative work among team members;
3. signCollect Hub which connects signCollect Studio and signCollect Dashboard to several other components of the larger software ecosystem in which signCollect operates.

The platform aims to facilitate discussions to identify signs to be added to the dataset, collecting

multi-view video recordings and 3D motion capture recordings of signs, managing validation of the recordings, and gathering metadata (phonological and semantic descriptions, discussion notes). All information related to a given sign is collected in one entry, with one Annotation ID gloss (following the structure of lexicons on Global Signbank). The platform streamlines the process of collecting data and aims to serve as a basis for possible future extensions (see Section 7) which may further enhance the data collection process. The NGT dataset on Global Signbank and the signCollect platform running at the University of Amsterdam are synchronised every time there is a change in their databases so that they always share the same dataset (see also Figure ??).

### 1.3. Other Approaches

Other projects working on sign language lexicons or corpora have also developed tools to support data collection and data management. Prominent examples are the GlossLexer system of Hanke et al. (2001) and the iLex system of Hanke and Storz (2008). In comparison to these existing systems, signCollect aims to further optimize the data collection and data management pipeline by automating several steps in the workflow, and also aims for smooth integration with the Global Signbank platform (although it may also be used in projects that do not make use of Global Signbank).

### 1.4. Paper Outline

The paper is organized as follows. First, Section 2 goes into the touchless recording procedure in signCollect Studio – the most innovative component of the signCollect platform. Then, Section 3 discusses

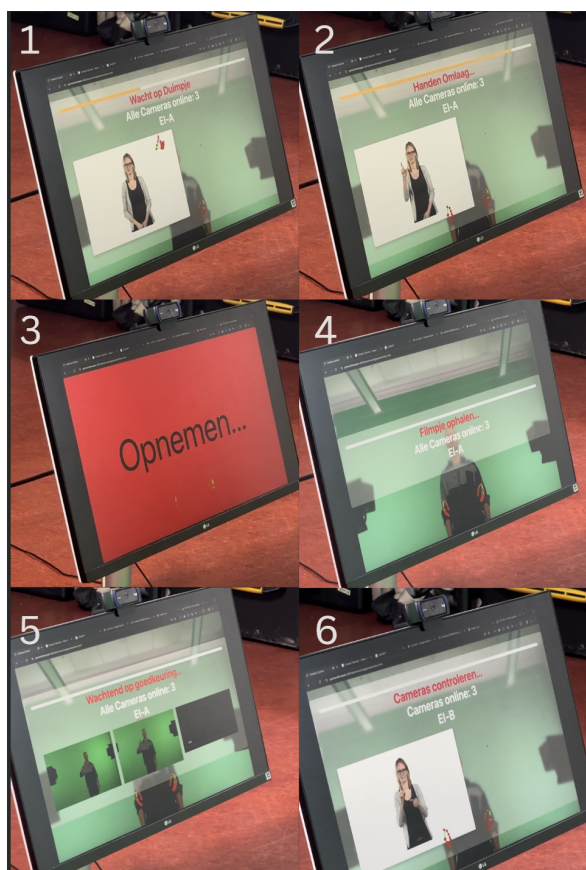


Figure 1: Touchless User Interface

the ecosystem that signCollect is part of, Section 4 turns to the general workflow that signCollect facilitates, Section 5 highlights some specific hardware and software components, Section 6 describes two concrete issues that signCollect helps to resolve, Section 7 discusses some avenues for future extensions, and Section 8 concludes.

## 2. A ‘Touchless’ Recording Pipeline

**Setup** Our basic recording setup involves three cameras. After placing and connecting the cameras and lights, and starting up the signCollect system, the system first checks if the cameras are connected and have the desired settings; if not, it will display a warning pop-up, which means that the settings have to be adjusted. When all settings are as desired, the warning pop-up disappears. The signer views a list of lexical entries that need to be recorded and also has the freedom to add/remove entries. The signer does a first set of recordings to check lights, their position w.r.t. the cameras, and to test that signCollect (which includes Mediapipe Gesture Recognition) correctly recognizes their control gestures (thumbs up and down). Then they can proceed. The steps described below are also visualized in Figure 1.

