Published in Image Processing On Line on 2025–07–00.
Submitted on 2024–06–28, accepted on 2025–06–17.
ISSN 2105–1232 © 2025 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2025.560

A Brief Review and Analysis of Two Methods for Automatic Sign Language Segmentation

Ariel E. Stassi[1], J. Matías Di Martino[2] and Gregory Randall[3]

[1]CENUR Litoral Norte, Universidad de la República, Uruguay
[2]Universidad Católica del Uruguay, Uruguay
[3]IIE, Facultad de Ingeniería, Universidad de la República, Uruguay

Communicated by Pablo Musé *Demo edited by* Quentin Bammey and Ariel E. Stassi

Abstract

Sign language segmentation is a fundamental task in sign language processing to implement automatic translation systems. In this work, we study and compare the performance of two state-of-the-art methods for automatic sign language segmentation: "Automatic Segmentation of Sign Language into Subtitle-Units" [4] and "Linguistically Motivated Sign Language Segmentation" [12]. Each method has an online demo available that can be used to run the approaches here presented on example videos, varying parameters such as considered pose models and probability thresholds. Both methods use pauses and movements of the derived skeletons to detect the limits of a phrase. We consider two datasets, one of American Sign Language (the test set of How2Sign) and one of Uruguayan Sign Language (LSU-DS). For the evaluation, we consider two metrics used in the paper [12]. In the case of LSU-DS, as we have triplets of simultaneous videos taken from different points of view, we propose to use the IoU dispersion among points of view to estimate the coherence of the temporal segmentation of a unique signer simultaneously observed by different cameras. The performances of the different variants of each method are evaluated, showing the limits of the methods, the datasets, and the metrics to capture the quality of the automatic solutions.

Source Code

The source code and documentation for these algorithms are available from the web page of this article[1]. Usage instructions are included in the `README.md` file of the archive. The original implementations of the methods are available here: Automatic Segmentation of Sign Language into Subtitle-Units[2] and Linguistically Motivated Sign Language Segmentation[3].
This is an MLBriefs article. The source code has not been reviewed!

Keywords: sign language processing; sign language segmentation; automatic movement analysis; Uruguayan sign language; LSU

[1]https://doi.org/10.5201/ipol.2025.560
[2]https://github.com/hannahbull/sign_language_segmentation
[3]https://github.com/sign-language-processing/segmentation

1 Introduction

Sign language processing is the domain that focuses on the analysis and treatment of sign language data; examples include sign language recognition and translation as well as sign language production [1, 7]. Sign language translation transforms an input video with sign language content to its written or spoken counterpart [7]. These techniques lie in the intersection between computer vision, machine translation, and linguistics [7]. Linguistics is required to define "meaningful units" of the signed language, computer vision allows extracting relevant information from input videos, and machine translation is used for mapping the signed language into spoken or written language [7]. Sign language translation training generally uses video clips of phrases as input, both the signed content and the corresponding text. Consequently, the performances of sign language recognition and translation systems is often conditioned by the accuracy of the temporal segmentation of the sign language video into meaningful units, e.g., isolated signs or phrases [12].

In this paper, we explore and discuss two state-of-the-art alternatives for sign language segmentation: the method presented in "Automatic Segmentation of Sign Language into Subtitle-Units" [4] (see Section 2.1), and the solution proposed in "Linguistically Motivated Sign Language Segmentation" [12] (see Section 2.2). Our main contributions include (i) comparing the performance of these two methods for sign language segmentation in different phrases, (ii) providing and implementing a platform to test the methods online, (iii) evaluating both strategies on a novel Uruguayan Sign Language dataset LSU-DS, (iv) discussing the limitations of the considered metrics for describing the quality of segmentation solutions.

2 Methods

In this paper, we analyze two state-of-the-art sign language segmentation methods:

- "Automatic Segmentation of Sign Language into Subtitle-Units" [4] hereafter referred to as ASSLiSU.
- "Linguistically Motivated Sign Language Segmentation" [12] hereafter referred to as LMSLS.

Now, we proceed to describe each one briefly.

2.1 Automatic Segmentation of Sign Language into Subtitle-Units

This method segments a sign language video into phrases. To formulate the problem, Bull et al. argue the need for a phrase-like unit to segment sign language videos. Consequently, the authors conceive and define a *subtitle-unit* (SU) as "a segment of video corresponding to temporal boundaries of a written subtitle in an accurately subtitled sign language video" [4]. Here, subtitling sign language videos must be exclusively based on visual cues, and the phrases in the annotated text must be correctly aligned to the identified segments.

2.1.1 MEDIAPI-SKEL

This method was trained on the MEDIAPI-SKEL corpus [3], a dataset of French Sign Language with manually aligned subtitles in the SU sense previously described. The videos are generally *naturally signed*[4], with some rare cases of interpreted sign language. This dataset consists of 27 hours of

[4]In this context, *naturally signed* is sign language as a source language without particular restrictions of grammar, signing time, and other language aspects.

2D skeleton sequences derived from OpenPose [5] and 20187 subtitles. The SU average duration is 4.2 seconds, and videos have an average length of 4.5 minutes. The training, validation, and test data contain 278, 40, and 50 videos, respectively. The OpenPose sequences are derived from videos at different angles, and approximately 20% of the videos contain multiple signers. Each video was annotated in the following way: a high value is associated with the frames inside the time support of a phrase, and a low value if the frames are not part of the phrase [4].

2.1.2 Preprocessing of Skeleton Sequences

As a first step, the authors propose to clean the data by automatically identifying and tracking the dominant signer in the OpenPose sequence and removing irrelevant keypoints for the segmentation task. Briefly, the preprocessing steps proposed are:

1. Normalize the frame rate of each sequence to 25 fps.
2. Discard the legs and feet keypoints, as they are noninformative for sign language segmentation.
3. Identify and track each person from the skeleton sequences.
4. Remove information from non-dominant signer subjects in the scene, following criteria of hand visibility, hand size, hand movement, and a minimum time of appearance of the signer in the scene.
5. Estimate missing skeleton keypoints using past and future available data.
6. Smooth the temporal dynamics of each considered keypoint.
7. Normalize each skeleton sequence so the x and y coordinates have zero mean and unitary variance.

2.1.3 Pipeline for SU Detection

The authors propose to define a Spatio-Temporal Graph Convolutional Network (ST-GCN) from each sequence of (preprocessed) OpenPose skeletal data. ST-GCN follows the same structure proposed in the original article [17]. In the spatial domain, the ST-GCN edges' structure is defined between 2D keypoints at body, face, and hand levels, following the same criteria as the OpenPose model. In the temporal domain, the ST-GCN edges' structure is defined between each considered keypoint and the same one across consecutive time steps [17]. The edges of ST-GCN have learnable weights during the training process. The graph convolutions are computed across the spatial and temporal edges of the graph.

The graph representation model comprises nine layers of ST-GCN units, with 64, 128, and 256 output units for the first three, middle three, and last three layers, respectively [4]. The resulting higher-level feature maps of 256 dimensions are fed into a Bidirectional LSTM (BiLSTM) to do a graph regression to predict the probabilities of boundaries of SU over time. This downstream BiLSTM is the most significant difference from the method proposed by Yan et al. [17], originally devised to solve action classification at a video level. Finally, segment tagging is made by a hard threshold with a default value of 50% in the probability sequence.

Figure 1 summarizes the pipeline for ASSLiSU, in italics we indicate the signals (input/output) involved in each pipeline step.

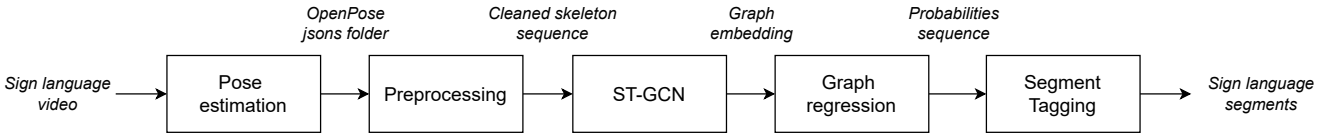


Figure 1: Block diagram with the main stages of ASSLiSU, based on the Figure 3 of [4].

