VISSE

Visualizando la SignoEscritura

VisSE Corpus of Spanish SignWriting: Annotation guide

Antonio F. G. Sevilla

<afgs@ucm.es>

March 8, 2022

Corpus Contributors

Antonio F. G. Sevilla

José María Lahoz-Bengoechea

Alberto Díaz Esteban

David Pascual García

The canonical URL for the VisSE corpus is
`https://doi.org/10.5281/zenodo.6337885`

Collection and annotation of this corpus was possible thanks to the program for funding of research projects on Accesible Technologies financed by Indra and Fundación Universia.

Contents

History of the corpus

Version 1 (2022-01-18)

Version 1 of the corpus was released as a final product of the VisSE (Visualizing SignWriting) project[1]. It is the version used by many of the project's results and artifacts, and follows most of the decisions described in this document.

Version 2 (2022-03-08)

Version 2 of the corpus is the first release really expected to be used by the research community, and special attention has been put to making annotations consistent and weeding out errors. It features the following improvements:

- Upgraded to `quevedo` dataset version 2.
- Consistent destructuring of graphical components in movement depictions (No double `ARRO`ws, forearm twists and shakes, ...).
- Renamed logogram subsets to a more consistent schema.
- Added fully annotated subsets `'A1_2'`, `'A1_3'` and `'A2'`, which augments the number of logograms by 742, containing more than 4000 new graphemes.
- Split bisyllabic logograms into two different logograms.

[1] `https://www.ucm.es/visse`

1 Introduction

SignWriting is a writing system for sign languages[1]. It uses the graphical possibilities of the bidimensional page to encode the visual characteristics of movement and space used in sign languages, so it is very different to other writing systems, especially in its non-linear nature.

The VisSE corpus is a collection of SignWriting instances, annotated graphically and semantically to capture all the meaning, both conventional and visual, of SignWriting. The samples are all handwritten, and codify signs or parts of signs from Spanish Sign Language. The samples were collected by Dr. José María Lahoz-Bengoechea during a span of years while learning Spanish Sign Language at Universidad Complutense de Madrid, and cover a wide range of vocabulary. However, since they were originally a tool for his private study and not for research, there may be minor errors and inconsistencies in the transcription. They should not be viewed as a collection of Spanish Sign Language Signs, but rather as a collection of SignWriting examples. Nonetheless, a tentative "gloss" (in Spanish) is included for most signs.

Due to the special nature of SignWriting, a tailored annotation schema is needed for its proper codification, and this is reflected both in the digital format of the corpus and in the logical structure of the annotations. This document is a guide for the later as well as a full and normative specification of it. The logical structure and possible values for the annotations are given, as well as some motivation or explanations when needed. For more detail on this, please see our forthcoming article "Building the VisSE corpus of Spanish SignWriting".

[1] Valerie Sutton and Adam Frost (2008). *SignWriting: sign languages are written languages!* Center for Sutton Movement Writing.

It is not within the scope of this document to explain SignWriting. Interested readers can see the excellent and extensive documentation available online at `https://signwriting.org/`.

1.1 About examples

In this document, glyphs from the official Sutton SignWriting digital fonts are used to present examples for the different graphemes. This is done for two main reasons. On one hand, it is useful to present a "standard" and abstract example in each case and not choose any actual handwritten grapheme from the corpus as a "better" example. On the other hand, it may help users knowledgeable of SignWriting understand the schema even if some of the handwriting in the corpus is inconsistent or non standard.

It does not mean that only graphemes identical to the digital glyph will be tagged as such. SignWriting is very complicated, and there are variations available for many of its symbols. Additionally, handwriting tends to be less rigid than digital fonts, so variation is to be expected. Finally, some graphemes which are different symbols in the SignWriting specification have been tagged with the same label in this corpus. This is a conscious decision based on the phonology and the meaning of the graphemes in use in Spanish Sign Language, rather than the accurate and detailed phonetic description that SignWriting provides.

1.2 Annotation schema

SignWriting transcriptions are arrangements of symbols in a 2D space that represent the configuration of the hands, their orientation and movement, and other relevant features of signs in Sign Language. Each transcription can represent a single sign, or part of it. We call each independent arrangement of symbols a "logogram", and we call each of the different symbols or graphical components a "grapheme".