**Step 1: The signer views the lexical entry** The Annotation ID gloss of a lexical entry is being displayed, together with an earlier made quick recording so the form of the sign associated with this particular Annotation ID gloss is clear. When ready the signer gives a thumbs up for half a second. The interface shows a progress bar that fills to 100 percent as long as the thumb is up and no extreme movements are detected. This means that the progress can be stopped anytime by interrupting the thumbs up gesture.

**Step 2: The signer is ready to produce the sign** The application first checks again if the settings of all cameras are the same, the batteries are at a level of at least 10 percent and that there is sufficient space on the memory cards. Then it displays ‘Ready to Record’. When seeing this, the signer lowers their hands. To avoid the signer looking at the display instead of at the front camera, the display turns bright red, which is a signal that is also visible from the corner of one’s eyes.

**Step 3: Cameras start capturing** The application detects that the hands are down for at least half a second, then it proceeds to send a START CAPTURE signal to all cameras. The cameras send back a confirmation that they are recording. The system displays a message ‘Recording’ with a red background, and the cameras are rolling. Otherwise the background turns blue and a warning message is shown that the application hasn’t received confirmations from all cameras; this can happen when one of the cameras has crashed or when the USB connection is interrupted. When the application has received all confirmations it also produces a high pitch sound for the purposes of synchronizing the video recordings of the different cameras. The signer can start producing the sign.

**Step 4: The sign has been captured** When the signer has produced the sign, the hands are lowered to a rest position. The system recognizes this and shows a progress bar as visual feedback for half a second, as long as there is no other extreme movement detected. The videos captured by the cameras are displayed. Depending on the settings of the user, the videos are either displayed all at once, or alternatively one after the other.

**Step 5: The signer approves or rejects the recordings** To approve the recordings, the signer makes a thumbs up gesture. Again a progress bar will be shown for one second, and then the captured videos are saved in a database to batch process them when the signer has finished recording all the listed signs. However, when the captured videos do not meet the requirements, the signer uses the

thumbs down gesture. The system shows the same lexical entry again to recapture the sign and ignores the rejected videos.

### Step 6: signCollect Studio collects all the data

When all the signs from the list are recorded, a window appears and says: 'ALL DONE'. Then the system asks the cameras to switch from Capture State to Media Transfer state. After that the videos are downloaded. A download progress bar is shown to the signer with a warning not to touch the USB-C cables and cameras. After all video files are downloaded from the cameras to the studio computer, they are uploaded to the signCollect database on a file server. The videos can immediately be viewed by other team members through a web interface. They can validate the video recordings and add metadata.

We now show how this capture procedure fits into the bigger ecosystem and how it relates to data storage and processing.

## 3. The signCollect Ecosystem

Figure ?? shows a diagram of the structure and data flow within the signCollect ecosystem. There are various components with connections between them – the following paragraphs explain the components from left to right.

**Global Signbank** Global Signbank and signCollect communicate with each other, receiving, sending, and acknowledging requests and responses. The communication uses standard JSON arrays to represent lexicon entries. When a user updates a field in signCollect, the platform saves the entry in the database. If the lexicon entry is ready to be published, signCollect sends the JSON array to Signbank. Signbank then updates the entry and replies with 'success' or 'error' to indicate whether the format adheres to the JSON standard and the relevant fields have been successfully updated.

**signCollect** signCollect is connected to all components of the ecosystem and is linked to a database on a separate server. The database contains information of all the lexical entries that are saved via the signCollect interface. Depending on the status of entries, connected components get a request from signCollect to process the metadata of the respective entry.<sup>1</sup>

---

<sup>1</sup>Every entry has fields for preliminary recordings, phonological information, gloss name, and media files. Depending on how many fields are entered and checked the platform defines a certain status for the entry.

