Per Channel Automatic Annotation of Sign Language Motion Capture Data

Lucie Naert, Clément Reverdy, Caroline Larboulette, Sylvie Gibet

Per Channel Automatic Annotation of Sign Language Motion Capture Data

Lucie Naert, Clément Reverdy, Caroline Larboulette, Sylvie Gibet
IRISA Lab., Université Bretagne Sud
Vannes, France
{lucie.naert, clement.reverdy, caroline.larboulette, sylvie.gibet}@univ-ubs.fr
lsf.irisa.fr

Abstract

Manual annotation is an expensive and time consuming task partly due to the high number of linguistic channels that usually compose sign language data. In this paper, we propose to automatize the annotation of sign language motion capture data by processing each channel separately. Motion features (such as distances between joints or facial descriptors) that take advantage of the $3D$ nature of motion capture data and the specificity of the channel are computed in order to (i) segment and (ii) label the sign language data. Two methods of automatic annotation of French Sign Language utterances using similar processes are developed. The first one describes the automatic annotation of thirty-two hand configurations while the second method describes the annotation of facial expressions using a closed vocabulary of seven expressions. Results for the two methods are then presented and discussed.

Keywords: automatic annotation, automatic segmentation, motion capture, French Sign Language, linguistic channels

1. Introduction

Whether we want to linguistically study sign languages, use digital data to identify salient linguistic components, recognize or synthesize continuous signing, an annotation of the data is needed. The annotation of sign language is a two-step process. The first step, called *segmentation*, consists in dividing the stream of data in segments of interest. Those segments are then identified in a second step called *labeling*. This annotation might be done manually but is a fastidious, time consuming and expensive task. Not only does it require the skills of language experts, but it is also subject to inaccuracies and mistakes as the experts may not have exactly the same segmentation criteria. In the particular case of sign language, the annotation process needs to be done by experts in sign language and gesture annotation. In addition, sign languages are expressed simultaneously on multiple channels (manual configuration, wrist orientation, body posture, facial expression, etc.), thus complicating the task of the annotators. When comparing the duration of the annotation process to the duration of the data to annotate, (Dreuw and Ney, 2008) introduces a real-time factor of 100 (i.e. all the manual and non manual features of a 1 minute video of sign language will be annotated in 100 minutes). We propose to automatically annotate each channel separately following the scheme of Figure 1.

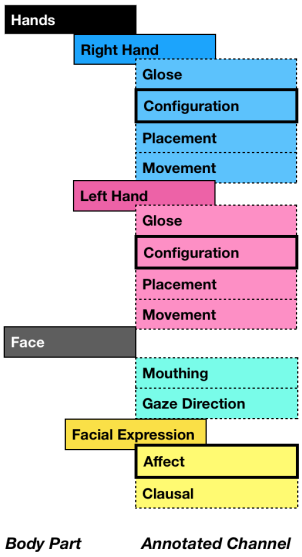


Figure 1: Our intended annotation scheme: currently, we automatically annotate the hand configuration and affect channels (boxes with bold borders).

Automatic annotation of sign languages could reduce the annotation costs but is still a challenging and yet to be solved task. One way to tackle this problem is the *machine learning* approach (e.g. use of Bayesian/statistical models or artificial neural networks) which aims at automatically learning the parameters of a model from a sample of manually annotated data. This model is then used to automatically annotate new data. The corpus intended to train the machine learning model must thus be designed carefully before recording sign language data using either video or motion capture technologies. Despite being easy to use and relatively cheap, video does not preserve the third spatial dimension of motion. Furthermore, the resolution of classic video recordings is rarely high enough to obtain a precise segmentation. Motion capture (MoCap) technologies offer a better precision, both spatially and temporally, and spatial information that is not available with 2D cameras. MoCap data can be used for sign language analysis – using motion descriptors computed from the 3D data, such as distances between joints, velocity or curvature of selected joints – as well as for synthesis by using the captured motions to animate a signing avatar. In this paper, we will address the problem of the automatic annotation of sign language MoCap data, focusing more specifically on two main channels: facial expression, and hand configuration.