2.1.4 Considered Models

In this work, we considered six models described in [4][5]:

- **full:** It considers all the keypoints in the upper body, i.e., hands, face, and body; the legs and foot keypoints are excluded, as discussed above.
- **body:** It considers the keypoints of the upper body, excluding the face and hands.
- **hands:** It considers only the hands' keypoints.
- **face:** It considers only the face keypoints.
- **face+body:** It considers the upper body and face keypoints.
- **body+hands:** It considers the upper body and hands' keypoints.

Each one of these models determines how the ST-GCN units are built (see Figure 1).

2.2 Linguistically Motivated Sign Language Segmentation

This method segments sign language videos at two levels: individual signs and phrases, i.e., a language unit comprising several adjacent signs. To do this task, the authors propose a model that classifies each input video frame into one of three categories: beginning (B), inside (I), and outside (O) [12]. The so-called BIO tagging was initially proposed for text chunking task [15]. The segmentation task was tackled in a previous work [13] by formulating a frame-level classifier with inside (I) and outside (O) labels as possible outputs. Note that in continuous sign language, a sign or phrase beginning could be next to the previous one without an O frame in between [12]. For this reason, category B was included in the formulation. BIO tagging has been particularly useful for identifying more than 99.7% of segments of the Public DGS Corpus with frame rates of 25 fps or greater [12].

2.2.1 Public DGS Corpus

Public DGS Corpus is a dataset composed of more than 50 hours of video of German Sign Language, frequently abbreviated as DGS, made from recording sessions of one or two native DGS signers. Public DGS Corpus comprises conversations between deaf people, often with a moderator, and has the gloss annotations and translation to two written languages (German and English), including the time spans for each signer, both for the gloss and for the translated phrases [10].

LMSLS method was trained on the Public DGS Corpus, using 714 videos with an average duration of 7.55 minutes at 50 fps. The time support for each gloss was considered the ground truth for the sign segmentation task. Phrase segments were obtained from translations, assuming that the time span is from the start of the first sign to the end of the last sign in the sequence. After filtering some unlabeled data, Moryossef et al. split the data, obtaining 583 videos for training, 12 for validation, and 17 for testing, with an average number of signs and phrases per video of 613 and 111 for the training set, respectively.

[5]There are slight differences between the denominations given in the original paper [4] and the repository, e.g., "head" keypoints in the code repository corresponds to "face" keypoints in the paper.

2.2.2 Input Representation for Phrase Boundaries

The authors model pauses and movement by computing optical flow directly from the sequence of pose estimations. Following the approach from [13], the input for segmentation is the optical flow of a set of 2D points $P = \{(x_1, y_1), \ldots, (x_K, y_K)\}$ corresponding to full-body pose estimation (including body, hand, and face information). A frame rate invariant optical flow F at a time t is defined as

$$F(P)_t = ||P_t - P_{t-1}||_2 \cdot fps, \tag{1}$$

where fps is the frame rate of the input video and $|| \cdot ||_2$ denotes the L2 norm operation. To avoid false movements derived from unreliable pose estimations, in cases where $p \in P$ is not correctly identified in a given frame t, the authors impose $F(p)_t = F(p)_{t+1} = 0$.

2.2.3 Input Representation for Sign Boundaries

Signs generally include a limited number of handshapes. For example, in the American Sign Language Lexicon Video Dataset, the signs are characterized by initial and final handshapes [14]. Studying the SignBank dataset[6], Moryossef et al. showed that more than 80% of the 705151 considered signs are based on one or two handshapes [12]. Based on these observations, the authors propose using a 3D hand normalization in scale and orientation to capture the sign boundaries better. Even if the authors conclude that the 3D hand normalization effect could be negative due to the poor depth estimation, we have decided to include this model variation in our study.

2.2.4 Pipeline for Temporal Segmentation

Moryossef et al. propose the following pipeline for sign language segmentation [12]:

1. **Pose estimation.** The authors adjust the frame rate of the input video to 25 fps and then estimate the body poses using the Mediapipe Holistic Pose estimation [11]. The output of this step is a tensor with dimensions (frames × keypoints × axes).

2. **Pose normalization.** The keypoints corresponding to the legs are removed, and a normalization of the estimated coordinates is applied such that the mean distance between the shoulders of each signer equals 1, and the mid-point between shoulders is at $(0, 0)$ in the xy plane [6].

3. **Optical flow.** Optical flow is calculated by using Equation (1).

4. **3D hand normalization.** The authors look for a consistent representation of 3D handshapes over the frames. They align the back of the hand with the xy plane. Then, the hand is rotated in the xy plane such that the middle finger is aligned with the y axis.

5. **Sequence encoder.** The pose sequence is flattened, projected into a 256-dimensional space, and then fed to an LSTM encoder.

6. **BIO tagging.** On top of the encoder, the authors place two BIO classification heads, one for signs and one for phrases. The authors propose a weighted cross-entropy loss, considering B a much less frequent label than the I and O labels. Empirically, the authors adjust the relation of B:I:O as 1:5:18 for signs, and 1:58:77 for phrases.

[6]Please visit https://auslan.org.au/.

7. **Greedy segment decoding.** The sign/phrase segment beginnings B are obtained at frame level by applying a threshold t_b over the probabilities sequences estimated by the BIO tagging. The endings of the segments are determined by predictions of the O or B label above the thresholds t_o and t_b, respectively.

Figure 2 shows the block diagram of the different strategies for LMSLS: E0, E1, E2, E3, and E4 correspond to different combinations of stages in the pipeline.

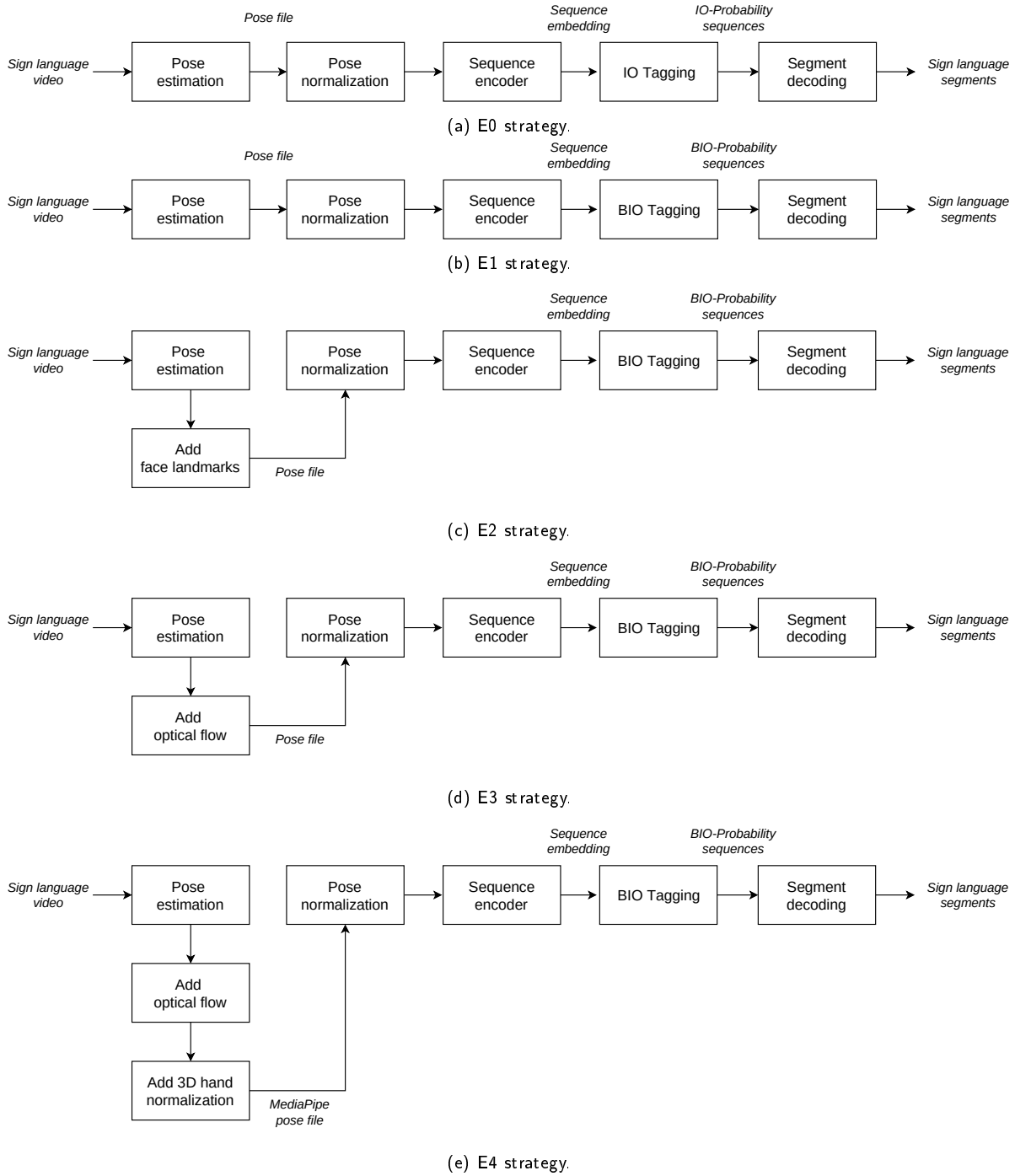


Figure 2: Block diagram with the main stages for different strategies of LMSLS.