**Studio hardware** The Signbank team in Amsterdam uses a video recording studio and a 3D Motion Capture studio. The hardware in these studios is controlled by signCollect, their output is processed and files with metadata are saved in the database on the file server.

**Data Storage** The storage contains video files, motion capture files, FBX/GLB files, a database, and metadata for AI tools. Every file is linked with a signed consent form of the participant involved. If there is a request to process the files, signCollect checks whether permission for publication has been given or not. The platform also regularly checks if consent has expired. In this case, the files are deleted (or in the future possibly anonymized using AI/ML with a human check).

**AI/ML Server** The server contains AI tools that are helpful for processing video files and motion capture files. In the future, additional AI tools will be installed to support the Signbank project team in adding metadata (e.g., automatic recognition of phonological features of a sign like the hand configuration or facial features) and also provides an environment for researchers and students to test new tools on the data (see Section 7).

## 4. The signCollect Workflow

### 4.1. Preparation

Figure 3 provides an overview of the signCollect workflow. During the data preparation phase, the Signbank team creates a list of signs to be added to the dataset. Then they need to make sure these signs do not already exist in the dataset. Therefore, phonological information is added as searchable metadata – a feature already available in Signbank. If the sign is not available in Signbank yet, we proceed by adding a lexical entry and a quick recording for easy reference using a webcam. See Klomp et al. (2024) for further discussion of the procedure of selecting signs.

### 4.2. Recording

When a substantial list of lexical entries with status 'ready-to-record' has been reached, a recording session is planned. The signer checks the list of lexical entries in the signCollect Dashboard before preparing the studio to record the signs. After preparing the studio with the required cameras and lights and checking the connections, the signCollect Studio system is started and checks the connections and properties of all cameras (battery and memory). If no warning is displayed on the screen of signCollect Studio, then the signer proceeds to