Previous work on automatic annotation on video and MoCap data are described in section 2. The specification, capture and manual annotation of the input sign language data

used in this study is introduced in section 3. In section 4, the annotation methodology is presented and illustrated with the examples of the automatic annotation of the hand configurations and the facial expressions in French Sign Language. Section 5 describes the results of the automatic annotation applied to those two channels. Finally, section 6 discusses the perspectives and challenges of our method.

2. Related Work

In this section, previous studies on automatic annotation of sign language are presented. For gesture segmentation in general, a very complete framework is developed in (Lin et al., 2016). It provides a general overview and a comparison of several works in human motion segmentation using different data sources (camera, MoCap, sensors, etc.) but does not address the problem of "per channel segmentation" or the particular application of sign language processing.

Continuous signing segmentation and recognition can be opposed to isolated sign language recognition. Coarticulation effects present in continuous signing and absent from isolated signs make the segmentation of the former harder. Most of the existing work on the automatic segmentation of continuous signing relies on video footage to segment at a gloss-level. (Kim et al., 2002) take advantage of Hidden Markov Model (HMM) to segment a continuous stream in Korean Sign Language into signs segments. Similarly, (Yang and Sarkar, 2006) perform sign segmentation of continuous American Sign Language using Conditional Random Fields (CRF). In their article they demonstrate the superiority of the CRF approach (85% accuracy) compared to HMM (60%). A different approach was developed by (Lefebvre-Albaret et al., 2008). It presents a computer-aided segmentation of sign language sequences based on the detection of motion cues such as symmetry, repetition and hand trajectories templates. The algorithm is helped by the punctual intervention of an operator who has to specify one frame belonging to each sign.
However, those segmentation approaches do not take into account the multichannel aspect of sign languages and lead to segmentation schemes highly dependent on the context of the utterance, i.e. the segments implicitly contain the coarticulation effects of the sequential signs. The resulting segmentation is thus hardly reusable in a different context, for example to synthesize new utterances. A lower-level segmentation, based in particular on phonological elements would facilitate sign composition in various contexts in order to produce new utterances. However, although several studies address the issue of the annotation of sign language video data at a gloss level, little attention has been given to the automatic annotation of the different linguistic channels of sign languages. The work of (Stokoe, 1960) gives a phonological structure to signs by specifying three linguistic parameters to describe all signs: hand motion, hand configuration and hand placement. Each feature can take a discrete value in a limited vocabulary. Two complementary features were later added, hand orientation (Battison, 1978) and non manual features such as shoulder tilt or facial expressions. Many phonological structures use those five features to define signs which can be used as a basis for video annotation : the segments are of a finer level than the gloss segmentation and retain a linguistic value. In an early work, (Vogler and Metaxas, 2001) break signs into "phonemes" (close to the previous five features) and use HMM on the combination of the phonemes to recognize signs. The channels are processed separately but the ultimate purpose is gloss recognition and not channel annotation. In addition, this work is based on the Movement-Hold model which has been later replaced by a more precise phonetic model (Johnson and Liddell, 2011). Furthermore, due to the difficulty of capturing the finger movements, the hand configuration channel was not processed and the authors of the article also chose not to deal with non manual features. More recently, (Dilsizian et al., 2014) propose to add some linguistic knowledge about the composition of lexical signs to considerably improve the performances of the recognition system.

Work on sign language MoCap data is scarcer than on video. For gloss-level segmentation, (Naert et al., 2017) use some kinematic properties of the two wrists that can be extracted from MoCap data. At a finer level, (Héloir et al., 2005) focused on the segmentation of hand configurations using Principal Composant Analysis (PCA) but the work is restricted to fingerspelling segmentation.