Graphemes have internal structure, both in their graphical properties (strokes, fill) and in their presentation (rotation, reflection). We encode these properties in a set of tags for each grapheme, a mapping of feature names to feature values that stores their meaning 'in isolation'.

However, the meaning of each grapheme is not only determined by its graphical properties, but also by its position relative to the other graphemes in the logogram. This is codified for each grapheme with a 'bounding box', a square region within the logogram within which the grapheme can be found.

1.2.1 Grapheme tags

Not all classes of grapheme require the same number of features to annotate them. For this reason, graphemes are first roughly classified into six groups: `'HEAD'`, `'DIAC'`, `'HAND'`, `'ARRO'`, `'STEM'` and `'ARC'`. This is stored in the `'CLASS'` feature, common to all graphemes in the corpus. A more detailed classification of graphemes is then stored in the `'SHAPE'` feature, which is also common to all graphemes. The `'CLASS'+'SHAPE'` combination establishes the full 'lexical' meaning of the grapheme, but some grapheme `CLASS`es can have additional `'VAR'`, `'ROT'` and `'REF'` features which encode the rest of its 'spatial' meaning.

Each different `'CLASS'` in the corups has a section in this guide, explaining the possible values of `'SHAPE'`, as well as any further tags needed to annotate them. In the case of `HAND`s, due to their complexity, a full chapter is dedicated to their annotation.

1.2.2 Bounding boxes

The location of each grapheme within the logogram is stored alongside its tags in the 'bounding box' attribute. This is a 4-tuple of floating point numbers, in the format (cx, cy, w, h). (cx, cy) are the coordinates of the center of the box relative to the logogram. The coordinates of the logogram go from 0 to 1, $(0, 0)$ being the top left corner, and $(1, 1)$ the bottom right one. (w, h) are the width and height of the grapheme region, again relative to the width and height of the logogram (so ranging from 0 to 1).

To visualize and create bounding boxes, a graphical tool is needed. Quevedo's web interface provides such a tool, and shows the boxes as in figure 1.1. Boxes should cover the full graphical extent of the grapheme, preferably with some padding around it. The exact location of the boxes is not important, but rather the general relative colocation of graphemes, as well as the full area covered by the grapheme's symbol.

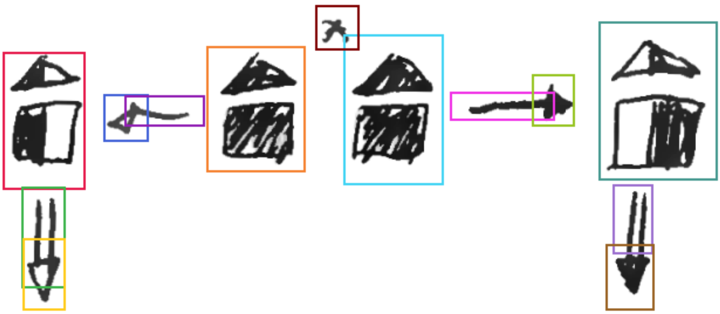


Figure 1.1: Graphical example of general bounding box annotation.

1.3 Programmatic access

This corpus is formatted as a Quevedo[2] dataset, so the easier way to access the annotations is using the Quevedo library or command line tool. Since annotations have an important graphical component, Quevedo's web interface can be especially useful in their visualization. To install Quevedo with the web interface, the command '`pip`

[2] Antonio F. G Sevilla, Alberto Díaz Esteban, and José María Lahoz-Bengoechea (2022). "Quevedo: Annotation and Processing of Graphical Languages".

`install quevedo[web]`' can be used in any system with python and pip installed.

Nonetheless, the annotations are stored in an open and standard format, so they can also be manually inspected or consumed by other tools.

1.3.1 Format on disk

Logograms in the corpus are stored in the `logograms` directory. They are split into subsets according to when they were collected, but subset structure is more organizational than semantically relevant. In each subset directory, `logograms/A1_1` for example, logograms are sequentially numbered starting from the number 1.

Each logogram instance in the corpus consists of two files. First, the source image, with file extension `.png`, and then the annotation itself, with the same name but extension `.json`. For example, the annotation data corresponding to image `logograms/A1_1/1.png` is in file `logograms/A1_1/1.json`.