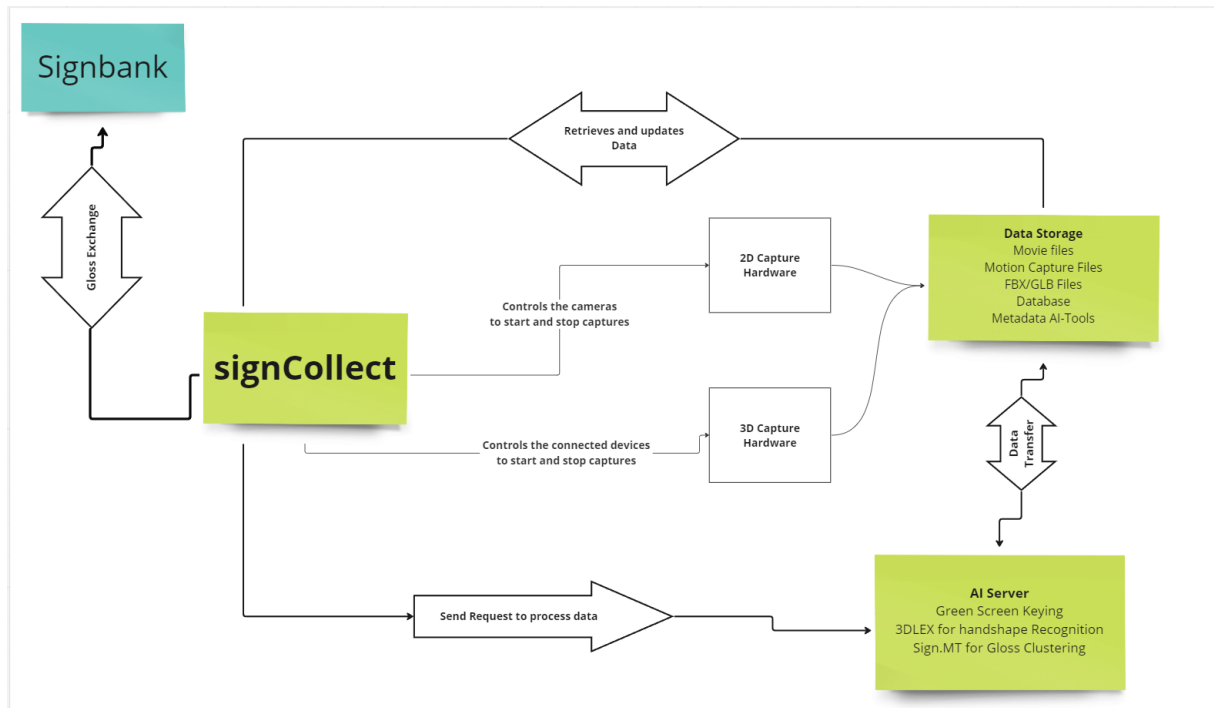


Figure 2: Intranet of signCollect

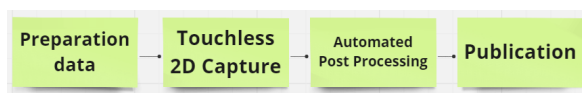


Figure 3: Workflow of signCollect

record by giving a thumbs up towards the main camera. The signer's control gestures are translated into commands by signCollect Studio (Section 5.2). Besides gesture control, signCollect Studio also offers signers another control option, making use of foot pedals. Both control options allow signers to work without interruption and without the need for additional technical assistance staff.

During each recording session, the system automatically verifies that data is stored properly and in the right format, making it immediately accessible to other team members. As described in Section 2, recordings are already reviewed during the recording session by the signer and re-recorded if needed to meet stringent quality standards, including a 60FPS frame rate, 4K resolution, absence of motion blur, proper lighting, and clear visibility of the sign. Recordings are captured from three distinct angles to provide comprehensive visual data. The same rigorous standards apply to 3D captures.

### 4.3. Automated Post Processing

In Automated Post Processing, signCollect Studio automatically retrieves files from the cameras and stores them on a file server. The signs are already

segmented because they are recorded individually. The file server is linked to an AI server, which compresses the files and performs keying, meaning that the green screen background is converted to a color determined by the Signbank project team. The AI server processes the recordings for Mediapipe Landmark Detection, which in future work could be used for recognizing phonological features such as hand configuration through various AI tools. This would further automate the workflow. In the end the recordings that have been made in the studio get the status 'To be checked before publication'. The Signbank project team reviews them and then approves or rejects them. When approved, the recordings get uploaded automatically to Global Signbank. When rejected, they go back to the status 'Ready to be recorded'.

### 4.4. Publication

When the lexical entry has been approved for publication by the Signbank team it receives the status 'Ready for publication'. signCollect proceeds to upload the media files and metadata to Global Signbank. The Signbank server reports back whether the upload has been processed successfully. If the upload is successful, the sign is available on the Global Signbank website. The information also remains available within the signCollect system. Any further changes are synchronised between Global Signbank and signCollect.



## 5. Hardware and software components

### 5.1. Hardware

**Sony FX30 Camera** For the recordings, we employ three Sony FX-30 camera units with a standard lens of 36-105mm at an F/4 aperture, a shutter speed of 1/500, and an ISO setting of either 200 or 400. The cameras are positioned four meters away from the signer at angles of 25, 0 and -25 degrees, respectively. They are connected with USB-C to signCollect Studio which facilitates data transfer and operational commands, such as start/stop controls.

**Tentacle Sync E** Each Sony FX30 camera is connected to its own Tentacle Sync device, which generates time-code for synchronization purposes (Tentacle, 2024). The Tentacle Sync devices are all interconnected via Bluetooth, ensuring they remain synchronized with the same time-code during capture, avoiding time drift between recordings of the same sign by different cameras.

**iPhone with Live Link Face app** Besides three video cameras our studio setup optionally includes an iPhone as well, with the Live Link Face app by Unreal Engine (Live Link Face, 2024) which captures facial expressions of the signer and streams the data to signCollect. This data is saved in blendshapes CSV format and is added as metadata of the given sign.