2.2.5 Considered Models

In this work, we considered three models, named in the same way as the original paper [12]:

- **E0:** Uses the IO tagger proposed by Moryoseff et al. in [13].
- **E1:** Uses the BIO tagger instead of the IO classifier.
- **E4:** Uses the BIO tagger, including the optical flow and 3D hand normalization in the pose information.

The pair $(t_b; t_o)$ are fixed in percentage to $(60; 50)$ for signs, and $(90; 90)$ for phrases. Unless otherwise indicated, these will be the default values for the experiments presented later in this report.

3 Demo

The demo allows running both methods for carrying out a temporal segmentation of sign language content. It takes as input a raw video, and the user can choose different parameters depending on the selected method. For the demo associated with ideas presented in "Automatic Segmentation of Sign Language into Subtitle-Units," the parameters are the keypoints considered in the construction of the ST-GCN units and the probability threshold for segment tagging, as described in Section 2.1.4. For the demo of the technique presented in "Linguistically Motivated Sign Language Segmentation," the parameters are the strategy used to obtain the segments as well as the threshold values for IO or BIO tagging previously described in Section 2.2.5. The obtained output is a subtitle file with the periods associated with each sign language segment, phrase, or sign, depending on the considered method. In addition to the subtitle file(s), the method outputs an ELAN file to analyze the predicted annotations in a detailed manner. ELAN is a software for textual annotations in video and audio recordings, commonly used in the sign language academic community[7].

4 Experiments

4.1 Sign Language Datasets

4.1.1 How2Sign

How2Sign is a multimodal dataset containing American Sign Language translations [8]. In this work, we only consider the test set. How2Sign test set includes one signer in the scene and comprises 184 videos, totaling 335.52 minutes and 2357 phrases, with an average of 12.8 phrases per video. We considered the frontal view of the Green Screen RGB videos (see Figure 3) and the annotations contained in the English Translation (manually re-aligned) to construct the ground truth.

Figure 3: Example frames of a How2Sign video. Extracted and modified from [8].

[7]For additional information about ELAN software, please visit https://archive.mpi.nl/tla/elan.

4.1.2 LSU-DS

LSU-DS is a dataset of Uruguayan Sign Language samples [16]. The considered subset corresponds to the Linguistic Task 3 of LSU-DS, composed of 190 phrases captured by three simultaneous cameras in controlled lighting conditions and with a fixed background. Figure 4 illustrates each used camera's point of view (POV). The phrases were made by nine signers and are isolated, and the beginning and the end times are annotated, as well as the gloss in each video. The used subset includes 570 videos, totaling 78.78 minutes, i.e., 26.26 minutes for each POV.

Figure 4: POV of each considered camera in LSU-DS: 'cam2', 'cam0', and 'cam1' correspond to the left, central, and right views, respectively.

4.2 Metrics

In the present study, we follow two evaluation metrics used in [12] in combination with two new metrics proposed here:

- **Intersection over Union (IoU)**. Let S be the set of sign language estimated segments for a given video and S_g the ground-truth set of segments. Then, $\text{IoU} = \frac{S \cap S_g}{S \cup S_g}$. This metric is typically used in segmentation problems to measure the overlap between the estimated and the ground-truth segments for each considered input. In general, $0 \leq \text{IoU} \leq 1$ and the greater the IoU the better. Note that IoU was calculated without any alignment or post-processing of the estimated segments concerning the ground-truth.

- **Percentage of Segments (Seg)**. Given a method, let $\#S$ be the cardinality of the estimated sign language segments for a given video and $\#S_g$ the cardinality of the ground-truth set of segments. Then, $\text{Seg} = \frac{\#S}{\#S_g}$, $0 \leq \text{Seg} \leq \infty$. Ideally, $\text{Seg} = 1$. As explained in Section 4.1.2, $\#S_g = 1$, for all the LSU-DS videos; then $\text{Seg} = \#S$.

- **IoU dispersion among POV**($\bar{\sigma}_{\mathbf{POV}}$). Let's call a *multi-video* a set of simultaneous videos taken from different points of view of the same scene. Let N be the total number of considered POV videos for each multi-video. Let IoU_i be the metric for the $i-$POV of a given multi-video, with $i = 1, \ldots, N$. In the case of LSU-DS, $N = 3$, and we talk about triplets. Then $\bar{\sigma}_{POV} = \frac{1}{M} \sum_{m=1}^{M} \sigma_m$, where M is the number of multi-videos and σ_m is the standard deviation of $\{\text{IoU}_1, \ldots, \text{IoU}_N\}$. When a method does not give an output for all the POV of a muti-video, we eliminate this instance from the metric computation. This paper proposes this metric to quantify each method's robustness to changes in POV.

- **Number of Empty Outputs (NEO)**, this paper proposes this metric to quantify the videos without segmentation output for a given POV–method–model combination. Ideally, $\text{NEO} = 0$. NEO will be reported as a percentage concerning the total number of inputs.

4.3 Experiment 1: Phrase Segmentation in How2Sign Test Set

In this first experiment, we are interested in evaluating the performance of the methods studied for phrase segmentation in the test set of How2Sign previously described in Section 4.1.1. Table 1 shows that the LMSLS variants perform better than ASSLiSU in the IoU metric sense.

	IoU ↑	Seg^{+}	NEO ↓	IoU* ↑	Seg*	IoU** ↑	Seg**
ASSLiSU "full"	0.63 ± 0.20	4.15 (41.0)	0	0.74 ± 0.10	2.62 (6.6)	0.77 ± 0.09	2.88 (6.1)
ASSLiSU "body"	0.69 ± 0.22	2.96 (35.0)	0	0.81 ± 0.12	1.7 (4.5)	0.85 ± 0.11	1.86 (4.5)
ASSLiSU "hands"	0.54 ± 0.17	6.90 (73.5)	0	0.62 ± 0.09	4.36 (15.6)	0.63 ± 0.10	4.66 (10.8)
ASSLiSU "face"	0.64 ± 0.21	3.69 (36.5)	0	0.75 ± 0.11	2.31 (4.9)	0.79 ± 0.10	2.45 (4.9)
ASSLiSU "face+body"	0.68 ± 0.22	3.34 (39.5)	0	0.81 ± 0.11	1.87 (4.8)	0.84 ± 0.10	1.92 (4.5)
ASSLiSU "body+hands"	0.65 ± 0.21	4.25 (50.5)	0	0.77 ± 0.11	2.37 (5.6)	0.81 ± 0.11	2.52 (5.6)
LMSLS E0	0.77 ± 0.24	1.37 (16.5)	0	0.89 ± 0.17	0.65 (1.7)	0.90 ± 0.20	0.68 (1.7)
LMSLS E1	0.77 ± 0.22	4.23 (44.5)	0	0.90 ± 0.07	2.63 (8.2)	0.92 ± 0.05	2.67 (5.4)
LMSLS E4	0.77 ± 0.22	4.40 (43.0)	0	0.89 ± 0.06	2.81 (10.6)	0.92 ± 0.04	2.83 (7.0)

Table 1: Metrics for different POV–method–model combinations for phrase segmentation on the How2Sign test set videos (best values , second-best values). IoU is reported as mean±standard deviation. Seg is reported as the mean among the considered videos, and its maximum value is between parenthesis. $^{+}$Seg is better the closer the value is to 1. NEO is reported in percentage. The metrics marked with * and ** correspond to videos with an annotation coverage greater than 90% (61 from the original 184 videos) and 95% (25 from the original 184 videos), respectively.