The recognition of facial expressions has received increasing attention in recent years, mainly from the computer vision community. Regarding the existing datasets, the availabiliy of 2D recording devices made possible the creation of large data collections, such as the Cohn-Kanade Dataset (posed facial expressions) and its extension (Lucey et al., 2010) (posed + non-posed facial expressions) or the MUG database (Aifanti et al., 2010) (posed + non posed) with many (up to hundreds) actors. High resolution 3D facial databases with expressions have also been created using binocular/structured light cameras (Zhang et al., 2014), (Yin et al., 2008). The frame rate of such optical device is often limited to 60 or less frames per second (usually 20/30 fps) which may be insufficient for those who are interested in dynamic expressive variations. MoCap techniques can capture movements up to 200 fps and more, which makes them much more powerful for analyzing fine expressive variations. Nevertheless, the publicly available facial MoCap databases are still scarce. The multimodal database described in (Busso et al., 2008), which includes facial MoCap with speech and elicited emotional expressions is one of the few existing ones.

This paper presents the automatic annotation of continuous signing in French Sign Language using MoCap data on two linguistic channels : the hand configurations and the facial expressions.

3. Input Data

This section describes the definition and recording of sign language data as well as the specification of the two corpora that have been used for the studies.

3.1. General Considerations

To develop models for automatic annotation, the first solution that might come to mind is to record all the existing signs in all the possible contexts in order to cover exhaustively all the possible cases of sign production. While this can be attempted (with varying degrees of success) for oral languages by, for example, retrieving huge databases from the Internet (e.g. Wikipedia pages, Twitter posts, etc.), this is impossible for sign languages. Indeed, (i) sign language databases are scarce, especially MoCap databases, (ii) sign languages use the 3D space and the temporal dimension which leads to the production of an infinite number of combinations of the different physical channels, and (iii) sign language cannot be limited to their standard, reference signs: many sign language mechanisms such as classifier-predicates are as important as standard signs and depend strongly on the context of the sentence.

Instead of collecting a large set of random data, it might be more relevant to design a corpus specifically suited to the studied problem. One way to reduce the complexity of the capture is to consider each chosen channel separately. Indeed, each channel can display a limited number of different behaviors. For example, we can enumerate a limited number of different hand configurations in sign languages (often less than fifty in French Sign Language). As a consequence, a corpus designed to study and automatically annotate the hand configurations would focus only on a small number of signs to cover all the possible hand configurations.

To sum up, for the application of automatic annotation, a corpus containing many repetitions of a limited number of different occurrences of the studied element will be preferred. The variability induced by a different context in the element production (for example, by capturing the same hand configuration in different signs) or by a different signer, must be recorded in order to improve the generalization capacity of the resulting model.

3.2. Corpora

Two different corpora were used for the presented work: *Sign3D* (Gibet et al., 2015) to annotate hand configurations, and *FEeL*, a novel corpus that is still under development, for the facial expressions.

Characteristics

The *Sign3D* corpus contains eight sequences of motion. Each of these sequences is composed of one to five French Sign Language utterances. The utterances are messages about the opening hours and entrance fees of various town places (swimming pool, museum, etc.), as well as the description of various events (exhibitions, theatre play etc.). The capture was performed on one signer using a *Vicon* MoCap system and an eye-tracking device to follow gaze direction. Facial expressions, body and finger motions were simultaneously recorded during approximately 9 minutes at 100 fps (around 54000 frames in total).