The `json` annotation file is a dictionary of attributes. It has a `'graphemes'` key, an array of the different graphemes found in the logogram, each of them having a `'box'` key with the bounding box coordinates (an array) and a `'tags'` key with a mapping from tag names to tag values.

1.3.2 Other corpus objects

Inside the corpus root directory there are a number of other directories not mentioned above. The `'graphemes'` directory stores isolated graphemes samples. There are currently no such samples in the corpus, but they can be automatically extracted from the logograms. The `'networks'` directory stores the weights for neural networks trained with the corpus data, and useful code for processing the dataset can be found in the `'scripts'` directory. For more information on the dataset structure, please refer to Quevedo's documentation at `https://agarsev.github.io/quevedo`. For information on

```
{ "graphemes": [
    {
      "tags": {
        "CLASS": "HAND", "SHAPE": "PICAM-",
        "VAR": "hb", "ROT": "N", "REF": "n"
      },
      "box": [ 0.3160, 0.4671, 0.3218, 0.6927 ],
      ...
    },
    {
      "tags": {
        "CLASS": "ARRO", "SHAPE": "b", "ROT": "N"
      },
      "box": [ 0.5404, 0.1947, 0.1441, 0.1434 ],
      ...
    },
    ...
  ],
  "meta": {
    "gloss": "Carrera universitaria",
    ...
  },
... }
```

Listing 1: Example json annotation file.

the machine learning artifacts, please refer to our forthcoming article[3].

The rest of this document covers the annotation schema for the different grapheme classes.

[3] Antonio F. G Sevilla, Alberto Díaz Esteban, and José María Lahoz-Bengoechea (2021). "Automatic SignWriting Recognition".

2 Invariant Graphemes

Some graphemes of SignWriting are always represented upright and in the same orientation, namely `'HEAD'` and `'DIAC'` graphemes. In the case of `'HEAD'`, the body is normally assumed to be upright while signing, so it is drawn as such. When there are alterations to this principle, such as head movement, they are indicated with additional symbols rather than by transforming the grapheme. In the case of `'DIAC'`, these are abstract symbols with no internal spatial information, so are always represented with a constant, invariant shape (see their section for a caveat).

Therefore, `'HEAD'` and `'DIAC'` graphemes only use the `'CLASS'` and `'SHAPE'` tags in the corpus tag schema. The possible `SHAPE`s are enumerated in the following.

2.1 HEAD

`'HEAD'` graphemes represent the location of the sign relative to a number of different bodily locations in the head, and at the same time depict some of the non-manual parameters of the sign. In this corpus, this articulation is only annotated for the mouth. There rarely appear logograms with other marks, such as head or eye inclination, but it has not been annotated (for now).

In `'HEAD'` graphemes, the `'SHAPE'` tag specifies the holistic meaning of the full grapheme, including place of articulation and mouth gesture. Since `'HEAD'` graphemes are invariant, no other tags are used.

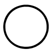
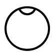
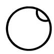
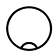
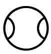
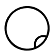
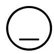
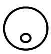
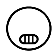
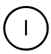
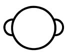
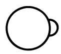
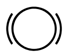
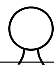
`face`		`fore`		`forer`		`chin`	
`cheeks`		`cheekr`		`cheekl`		`mouth`	
`moutho`		`smile`		`teeth`		`tongue`	
`nose`		`ears`		`earr`		`eyes`	
`eyer`		`hair`		`back`		`neck`	

Table 2.1: Values of `'SHAPE'` for `'HEAD'` graphemes.

2.2 DIAC

'DIAC' graphemes are small, invariant marks, such as contact or dynamics marks or internal movements of the hand. They are classified as such due to their graphical characteristics, rather than after a thorough examination of whether they count as diacritics or not, or due to some internal semantic coherence of the class.

Their concrete meaning is codified in their 'SHAPE' tag. While some are graphical transformations of each other, their rotation and reflection is not productive so 'ROT' or 'REF' tags are not used.

However, sometimes DIACs can appear rotated or reflected for stylistic reasons. This does not alter meaning, but it is important to distinguish between DIACs that are mirror one of the other, with different meanings, and DIACs that accept some stylistic variation to better convey location or what other graphemes they modify.

touch	∗	inter	|∗|	brush	⊙
grasp	+	between	|+|	rub	@
flex_hook	●	flex_base	∨	flex_alt	⩔
ext_hook	∘	ext_base	∧	ext_alt	⩓
strike	#	tense	~	wiggle	⩕
sym	◡	anti	◡◡	altern	ω
fast	∠				

Table 2.2: Values of 'SHAPE' for 'DIAC' graphemes.