The evaluated methods tend to overestimate the number of segments (see Table 1). In some cases, they produce more than 70 times the ground-truth number of segments. To understand this fact, we analyzed the output segmentation for some illustrative samples. Figure 5 shows the segmentation produced by all the combinations of methods and strategies for the video 'G3HKHxevpFI-5-rgb_front.mp4'. This sample was chosen because Seg = 1 and the IoU $\approx$ 1 for the E0 strategy of LMSLS. In this situation, both metrics can be interpreted as showing an ideal behavior for this strategy. However, the figure shows that even if the number of segments is correct, approximately half of the detected limits are wrong. The IoU is roughly accurate, but it does not consider the actual structure of the segments. This is true for all the combinations of methods and strategies, as shown in the figure, even if the performance is worse than the E0 one.

Figure 6 shows the estimated segments for the sample 'fZq8wTAYtmw-10-rgb_front.mp4', selected because it has the worst IoU value for all the strategies of the ASSLiSU. In this case, practically the entire video lacks annotations. This is a serious problem for the correct evaluation of these methods, and this fact must inform the interpretation of the results reported in Table 1. The high values of the Seg metric could be due to the poor performance of the studied methods, but they can also be explained by missing ground-truth annotation values. To evaluate the performance on the test samples with good annotation coverage[8], we reported the performance metrics on the partitions with more than 90% and 95% of annotation coverage (columns indicated with * and ** in Table 1) the impact of missing annotations in the potential performance evaluation motivated experiments 2 and 3.

Figure 7 shows the estimated segments for the sample 'G42xKICVj9U-5-rgb_front.mp4', an example of How2Sign with an annotation coverage greater than 95%. In this case, IoU = 0.953 and IoU = 0.957 for the strategies E1 and E4 of LMSLS, and Seg = 1 for the "full" strategy of ASSLiSU. Figure 7 shows that none of the methods segment the video according to the How2Sign annotations. In this example, we can see clear segment detections at the beginning and end of the video –seconds 0 and 120, respectively– and the pause around the second 66 was correctly detected by E1 and E4 and partially detected by E0 of LMSLS. In this case, it can be observed that around the second 33,

[8]In this context, we define the *annotation coverage* as the time proportion of a video that an expert annotates.

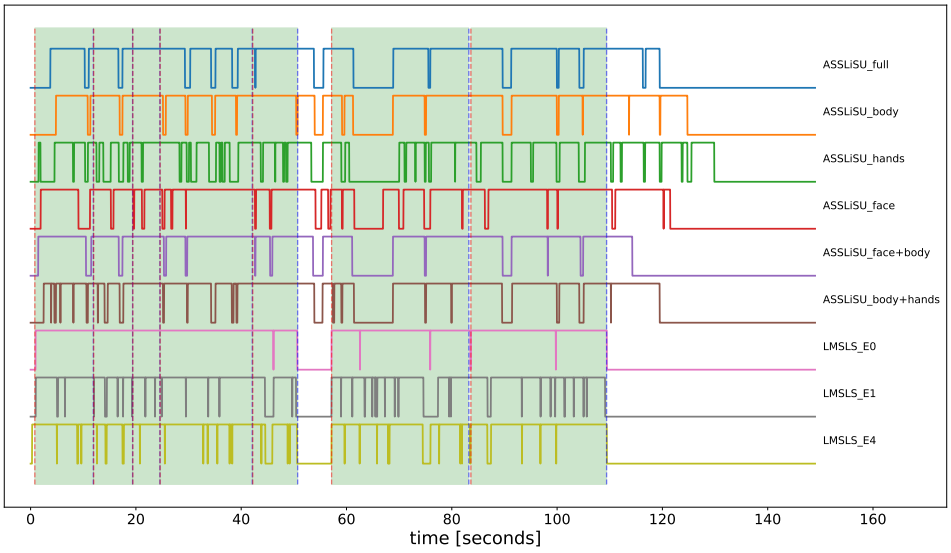


Figure 5: Estimated segments of all combinations of methods and strategies for the video 'G3HKHxevpFI-5-rgb_front.mp4' from the How2Sign test set. Ground-truth segments are marked in transparent green. Dashed red and blue lines signal the beginning and end of each ground-truth segment, respectively. The time axis (horizontal) duration corresponds to the total video duration. In each case, the high state signals the presence of a phrase, and the low state signals its absence. In general, it can be noted that all the combinations (except for LMSLS E0) tend to generate a larger amount of shorter segments w.r.t. the indicated segments in the data annotations.

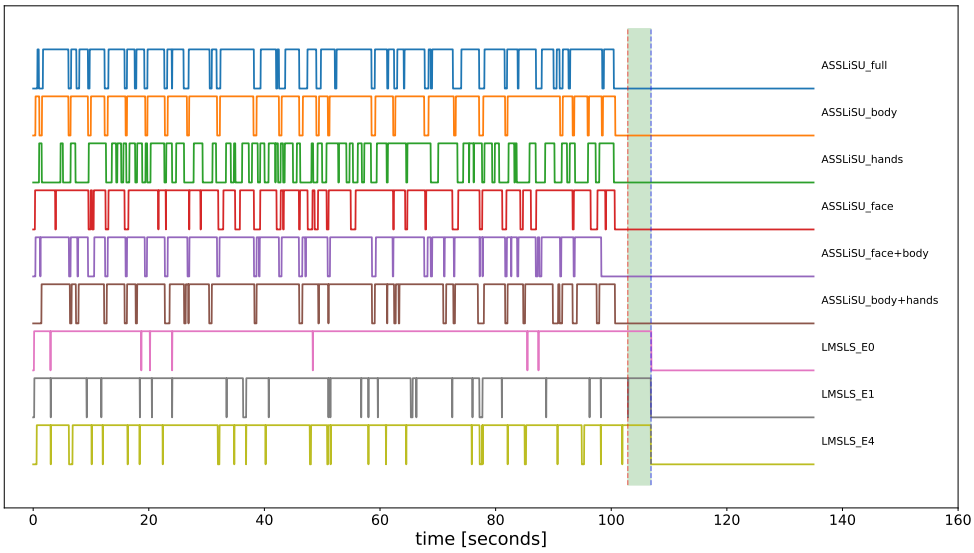


Figure 6: Estimated segments of all combinations of methods and strategies for the video 'fZq8wTAYtmw-10-rgb_front.mp4' from the How2Sign test set. Ground-truth segments are marked in transparent green. Dashed red and blue lines signal the beginning and end of each ground-truth segment, respectively. The duration of $x-$axis was taken from the total video duration. In each case, the high state signals the presence of a phrase, and the low state indicates its absence. In this case, please note that almost the whole timeline lacks annotations. This fact motivated us to consider only those videos with an annotation coverage above a given threshold, as previously explained.

there are overlapping subtitles in the labels. Although this is a labeling error, it is another aspect to consider when working with this method and data.

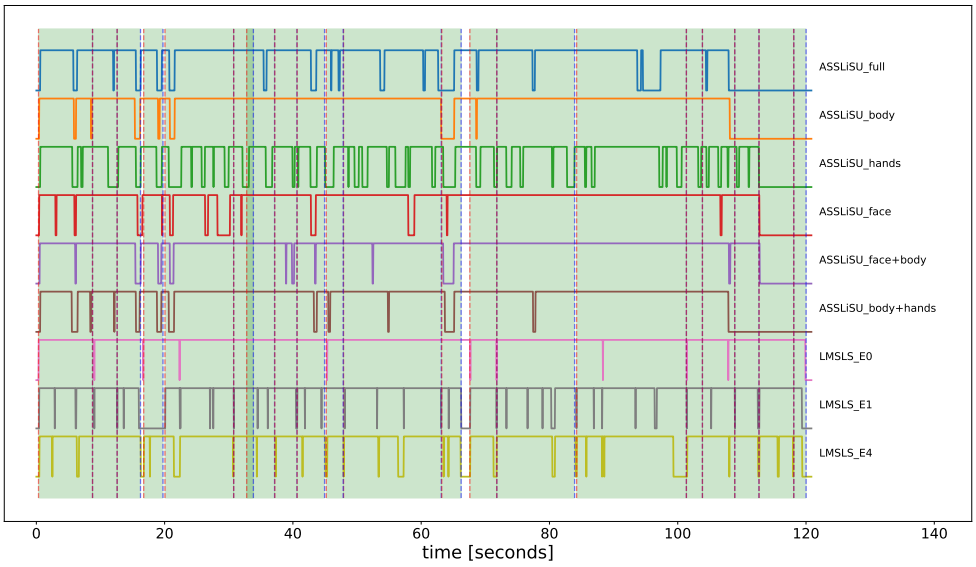


Figure 7: Estimated segments of all combinations of methods and strategies for the video 'G42xKICVj9U-5-rgb_front.mp4' from the How2Sign test set. Ground-truth segments are marked in transparent green. Dashed red and blue lines signal the beginning and end of each ground-truth segment, respectively. The duration of $x-$axis was taken from the total video duration. In each case, the high state signals the presence of a phrase, and the low state indicates its absence. In this particular case, it can be observed that around the second 33, there are overlapping segments in the annotated phrases of this video.