The *FEeL* corpus has been captured using two signers (learner level). Three kinds of sequences - corresponding to different sets of instructions given to the signers - were recorded. We worked exclusively on the affect channel of the face and chose to analyze this channel within the Ekman framework with six categorical classes of basic emotions (i.e. *anger - A*, *disgust - D*, *fear - F*, *joy - J*, *sadness - Sa*, *surprise - Su* and *neutral - N*), which was easier to annotate and more understandable by humans than continuous models (e.g. the *Pleasure - Arousal - Dominance* framework). Three kinds of sequences were recorded:

i) *Isolated Expressions - IE:* the signers were asked to perform a given expression five times, each expression had to be maintained several seconds before returning to neutral (e.g. for joy we have: $N-J-N-J-N-J-N-J-N-J-N$). Six sequences were recorded per signer, one for each class of affect.

ii) *Sequences of Expressions - SE:* the signers were asked to alternate a given expression with each of the five other expressions (e.g. for joy: $N-J-Su-J-A-J-F-J-Sa-J-D-J-N$). Five sequences were recorded per signer.

iii) *Expressive Utterances - EU:* Sign language sentences with emotional content were prepared. The signers were asked to repeat three times each sentence with a given affect (e.g. it was asked to the signer to sign the following sentence with disgust : "There is a spider on my pizza! Yuk!"). 18 sequences were recorded per signer, one for each sentence.

The corpus has been recorded via a *Qualysis* MoCap system. A total of 40 facial markers were tracked at 200 fps.

Manual Annotation

Manual annotations are used as reference and training data for our automatic annotation. It is thus necessary to have a thorough annotation. The *Sign3D* and the *FEeL* corpora have been annotated using the ELAN software (Max Planck Institute for Psycholinguistics, 2017).

The *Sign3D* corpus has been annotated on several channels including, but not restricted to, gloss, hand placement, hand orientation, mouthing, facial expressions and hand configurations. To reduce the error rate and to have a more consistent annotation, two annotators knowledgeable in French Sign Language validated each others' work.

Concerning the annotations of the *FEeL* corpus, we focused our efforts on the affect channel and, so far, a single annotator has been involved in the process. This annotator has been instructed to "subjectively annotate what he saw" with respect to the two following rules: (i) we distinguish two kinds of segments: the *transition* segments where the class vary from a starting expression to an ending expression, and the stable segments where the class doesn't vary along time; (ii) the name of a *transition* segment is the concatenation of the name of the starting class and of the name of the ending class (e.g. *NA* means that the transition come from the *neutral* class to the *anger* class). A stable segment is named according to the maintained expression displayed (e.g. *Sa* stands for *sadness*).

4. Automatic Annotation

This section describes the principal steps to automatically annotate a sign language channel for a given corpus. Motion descriptors are first computed in order to segment and

then label the sign language data. The examples of the annotation of the hand configuration and of the affect channels are detailed.

4.1. Descriptors

The raw data collected from motion capture is the vector of the $3D$ positions of the body markers along time and might not be the best representation to study either the hand configurations or the facial expressions. Indeed, it is often required to transform the initial data in order to get a descriptor that depends only on the phenomenon that we intend to analyze. For instance, if the system is supposed to recognize hand configurations, it should not be sensible to morphological differences between the signers.

Hand Configuration Descriptors

While the positions and orientations of the joints vary according to the chosen reference frame, the Euclidean distances between two articulations are invariant. The hand configurations are therefore described by the vector of the Euclidean distances between each joints of each hand. In our model, each hand has 26 joints (five per finger and one for the wrist) resulting in a total of 325 possible combinations. However, some distances are more relevant than others. For example, distances between two consecutive joints (e.g. the second and third joints of the middle finger) are physiologically similar to bones. Those distances only undergo small variations (due to noise in the data) and are not relevant to discriminate hand configurations.
Different subsets of the total number of distances have been tested in order to find the optimal features. A subset of the 29 most discriminating features was preferred (see. fig. 2). It consists of the distances between:

(1) the wrist and the extremities of the fingers (5 distances) to evaluate the bending of the fingers on the palm,

(2) the extremity of one finger with its neighbors (5 distances) to measure the gap between the fingers,

(3) the extremity of each finger and its corresponding knuckle (5 distances) to evaluate the bending of the fingers with respect to the knuckles, and

(4) the extremity of the thumb and the joints of the other fingers (14 distances) to specify the behaviour of the thumb.