3 Hands (HAND)

Hands are the most prominent articulators of Sign Language, and have many degrees of freedom and articulatory possibilities. They are represented in SignWriting with complex graphemes which encode in their graphical attributes the different features of the hand. In the corpus they are assigned the `CLASS=HAND`, and all four additional tags are used: `'SHAPE'`, `'VAR'`, `'ROT'` and `'REF'`.

3.1 SHAPE

The first feature of the hand is its "shape" or configuration: how the fingers are bent and placed to form a unique shape that acts as a unit. Graphically, the `SHAPE` tag is roughly the outline of the grapheme, mainly the strokes representing the fingers. Fingers are very flexible, so there are a great many possible configurations that the hand can adopt, and SignWriting strives to provide symbols for every one of them. However, not all of them are in use in every sign language. There is also allophonic variation, meaning some different finger configurations are perceived to be the same hand shape for native signers, and so there can be vacillation and inconsistencies in their transcription.

In this corpus, hand shapes are labelled according to the phonology of Spanish Sign Language, not phonetically, so some different symbols are tagged with the same `SHAPE`. The phonological base for our labeling is to be published in a forthcoming article. Since there is not a standard notation for hand shapes across languages, we use our own ASCII-based notation which is derived from the previously mentioned phonology. For users not interested in the underlying linguistic theory, these labels can be assumed to be arbitrary strings uniquely identifying the different configurations. Currently, 72 different hand `SHAPE`s can be found in the corpus.

3.2 VAR

Apart from finger configuration, hands can rotate in the three dimensions of space, which complicates their transcription in the flat page. SignWriting uses a combination of graphical features to represent hand orientation, encoded in the remaining tags for hands in this corpus.

The first of them is the `VAR`iation. Graphically, it encodes the "alteration" of the basic shapes encoded in the previous label. This variation can happen in two ways. First, the body of the hand can be filled with different patterns of black and white. White represents the palm, and black represents the back of the hand, as viewed from the point of view of the signer. The fingers can also be detached, meaning the orientation is horizontal.

Fill variation is encoded with the letters `'w'` (white), `'b'` (black), and `'h'` (half). Finger detachment is encoded by prepending the letter `'h'` (horizontal) to the tag. This gives six possible `VAR` tags:

w 	b 	h 
hw 	hb 	hh 

Table 3.1: Values for the `'VAR'` tag.

There is also the possibility that a hand grapheme has a "black left" and "white right" fill pattern. This is encoded as `'h'` or `'hh'`, and treated as a graphical reflection (see below for the `'REF'` tag).

3.3 ROT

To complete the graphical representation of hand orientation, `HAND` graphemes can also be rotated around their center. This rotation is not continuous but rather has 8 possible values, encoded in this corpus using the notation for the cardinal directions. The hand is considered to be pointing along is distal axis, that is, the straight line from the forearm to the fingertips when they are fully extended.

N		NE		E		SE	
NW		W		SW		S	
N		NE		E		SE	
NW		W		SW		S	

Table 3.2: Values for the `'ROT'` tag in `HAND`s.

3.4 REF

As a last transformation, `HAND` graphemes can appear "mirrored" in SignWriting. Mirroring of a grapheme is not reflective of any one phonological feature, but rather a graphical attribute that can be used to convey different meanings. For example, right and left hands are mirror images, so the corresponding graphemes can be mirrored to better identify each of them. "Black" `VAR`iants are also often mirrored, to better iconically depict the hand as it would be seen by the signer.

Therefore, the meaning of reflecting a grapheme has to be extracted from the context, and can not be deduced from the isolated grapheme at all. This also means that there is not a phonological criterium to decide on a "normal" form of a grapheme, so the criteria chosen in this corpus may seem arbitrary. However, they are chosen to maximize graphical homogeneity and predictability, which can help in the computational treatment of SignWriting.