### 5.2. Software components

**Google Mediapipe** The MediaPipe Gesture Recognition component (Mediapipe, 2024) captures and interprets hand gestures in real-time. Hand movements of the signer are being tracked and translated into labels. For instance, if the signer produces a 'Thumb Up' sign, then it is translated as the 'thumbUp' label. For our specific purposes we created a dataset consisting of thumb up, thumb down, and hands down recordings from different angles and different people and trained a Tensorflow Lite Model compatible with the Mediapipe interface. This model successfully recognizes the relevant control gestures.

**Sony Camera Remote SDK** The Sony Camera Remote SDK (Sony, 2024) provides users an ability to control the cameras. It includes example applications and complete documentation. We developed a new application with the SDK as library to give the signer the ability to control (start and stop) the cameras in conjunction with Mediapipe Gesture Recognition. signCollect Studio also executes

checkBattery(), checkISO(), and more functions to regularly check the health settings of the cameras.

**signCollect API** We have developed a signCollect API, which is integrated in signCollect Studio. It allows all components of the signCollect ecosystem to communicate with each other. When the signer proceeds to the next sign, the API collects and displays information about that sign from the database. Before proceeding to record the API checks the health and status of the cameras. Via Mediapipe Gesture Recognition the API checks whether the signer has given a thumbs up or down sign. After the signer has completed a recording the API checks the files and sends them to the file server.

## 6. Issues solved by signCollect

### 6.1. Issue One: Efficient Recording

In our basic set-up, we use three video cameras. Sometimes, we also capture facial data with Live Link Face on an iPhone. Without a system like signCollect this would mean that every recording session would require the presence of at least one person besides the signer to control and manage the devices. Moreover, the settings of all devices such as ISO, resolution, bit values, frame rate, and color values would have to be adjusted and checked manually before each recording session (because the devices are also used for other projects with different settings). After the recording session, all videos would have to be synchronized and edited using editing software. The editor would also have to manually specify the correct Annotation ID gloss for each recorded sign. All of these steps would require a significant time and would be error-sensitive. By automatizing most steps, data collection and quality control can be done much more efficiently.

### 6.2. Issue Two: Collaboration

With the signCollect system, it is easier for the Signbank team to collaborate and coordinate the procedures of selecting, recording, validating, and publishing lexical items. The system provides a fixed procedure to contribute new items to the dataset, avoiding data fragmentation and pollution.

## 7. Possible Future Extensions

We envision that the signCollect ecosystem may be further enriched in various ways in future work. For instance, each sign in Global Signbank has a phonological description, which currently needs to be provided manually. This requires a major time-investment. One possible direction would be to

implement a functionality in the signCollect system for semi-automatic phonological description (see, e.g., [Ranum et al., 2024](#)).

For the identification of new signs to be added, we envision using Gloss Clustering ([Moryossef, 2023](#)) based on the Corpus NGT ([Crasborn et al., 2008](#)), HoReCo ([De Sisto et al., 2023](#)), and possibly other corpora. Gloss Clustering presents a display of unique glosses per video or dataset with frequency counts and information as to whether it is a known or unknown gloss. This methodology can identify signs that could be considered for inclusion in the lexicon. Leveraging metadata from the corpora, the system could also indicate for each identified sign in which years it occurred most and where it was used.

These tools could aid the Signbank team in the identification and documentation of new signs.

## 8. Conclusion

The signCollect system has been developed to enable more efficient and consistent sign language data collection efforts. The main focus of this paper has been on the innovative ‘touchless’ multi-view video recording pipeline, which we have tested extensively. We can record between 60 to 120 lexical entries per hour and publish them within a day on Global Signbank, without the need for technical experts overseeing the recording and post-processing process. This streamlines the work of the Signbank team and offers other teams opportunities to similarly optimize their data collection processes. With feedback from workshop participants, we intend to further develop the system and make it available as an open-source tool for all sign language researchers.