The *Sign3D* corpus that has been used to study the hand configurations contains the data of a unique signer but, in order to have more generic results and to give each distance the same weight, it is necessary to normalize our features. The normalization was performed by dividing each of the 29 types of distances by its maximal value in the corpus. All the distances have therefore a value between 0 and 1. Those distances are then used to segment and label the hand configurations.

Facial Descriptors

A common approach to animate facial expressions of a virtual character is the blendshapes method. An expression

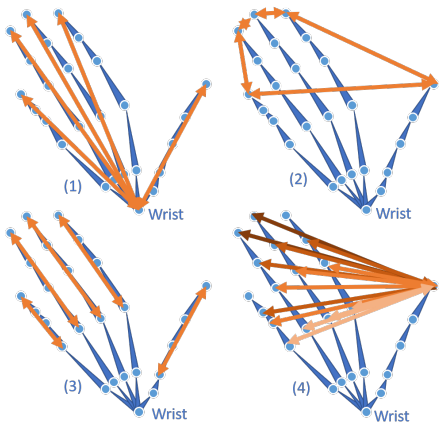


Figure 2: The subset of the 29 distances.

$Expr$ can be expressed as the sum of the mesh B_0 representing the neutral expression and a weighted linear combination of n basic deformations b_i expressed differentially from the neutral expression (see also fig. 3):

$$Expr = B_0 + \sum_{i=1}^{n} w_i \cdot b_i \qquad (1)$$

This method has the advantage of providing a light representation (in our case only 51 basic deformations) which leads to faster computations and facilitates storing in our database. In order to automatically obtain the appropriate set of parameters $\{w_{1..n}\}$ at each frame we have to face two problems: i) the targeted avatar and the signer don't have exactly the same morphology (the *retargeting* problem), ii) for one given expression E there might be multiple existing linear combinations $\sum_{i=1}^{n} w_i \cdot b_i$ that minimize the distances between the markers and the corresponding vertices of the mesh (the *non unicity* problem).

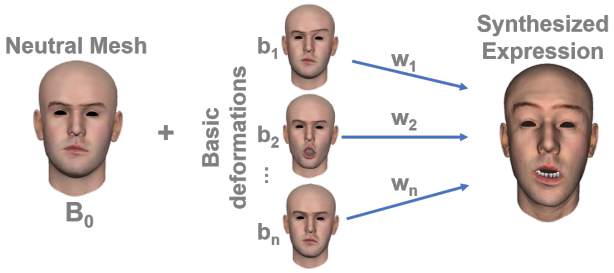


Figure 3: Synthesizing expressions from a linear combination of blendshapes.

We dealt with the *retargeting* problem as in (Bickel et al., 2007) or (Deng et al., 2006). Given one frame where the signer shows a neutral expression, a RBF regression is trained in order to make the link between the position of any point of the signer's face and the position it would have on the avatar's face:

$$\hat{M} = F_{RBF}(M) = \sum_{k=1}^{K} u_k f_k(M) \qquad (2)$$

where $\hat{M}$ is the estimated position of the signer's corresponding marker M retargeted on the avatar's mesh and $\{u_{1..K}\}$ are the optimized weights associated to the radial basis functions $\{f_{1..K}\}$.
The *non unicity* problem is formulated as a minimization problem in which the parameters $\{w_{1..n}\}$ are optimized so that the distances between the retargeted marker positions $\hat{M}_{1..40}$ and the corresponding vertices of the mesh $V_{1..40}$ are reduced. To ensure that the optimal weights found with this method do not generate visual artifacts, some constraints (e.g. non-negativity constraint) and/or some regularization energy that penalizes weights outside the $[0, 1]$ range can be incorporated. In our case, we introduced the *Thin-shell* model (Botsch and Sorkine, 2008) as a regularization energy that doesn't directly ensure that the weights stay between 0 and 1 but penalizes the bending and stretching deformations of the initial mesh:

$$\hat{W} = \underset{\{w_{1..K}\}}{\arg\min}(dist_{eucl}(\hat{M}_{1..40}, V_{1..40})^2 + E_{TS}) \quad (3)$$

with $\hat{W}$ the optimal vector of weights $\{w_{1..K}\}$ for the given expression and E_{TS} the *Thin-shell* energy. The vector of blendshape weights $\hat{W}$ is the chosen descriptor for the analysis of the affect channel.