Reflection is codified in the `'REF'` tag, which can take the values `'n'` (not reflected) or `'y'` ("yes", reflected). To decide whether an instance is reflected, the following algorithm is used:

1. Always, reflection must be decided from 'North' rotation. If a grapheme is rotated, it must first be (mentally) set upright.

2. If the `'VAR'` is `'h'` or `'hh'`, the variant with the black on the right is `'n'`, and the one with black on the left is `'y'`.

3. If the `'VAR'` is `'w'` or `'b'`, attention must be paid to the fingers. If they are in the same position as the unreflected `'h'` VARiant, then they are themselves not reflected. In other words, the `'h'` VARiant decides, and the `'b'` and `'w'` ones copy it.

4. If the `'w'` o `'b'` VARiant is not identical to the `'h'` one, attention is paid to the flexion of the fingers. If they bend to the left, the `'REF'` is `'n'`, otherwise it is `'y'`. In the case of the single little finger SHAPE, where the finger bends to one side but curls to the other, the not reflected grapheme is that where the finger is to the far left (white right hand). This step can also help decide the `'REF'` for other `'w'` or `'b'` graphemes without having to look up the `'h'` version.

5. Horizontal VARiants follow the same pattern as vertical ones.

The algorithm above is also important because when dealing with handwritten SignWriting, such as this corpus does, there can often appear "non-normative" uses of graphemes which are however understandable and need to be annotated. In any case, since pictures convey graphical information better than words, the following tables present some examples of `'REF'` tags.

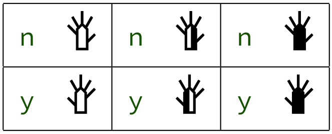


Table 3.3: `REF` in relation to `VAR`

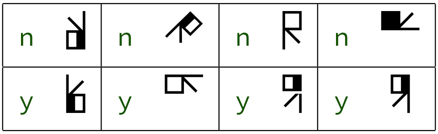


Table 3.4: `REF` in relation to `ROT`

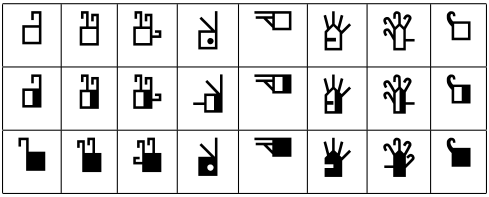


Table 3.5: `REF=n` for some flexed `SHAPE`s

3.5 Ambiguous graphemes

Sometimes grapheme `SHAPE`s are symmetric, meaning that the `REF`lected versions end up being graphically identical. In this case, `'REF'` is always taken to be `'n'`. In a few cases, rotation can also be ambiguous (for example the closed fist, which is a square). In this case, the first possible `'ROT'` in this sequence is chosen:

`'N'`→`'NE'`→`'E'`→`'SE'`...

4 Movements

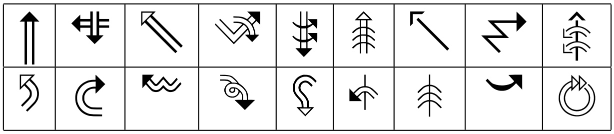
Hand movements are an integral part of sign language, and therefore a substantial part of SignWriting. They are codified with paths and arrows that iconically depict the 3-D movements of the hands in the page. To properly encode 3-D space in 2-D writing, they use graphical attributes to distinguish between planes of movement.

In the digital typographies of SignWriting, there are tens of thousands of characters to account for a wide variety of possible trajectories and types of movement. When devising an annotation schema for movement, and especially when dealing with handwritten SignWriting, using a flat classification is unpractical.

Therefore, in this corpus, movements are annotated by decomposing them into segments, and annotating the segments. These segments convey either straight (`STEM`) or curved (`ARC`) paths, and end of movement markers or "arrow heads" (`ARRO`). Additionally, since the shoulders and waistline are depicted in SignWriting with straight lines, they are provisionally annotated as `STEM`. The forearm shares many characteristics of `STEM`s, so it too is annotated as such.

Movement segments have directional information, annotated in their `'ROT'` tag, and plane distinctions, annotated in the `'SHAPE'` tag. Arrow heads have directional information as well, annotated in the `'ROT'` tag, but the `'SHAPE'` tag is used to annotate the type of arrow head, which is used in SignWriting to denote what hands move along each path.