The current codebase can be found at <https://github.com/rem0g/signCollect>

## 9. Acknowledgments

We thank Rob Belleman and Joey van der Kaaij for providing hardware to develop and run the signCollect software, Dalene Venter for assisting in writing the paper and Casper Wubbolts for testing the signCollect Studio. We gratefully acknowledge funding from the Platform Digital Infrastructures for the Social Sciences and the Humanities (PDI-SSH) in the Netherlands.

## 10. Bibliographical References

Steve Cassidy, Onno Crasborn, Henri Nieminen, Wessel Stoop, Micha Hulsbosch, Susan Even, Erwin Komen, and Trevor Johnston. 2018. Signbank: Software to support web based dictionaries of sign language. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Crasborn, Onno and Bank, Richard and Zwitserlood, Inge and van der Kooij, Els and Ormel, Ellen and Ros, Johan and Schüller, Anique and de Meijer, Anne and van Zuilen, Merel and Nauta, Yassine Ellen and van Winsum, Frouke and Vonk, Max. 2020. *NGT dataset in Global Signbank*. Radboud University, Centre for Language Studies, ISLRN [976-021-358-388-6](#).

Onno Crasborn, Inge Zwitserlood, and Johan Ros. 2008. Corpus NGT. *An open access digital corpus of movies with annotations of Sign Language of the Netherlands*. Centre for Language Studies, Radboud University, Nijmegen.

Mirella De Sisto, Dimitar Shterionov, M DeSisto, D Shterionov, Lien Soetemans, Vincent Vandeghinste, Caro Brosens, and Vlaams Gebarentaalcentrum. 2023. NGT-HoReCo and GoSt-ParC-Sign: Two new Sign Language-Spoken Language parallel corpora. In *CLARIN Annual Conference Proceedings*, page 6.

Thomas Hanke, Reiner Konrad, and Arvid Schwarz. 2001. GlossLexer: A multimedia lexical database for sign language dictionary compilation. *Sign Language & Linguistics*, 4(1-2):171–189.

Thomas Hanke and Jakob Storz. 2008. *iLex – a database tool for integrating sign language corpus linguistics and sign language lexicography*. In *Proceedings of the Third Workshop on the Representation and Processing of Sign Languages at LREC 2008*, pages 64–67.

Ulrika Klomp, Lisa Gierman, Pieter Manders, Ellen Nauta, Gomèr Otterspeer, Ray Pelupessy, Galya

Stern, Dalene Venter, Casper Wubbolts, Marloes Oomen, and Floris Roelofsen. 2024. An Extension of the NGT Dataset in Global Signbank. In *Proceedings of the Eleventh Workshop on the Representation and Processing of Sign Languages at LREC-COLING 2024*.

Live Link Face. 2024. Facial Capture with Live Link | Tutorial — dev.epicgames.com. <https://dev.epicgames.com/community/learning/tutorials/lEYe/unreal-engine-facial-capture-with-live-link>. [Accessed 29-02-2024].

Mediapipe. 2024. Mediapipe - google/mediapipe: Cross-platform, customizable ML solutions for live and streaming media. — github.com. <https://github.com/google/mediapipe>. [Accessed 29-02-2024].

Moryossef. 2023. sign.mt: Effortless real-time sign language translation. <https://sign.mt/>.

Ranum, Otterspeer, Andersen, Belleman, and Roelofsen. 2024. 3D-LEX v1.0 3D Lexicons for American Sign Language and Sign Language of the Netherlands. Submitted for publication. Submitted for publication.

Sony. 2024. Camera Remote SDK | SONY — support.d-imaging.sony.co.jp. <https://support.d-imaging.sony.co.jp/app/sdk/en/index.html>. [Accessed 29-02-2024].

Tentacle. 2024. Tentacle Sync E | tentacle sync — tentaclesync.com. <https://tentaclesync.com/sync-e>. [Accessed 29-02-2024].