4.2. Automatic Segmentation

The localization of segments of interest in a stream of sign language data is called the *segmentation*. It is done by detecting manually or automatically the temporal points corresponding to the beginning and the end of a behaviour (hand configuration or facial expression in our case). The coarseness of the behaviour to detect depends on the chosen annotation scheme. For example, sign language data can be segmented at a gloss level by detecting the beginning and end of a sign, or at a finer level such as facial expression, by detecting the beginning and end of an affect.

Segmentation of the Hand Configuration Channels
While signing, the signer alternates between stable periods where hand configurations stay the same (no or little motion of the fingers) and transitions between two consecutive hand configurations. The segmentation step therefore consists in separating the continuous signing sequences in two types of segments : *hand configuration* or *transition*. Only the *hand configuration* segments are labeled in the labeling step. To perform the segmentation, we assume that the variation of the distances is discriminating, i.e. the variation will be small during stable configurations and high during transitions. For each frame f, and for each selected distance SD, the variation of the distances $d(i, j)$ between two joints (i and j) is computed between the frame $f - 1$ and f. Those variations are summed (see Equation 4 for the right hand RH).

$$VarDist_{f,RH} = \sum_{i \in RH} \sum_{j \in RH d(i,j) \in SD} |d(i,j)_f - d(i,j)_{f-1}| \quad (4)$$

The segmentation relies on the use of a threshold. If *VarDist* is above this threshold, a *transition* segment will be detected. If *VarDist* is below the threshold, the segment will be recognized as a *hand configuration* segment. To select the value of the threshold and to evaluate our segmentation, we used the *Simple Matching Coefficient* (SMC) metric. It measures the similarity between two sets of values (here, the manual and the automatic segmentations). The SMC is the ratio of the number of overlapping frames between the two segmentations on the total number of frames. Figure 4 shows the variation of the SMC of the whole corpus with respect to the chosen threshold for the 29 normalized distances. The maximum (SMC = 81%) is reached for a threshold of 0.2. As manual annotation is performed by a human being on video footage, automatic annotation may be more accurate than manual annotation. Therefore, we consider this threshold satisfactory and it will be the one used in the following steps.

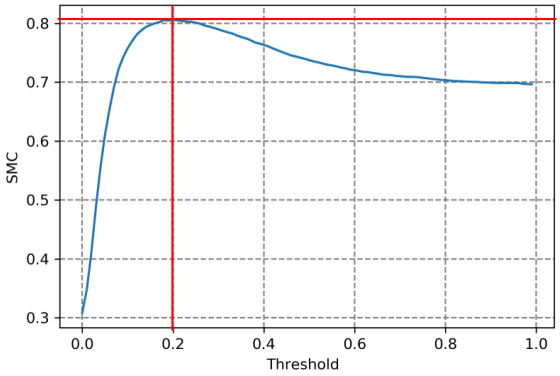


Figure 4: SMC with respect to the threshold values. The maximum (SMC = 81%) is reached for a threshold of 0.2.