Segments often overlap, depicting for example crossing movements of the hands, or a curved segment can be superposed over a straight segment to convey rotation simultaneous to displacement. If these segments have different `CLASS`es, their bounding boxes are annotated as usual. If the `CLASS` is the same, the overlap in bounding boxes can make them meaningless. This is very common with crossed forearms configurations, or with ‘X’-shaped movements. In these cases, straight segments are subdivided, and each division annotated independently, as in figure 4.1.



Small sample of possible movements as encoded in the Sutton SignWriting fonts.

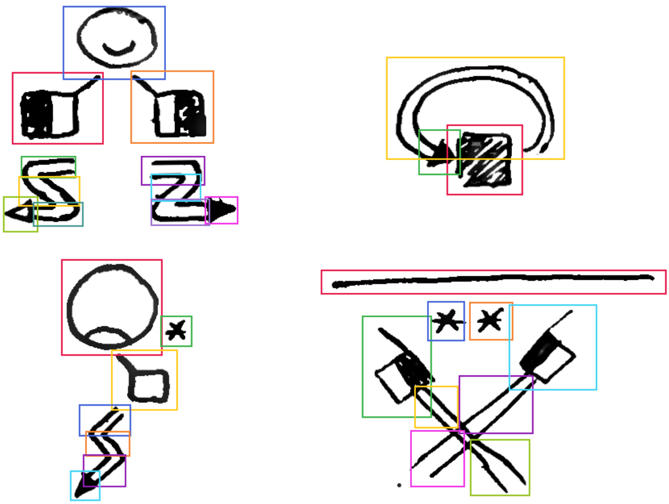


Figure 4.1: Graphical example of bounding box annotation for some complex trajectories.

4.1 ARRO

Arrow heads mark which hand moves, encoded in their `'SHAPE'` tag. They can be `b`lack (right hand), `w`hite (left hand) or `'j'`, for both hands joining in the movement. They also point in a cardinal direction, annotated in their `'ROT'` tag.

`b`	▲	`w`	△	`j`	^

Table 4.1: Values for the `'SHAPE'` tag for `'ARRO'`.

`N`	▲	`NE`	◥	`E`	▶	`SE`	◢
`NW`	◤	`W`	◀	`SW`	◣	`S`	▼

Table 4.2: Values for the `'ROT'` tag in `ARRO`s.

4.2 STEM

Straight lines can represent the shoulders or waistline, straight movements, the forearm, or both the forearm and a movement at the same time. Their direction is marked in the `'ROT'` tag using cardinal directions. Since they are symmetric, only half of the possible `'ROT'` values are used.

To distinguish between vertical and horizontal movements or forearms, `STEM`s can be single or double, which is annotated in the `'SHAPE'` tag. Shoulders and waists are always `s`ingle.

s	d

Table 4.3: Values for the `'SHAPE'` tag for `'STEM'`.

N	NE	E	SE

Table 4.4: Values for the `'ROT'` tag in `STEM`s.

4.3 ARC

Curved paths represent arcing or circular movements.

As in `STEM`s, `s`ingle and `d`ouble paths represent horizontal and vertical planes of movement. This is encoded with the first letter of the `'SHAPE'` tag. The second letter is used to determine the amplitude of the movement, and can take three values: `'q'` for ‘quarter’, a small arc; `'h'` for ‘half’, a bigger arc which covers around half a circle, and `'f'` for ‘full’ for fully circular paths. Distinction between `'q'` and `'h'` can be difficult without underlying understanding of the sign language depicted, but with some practice it becomes more intuitive.

To determine the `ROT`ation of `ARC`s, two points need to be mentally found. The first one is the center of the circle on which the `ARC` lies. The second one is the middle point of the `ARC` segment. ‘Fully’ circular `ARC`s still have a middle points, since they start and end at the arrow head. Once these two points are determined, the cardinal direction to annotate for the `'ROT'` is the direction in which points the segment from the center to the middle point. Table 4.6 will probably make this clearer.

sq 	sh 	sf 
dq 	dh 	df 

Table 4.5: Values for the `'SHAPE'` tag for `'ARC'`.

N 	NE 	E 	SE 
NW 	W 	SW 	S 

Table 4.6: Values for the `'ROT'` tag in `ARC`s.

List of tables and figures

List of Tables

List of Figures