4.4 Experiment 2: Phrase Segmentation in LSU-DS

In this experiment, we analyze the Linguistic Task 3 from the LSU-DS dataset (see Section 4.1.2) and evaluate the performance of both methods for the input video phrase segmentation. Since this set has only one phrase per video, with starts and endings in the signer's resting position and is correctly annotated and curated, it provides a great test set to evaluate models without missing annotations. This data allows for evaluating the methods under controlled conditions, measuring differences between strategies and POVs, and identifying the best and worst performance cases.

	IoU ↑				Seg$^+$			NEO ↓		
	cam0	cam1	cam2	$\bar{\sigma}_{POV}$ ↓	cam0	cam1	cam2	cam0	cam1	cam2
ASSLiSU "full"	0.65 ± 0.27	0.72 ± 0.24	0.65 ± 0.28	0.141	1.19 (3)	1.17 (3)	1.24 (3)	0	0	0
ASSLiSU "body"	0.66 ± 0.26	0.65 ± 0.24	0.53 ± 0.28	0.157	1.20 (4)	1.23 (3)	1.15 (3)	11.1	9.5	11.6
ASSLiSU "hands"	0.65 ± 0.27	0.66 ± 0.26	0.64 ± 0.28	0.146	1.32 (4)	1.29 (4)	1.22 (3)	0	0	0
ASSLiSU "face"	0.54 ± 0.26	0.55 ± 0.24	0.57 ± 0.25	0.144	1.36 (4)	1.46 (4)	1.51 (5)	1.6	1.1	1.1
ASSLiSU "face+body"	0.72 ± 0.24	0.69 ± 0.28	0.67 ± 0.29	0.139	1.20 (3)	1.11 (3)	1.15 (4)	0	0	0
ASSLiSU "body+hands"	0.58 ± 0.29	0.60 ± 0.28	0.54 ± 0.29	0.175	1.54 (5)	1.52 (7)	1.73 (5)	0	0	0
LMSLS E0	0.83 ± 0.13	0.84 ± 0.13	0.81 ± 0.16	0.037	1.47 (6)	1.42 (6)	1.54 (6)	6.3	5.3	5.8
LMSLS E1	0.79 ± 0.19	0.83 ± 0.15	0.76 ± 0.20	0.076	1.62 (5)	1.65 (4)	1.68 (5)	1.1	0	1.6
LMSLS E4	0.82 ± 0.18	0.81 ± 0.15	0.78 ± 0.19	0.063	1.50 (4)	1.65 (4)	1.53 (5)	0.5	0	0

Table 2: Metrics for different POV–method–model combinations for phrase segmentation on LSU-DS videos (best values , second-best values). IoU is reported as mean±standard deviation, except for the column $\bar{\sigma}_{POV}$ (see text for an explanation). Seg is reported as the mean among the considered videos, and its maximum value is between parenthesis. $^+$Seg is better the closer the value is to 1. NEO is reported in percentage.

As shown in Table 2, all the explored methods present a regular to good performance in terms of the IoU metric. Concerning ASSLiSU, the "full" and "face+body" strategies show the best IoU results for all the views (cameras and the variation between them ($\bar{\sigma}_{POV}$)). LMSLS's best IoU values are highlighted in red and correspond to method E0 (note that these are also the best values for all the tested methods). Note also that LMSLS E0 has the smallest $\bar{\sigma}_{POV}$, indicating that this model is the most stable between POVs. Considering the metric Seg, the best flavors of ASSLiSU were again the "full" and "face+body" strategies, which showed the lowest Seg values and the lowest number of outlier segments (between parenthesis in the table). The results provided by LMSLS are generally worse for this metric. Finally, the two strategies with the best performance in the IoU and Seg sense for ASSLiSU also have ideal NEO values. On the other hand, the best LMSLS strategy produces videos without estimated segments in the order of 5% or even more.

Figure 8 shows the segments produced by each method and strategy combination for the video 'T3_S05_sent01_r2_cam0.mp4' from LSU-DS. In this particular case, it can be noted that the methods ASSLiSU "hands", ASSLiSU "face+body", and LMSLS E0 perform flawlessly in the sense of the Seg metric but with considerable differences for the IoU metric. On the other hand, ASSLiSU "full" estimates a shorter segment than the ground-truth one. Finally, in the case of ASSLiSU "body", ASSLiSU "body+hands", LMSLS E1, and LMSLS E4, the strategies estimate one extra segment with a diversity of IoU values. Note that there are wrong-detected segments related to the beginning (ASSLiSU "body+hands") and the video's ending (LMSLS "E4").

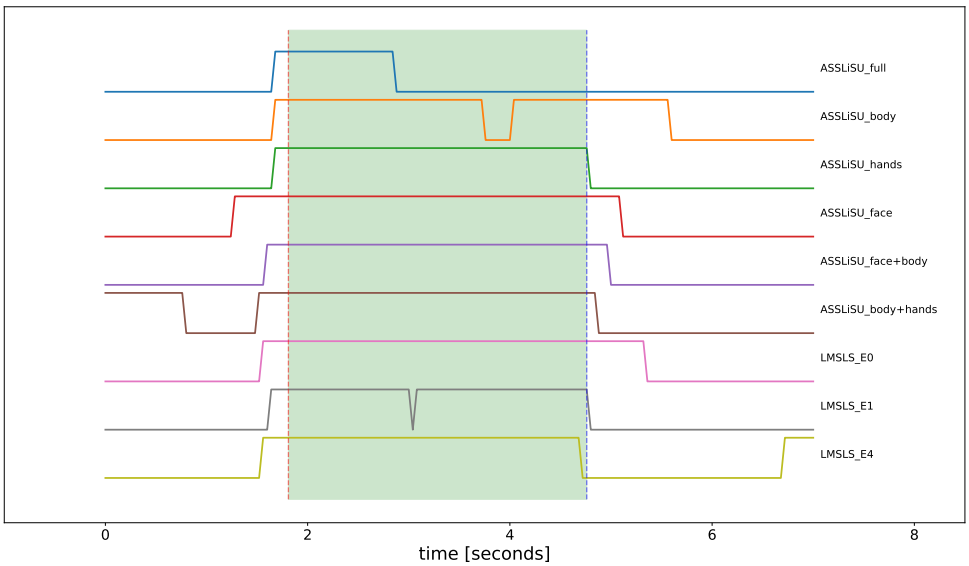
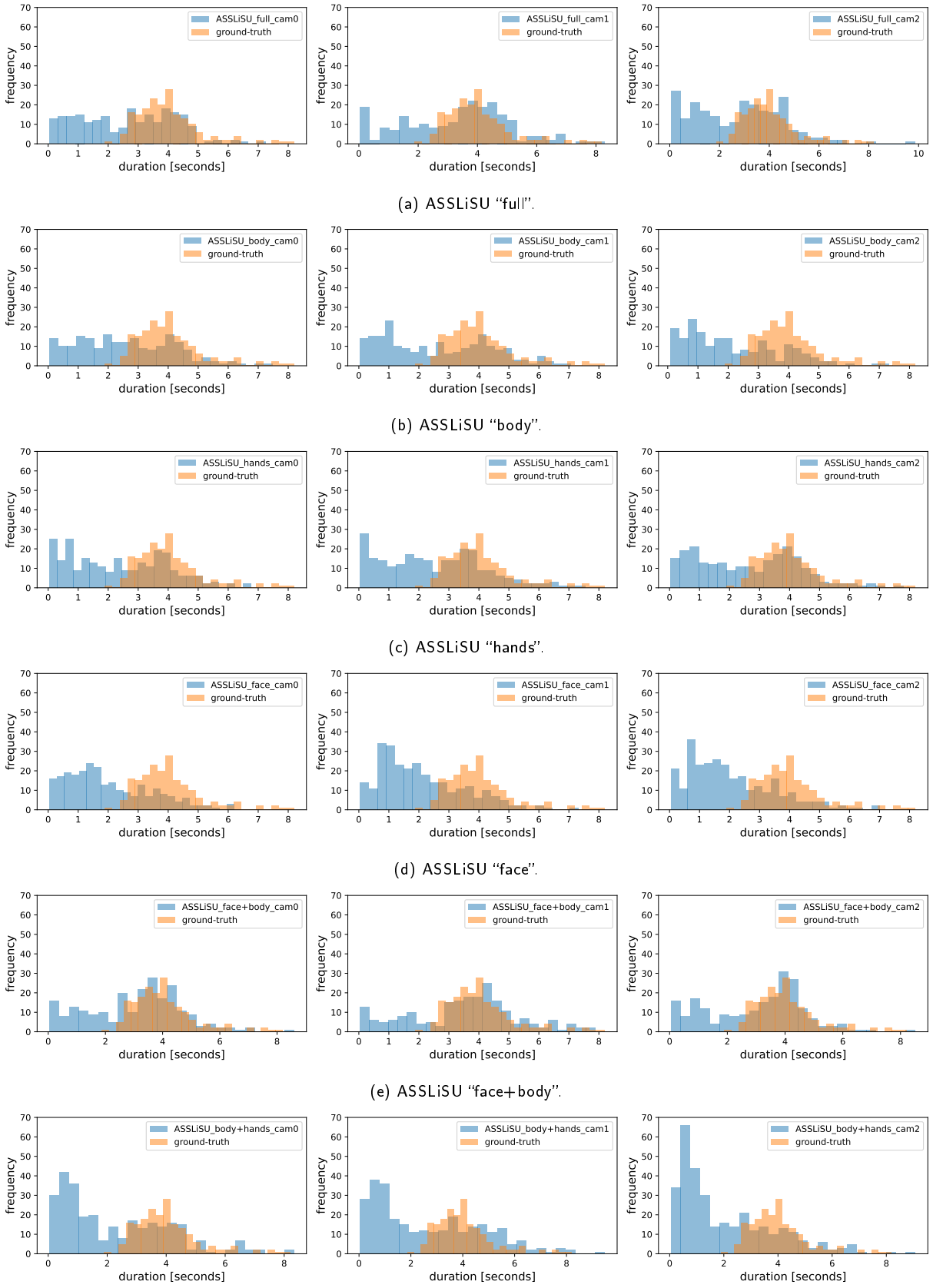


Figure 8: Estimated segments of all combinations of methods–strategies for the video 'T3_S05_sent01_r2_cam0.mp4' from the LSU-DS dataset. Ground-truth segments are marked in transparent green. Dashed red and blue lines signal the beginning and end of each ground-truth segment, respectively. The duration of $x-$axis was taken from the total video duration. In each case, the high state signals the presence of a phrase, and the low state indicates its absence.

Figures 9 and 10 show the histograms of the duration estimated by each combination of methods and strategies overlaid upon the histogram of the ground-truth duration for the three considered POVs. In both figures, there are many short-estimated segments. For ASSLiSU, the "body+hands" strategy presents the worst performance in this sense, especially in camera 2 (Figure 9f). This observation agrees with the results reported in Table 2. The ASSLiSU "hands" strategy produces many short segments, showing only a few estimations over 4 seconds (Figure 9c). This phenomenon is possibly due to hand pose noisy detection or the need to fine-tune the probability sequences segmentation thresholds in the BiLSTM stage in ASSLiSU or the Greedy Decoding step in LMSLS.



(a) ASSLiSU "full".

(b) ASSLiSU "body".

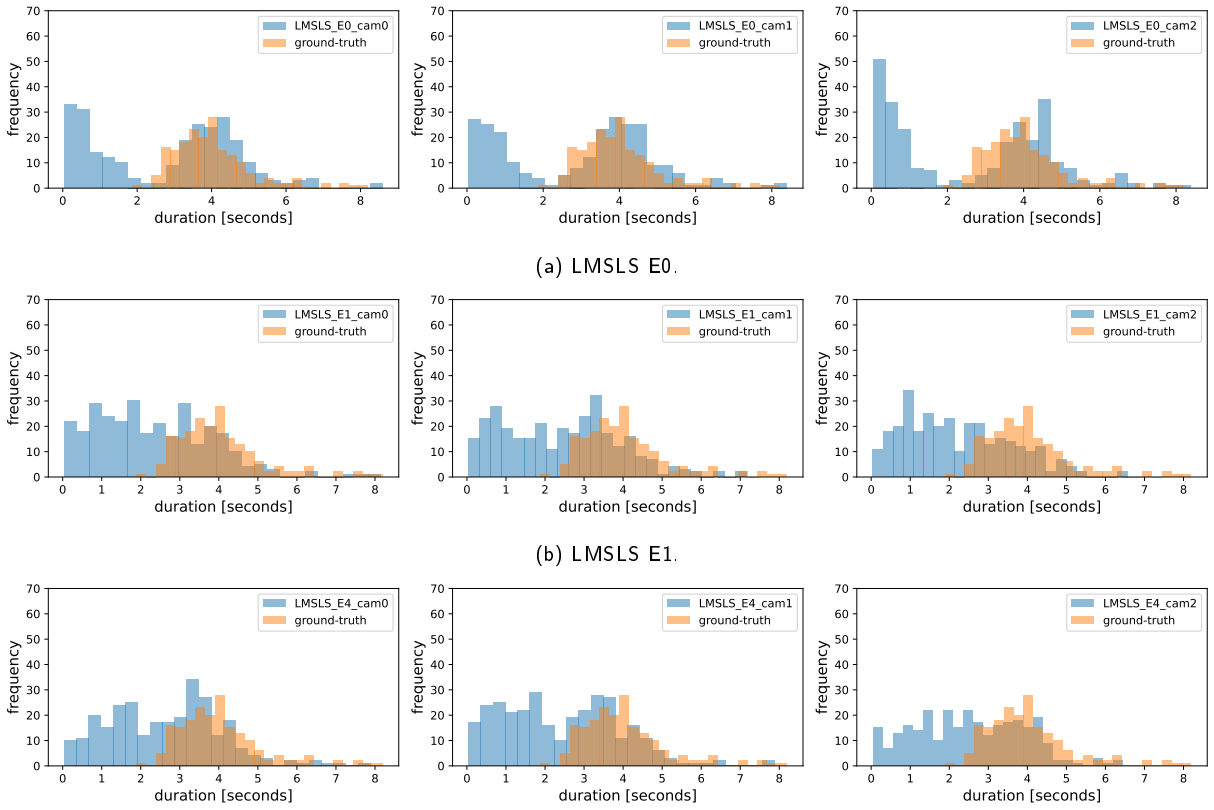
(c) ASSLiSU "hands".

(d) ASSLiSU "face".

(e) ASSLiSU "face+body".

(f) ASSLiSU "body+hands".

Figure 9: Histograms of the detected segment duration for the variants of ASSLiSU considered in LSU-DS phrases for each POV.



(a) LMSLS E0.

(b) LMSLS E1.

(c) LMSLS E4.

Figure 10: Histograms of duration for the variants of LMSLS evaluated for the LSU-DS phrases for each POV.

In Table 2, without considering ASSLiSU "hands", all the hands-including strategies show worse results for all metrics except for the NEO one. This is counterintuitive, but we must stress that we are segmenting whole phrases, and the small movements of the hands can be more prone to noise than the more significant movements of the body. Finally, for LMSLS, Figure 10 shows that camera 2 accumulates the greatest number of short segments, particularly for E0. This method does not show a limitation in the most prolonged duration of the segments, as observed previously for the ASSLiSU "hands" strategy.

Observing the results of ASSLiSU presented in Tables 1 and 2, and considering Tables 1 and 2 of the original paper of Bull et al. [4], it can be observed a correspondence between the reported method best flavors and the best ones observed in the experiments of this study, i.e., "full", "body" and "face+body", depending on the considered metric. However, for the LMSLS method, it can be noticed that the here-found best flavor of LMSLS is not the same as the best model flavor reported in the original paper of Moryossef et al. [12]. In Table 1 of the original work, the best method among those evaluated by the authors was E1, but for the experiments carried out in this work, the best method was E0.

To study whether there is an overfitting of the LMSLS method to the original training data, we conducted an experiment varying the thresholds $(t_b; t_o)$ used during the segment decoding process as explained in Section 2.2.4. Following [12], the variation range of both thresholds was between 10% and 90% in uniform steps of 10%. Figure 11 shows six heatmaps of IoU and Seg metrics for the models LMSLS E0, E1, and E4, considering LSU-DS phrases and cam0 POV. Remarkably, the results for cam1 and cam2 POVs follow the same general patterns. Black stars ★ and red triangles ▲ respectively signal the best and worst threshold combinations based on each metric. The default values of the thresholds were indicated by green square markers ■. Figure 11 shows that

E1 and E4 have a more stable behavior w.r.t. the threshold values. The IoU metric heatmaps (top row of Figure 11) show that the best threshold combinations always present a t_o value equal to 90%, the default value. However, the best IoU metric was obtained with values of t_b equal to 80, 40, and 60% for E0, E1, and E4, respectively. Even though none of these combinations matches the combination of $(t_b; t_o)$ used by default, it can be noted that the variation in terms of the IoU metric is not very large, particularly for the E1 and E4 strategies. On the other hand, analyzing the Seg metric heatmaps (bottom row of Figure 11), it can be noted that the Seg value is almost independent of the t_o value. Once again, the E1 and E4 strategies show a more stable behavior w.r.t. the E0 strategy when t_b value varies. To better see the performance gap, Table 3 shows the IoU and Seg values obtained for the default threshold combination along with the IoU and Seg values associated with the best and worst combinations for each of the LMSLS strategies.

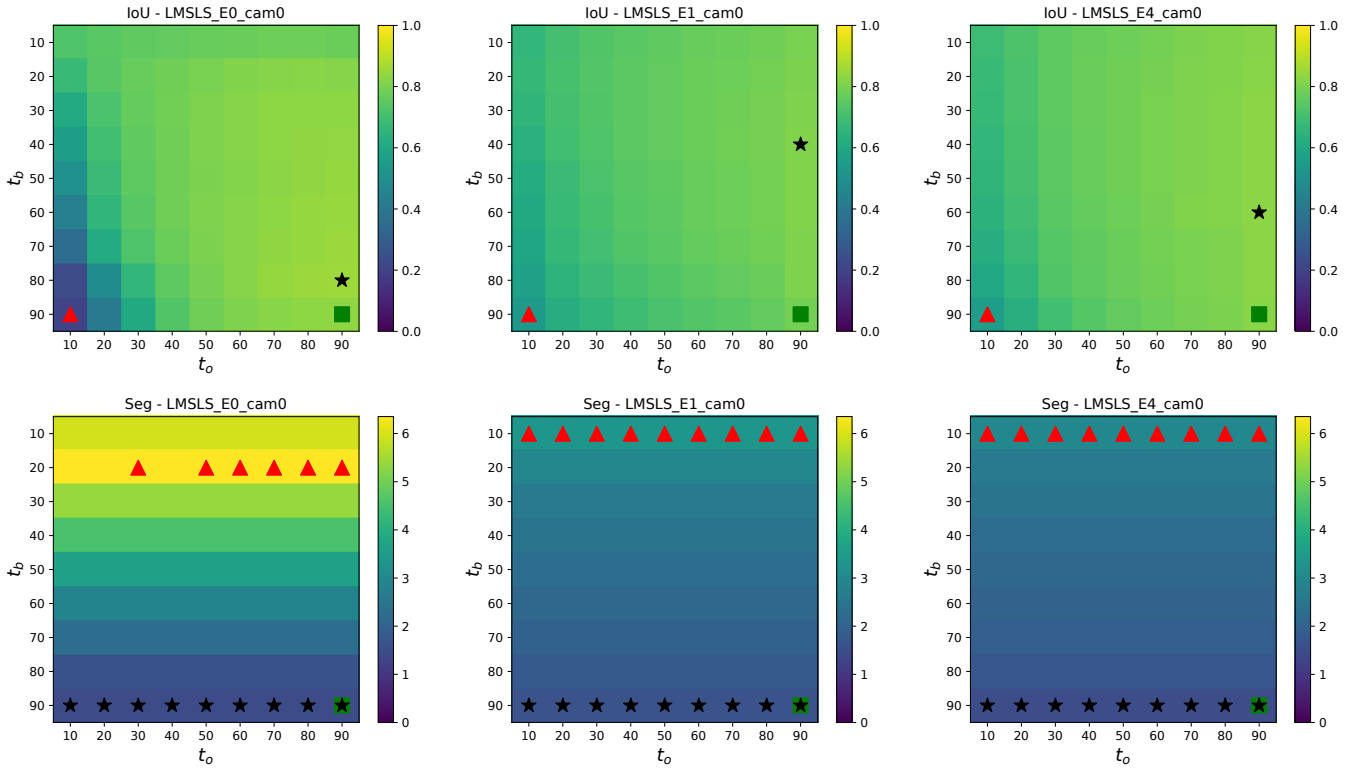


Figure 11: Heatmaps of IoU and Seg metrics for the models LMSLS E0, E1, and E4, considering LSU-DS phrases (only for cam0-POV videos).

	IoU ↑			Seg$^+$		
	default ■	best ★	worst ▲	default ■	best ★	worst ▲
LMSLS E0	0.832	0.854	0.205	1.47	1.47	6.35
LMSLS E1	0.792	0.811	0.542	1.62	1.62	3.33
LMSLS E4	0.823	0.827	0.558	1.50	1.50	2.97

Table 3: Metrics for different LMSLS strategies for the default, best, and worst combination of the pair $(t_b; t_o)$. IoU and Seg are reported as average values among all the LSU-DS phrases (only for cam0-POV videos).

4.5 Experiment 3: Sign Segmentation in LSU-DS

Table 4 shows the LMSLS's different strategies performance for the segmentation task at sign level on the LSU-DS videos. ASSLiSU was not considered in this experiment because it did not segment at the sign level. Only one video in the E0 and E4 strategies produced no segments (NEO $\neq 0$). In this respect, all approaches seem to work fine. Strategy E0 is the one that shows the best performance in the sense of IoU metrics. Concerning the Seg metric, strategy E0 underestimates the number of signs in each video, and the E1 and E4 strategies tend to overestimate that number. The sign segmentation problem is more fine-grained and complex than the phrase segmentation problem. This intuitive observation agrees with the empirical results; the IoU performance drops approximately 20% when the same algorithm switches from phrase to sign segmentation task.

	IoU ↑				Seg^{+}			NEO (%) ↓		
	cam0	cam1	cam2	$\bar{\sigma}_{POV}$ ↓	cam0	cam1	cam2	cam0	cam1	cam2
LMSLS E0	0.51 ± 0.15	0.48 ± 0.15	0.48 ± 0.14	0.045	0.85 (1.75)	0.92 (2.25)	0.85 (1.8)	0	0	0.5
LMSLS E1	0.47 ± 0.13	0.45 ± 0.13	0.41 ± 0.13	0.051	1.22 (2.4)	1.27 (2.4)	1.22 (2.5)	0	0	0
LMSLS E4	0.47 ± 0.12	0.42 ± 0.12	0.41 ± 0.14	0.052	1.19 (3.25)	1.27 (2.25)	1.2 (2.5)	0.5	0	0

Table 4: Metrics for different POV–method–model combinations for sign segmentation on LSU-DS videos (best values , second-best values). IoU is reported as mean±standard deviation, except for the column $\bar{\sigma}_{POV}$ (see text for an explanation). Seg is reported as the mean among the considered videos, and its maximum value is between parenthesis. $^{+}$Seg is better the closer the value is to 1. NEO is reported in percentage.

5 Discussion and Conclusions

This article compares two state-of-the-art methods for automatically segmenting sign language phrases. From a linguistic point of view, segmenting sign language phrases is a problem that can be solved according to different criteria, such as pauses and epenthesis [2, 9]. On the one hand, pauses imply the absence of signer's movement. On the other hand, epenthesis can be defined as interpolation movements that occur between two signs with different places of articulation, for example, the gesture of starting from a resting position towards the beginning of the first sign of a phrase or the gesture from the end of the last sign of a phrase towards a resting position. Although this movement has no meaning from a linguistic point of view, it is an important visual cue to solve the segmentation task. With this work we aim to contribute by making available to the community reproducible, transparent, and accessible sign language segmentation tools. Each method has an online demo that can be used to run the two studied methods on example videos or on new short videos that can be uploaded to the website. Both approaches rely on pose detection to solve the task. ASSLiSU uses OpenPose, while LMSLS uses MediaPipe.

Through the experiments in this article, we propose and discuss a methodology for evaluating this type of methods. The first experiment was conducted over a subset of the How2Sign dataset, a state-of-the-art benchmark of annotated American Sign Language phrases. This dataset has limitations for evaluating segmentation tasks. One of the identified main problems is the significant number of videos with only partial or empty annotations. To alleviate this limitation, we apply an annotation coverage criterion to filter out the How2Sign videos with a lack of annotations greater than a preset threshold. Except for the LMSLS E0 model, the results of Experiment 1 show that all the explored strategies tend to generate a larger amount of segments w.r.t. the ground truth annotations.

To overcome data annotation uncertainties, we decided to include the LSU-DS dataset [16], which contains curated and complete temporal annotations but is limited to single phrases. Experiments 2 and 3 were carried out in this more controlled scenario to evaluate sign language segmentation in

phrases and signs. The results of Experiment 2 show that the evaluated state-of-the-art methods present limitations when solving the phrases segmentation task. The duration histograms show that all the strategies tend to generate shorter segments than the ground-truth ones, with different degrees of error depending on the strategy used and the camera POV. Considering the metrics used for evaluation, it was observed that the best strategies of the ASSLiSU method match the best strategies reported in the original paper, i.e. "full," "body," and "face+body.". However, for the LMSLS method, inconsistencies were observed: while in the original paper, the best-reported strategy was E1, here, the best was E0. However, the results show that LMSLS E1 has a more stable behavior w.r.t. the threshold values than the observed for LMSLS E0. In the context of multiple and simultaneous video acquisitions (as the LSU-DS video triplets) we proposed a metric called IoU dispersion among POV. This metric suggests that LMSLS strategies show more consistent results among the three involved cameras than the ASSLiSU ones. Even when the results are not far in the sense of the considered metrics, after analyzing the experiments, we can conclude that neither method is good enough to solve the task without any fine-tuning or post-processing.

Observing the performance results of Experiment 3, we can say that sign-automatic segmentation is a more complex problem than phrase segmentation. This experiment only included the strategies of the LMSLS method and was carried out as an additional study. Once again, regarding the reported metrics, the E0 strategy performed better than the E1 and E4 ones. Following a similar scheme, a performance analysis could be performed based on threshold variation for sign segmentation in the future. This paper was centered on methods for phrase automatic segmentation, so the latter analysis is out of the scope of this work.

The experiments presented here showed some limitations of the most used segmentation metrics to capture the actual performance of a given method, mainly because their scope is global and they fail to capture the performance at a segment level. To do complementary studies in the future, it can be useful to consider the additional metrics presented in [13] that include omitted segments, pre and post-signing, bridged segments, and detection of non-existing segments, among others.

6 Acknowledgements

The authors of this review would like to thank the authors of the originals, who published their methods for carrying out this study. This publication was made possible thanks to funding from IFUMI, i.e., the Franco-Uruguayan Institute of Mathematics and Interactions.

7 Demo Blobs Credits

LSU-DS, Lingusitic Task 3 [16], downloaded from https://iie.fing.edu.uy/proyectos/lsu-ds/.

References

[1] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, C. Vogler, and M. Ringel Morris, *Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective*, in International ACM SIGACCESS Conference on Computers and Accessibility, ACM, 2019, pp. 16–31, https://doi.org/10.1145/3308561.3353774.

[2] D. Brentari, *A Prosodic Model of Sign Language Phonology*, A Bradford book, MIT Press, 1998.

[3] H. Bull, A. Braffort, and M. Gouiffès, *MEDIAPI-SKEL - A 2D-Skeleton Video Database of French Sign Language With Aligned French Subtitles*, in Language Resources and Evaluation Conference, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, eds., 2020, pp. 6063–6068. ISBN 979-10-95546-34-4.

[4] H. Bull, M. Gouiffès, and A. Braffort, *Automatic Segmentation of Sign Language Into Subtitle-Units*, in European Conference on Computer Vision (ECCV) Workshops, A. Bartoli and A. Fusiello, eds., vol. 12536, 2020, pp. 186–198, https://doi.org/10.1007/978-3-030-66096-3_14.

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, *OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 43 (2019), pp. 172–186, https://doi.org/10.1109/TPAMI.2019.2929257.

[6] S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici, *Gesture Recognition Using Skeleton Data with Weighted Dynamic Time Warping.*, in International Conference on Computer Vision Theory and Applications (VISAPP), 2013, pp. 620–625.

[7] M. De Coster, D. Shterionov, M. Van Herreweghe, and J. Dambre, *Machine Translation from Signed to Spoken Languages: State of the Art and Challenges*, Universal Access in the Information Society, (2023), https://doi.org/10.1007/s10209-023-00992-1.

[8] A. Duarte, S. Palaskar, L. Ventura, D. Ghadiyaram, K. DeHaan, F. Metze, J. Torres, and X. Giro-i Nieto, *How2sign: a Large-Scale Multimodal Dataset for Continuous American Sign Language*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2735–2744, https://doi.org/10.1109/CVPR46437.2021.00276.

[9] F. Grosjean and H. Lane, *Pauses and Syntax in American Sign Language*, Cognition, 5 (1977), pp. 101–117, https://doi.org/10.1016/0010-0277(77)90006-3.

[10] T. Hanke, M. Schulder, R. Konrad, and E. Jahn, *Extending the Public DGS Corpus in Size and Depth*, in Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives, European Language Resources Association (ELRA), 2020, pp. 75–82. ISBN 979-10-95546-54-2.

[11] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, et al., *Mediapipe: A Framework for Perceiving and Processing Reality*, in Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR), 2019.

[12] A. Moryossef, Z. Jiang, M. Müller, S. Ebling, and Y. Goldberg, *Linguistically Motivated Sign Language Segmentation*, in Findings of the Association for Computational Linguistics (EMNLP), 2023, pp. 12703–12724, https://doi.org/10.18653/v1/2023.findings-emnlp.846.

[13] A. Moryossef, I. Tsochantaridis, R. Aharoni, S. Ebling, and S. Narayanan, *Real-Time Sign Language Detection Using Human Pose Estimation*, in European Conference on Computer Vision (ECCV) Workshops, 2020, pp. 237–248, https://doi.org/10.1007/978-3-030-66096-3_17.

[14] C. Neidle, A. Thangali, and S. Sclaroff, *Challenges in Development of the American Sign Language Lexicon Video Dataset (Asllvd) Corpus*, in Workshop on the Representation and Processing of Sign Languages: Interactions Between Corpus and Lexicon, LREC, 2012.

[15] L. Ramshaw and M. Marcus, *Text Chunking Using Transformation-Based Learning*, in Workshop on Very Large Corpora, 1995. https://aclanthology.org/W95-0107.

[16] A. Stassi, M. Tancredi, R. Aguirre, A. Gómez, B. Carballido, A. Méndez, S. Beheregaray, A. Fojo, V. Koleszar, and G. Randall, *LSU-DS: An Uruguayan Sign Language Public Dataset for Automatic Recognition*, in International Conference on Pattern Recognition Applications and Methods, 2022, pp. 697–705, https://doi.org/10.5220/0010894200003122.

[17] S. Yan, Y. Xiong, and D. Lin, *Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition*, in AAAI Conference on Artificial Intelligence, vol. 32, 2018.