Segmentation of the Affect Channel
The affect channel is segmented in a similar way, before the labeling step . We aim at detecting the frames that are located at the border between two segments. Since the border frames are related to the transitions from one expression to another we consider the energy curves of the velocity and acceleration of the n blendshape coefficients:

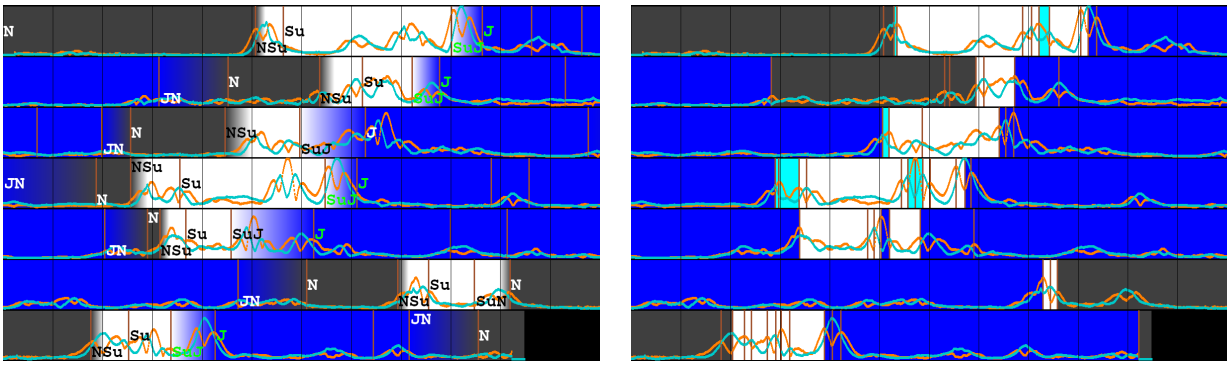
$$E_{VelBS} = \sum_{i=1}^{n} |\frac{dw_i}{dt}| \quad (5)$$

$$E_{AccBS} = \sum_{i=1}^{n} |\frac{d^2 w_i}{dt^2}| \quad (6)$$

In order to detect the local peaks, we consider the local optima of the curve $E^2_{Vel_{BS}} + E^2_{Acc_{BS}}$, and only keep those for which the local variation is important. This detection procedure is achieved by computing the derivative values on a window surrounding the detected optima, and applying a threshold. The orange and turquoise curves in Figure 5 show an example of segmentation using this method.

4.3. Automatic Labeling

The identification of the previously defined segments of interest is called the *labeling*. This task is highly dependent



Manually segmented and labeled sequence

Automatically segmented and labeled sequence

Figure 5: An example of automatic annotation: ”What? I won 1000 €!” repeated 3 times (white: surprise, blue: joy, cyan: fear; the orange curve stands for the sum of accelerations, the turquoise one for the sum of velocities; the vertical brown lines represent the limits of each segment; each vertical black line stands for 0.5 second).

on the chosen annotation scheme. Typically, it will consists in selecting the right label from a closed vocabulary to identify a segment.

Labeling the Hand Configuration Channels

A supervised machine learning approach is used to identify the handshape on each frame of the *hand configuration* segments. 32 classes were defined corresponding to 32 different handshapes (see fig. 6).

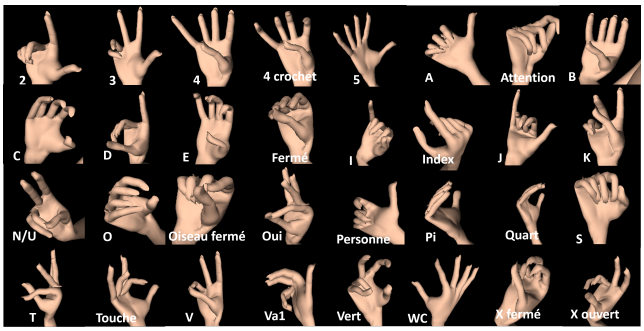


Figure 6: The 32 hand configurations played on an avatar.

Like for the segmentation step, the algorithm takes as input the chosen distances between the joints to characterize the hand configurations. Our machine learning classifiers are trained on 23533 manually annotated frames presenting those 32 configurations. The test set is composed of 3927 frames which amount to 14% of the total number of available examples. Our labeling approach sorts each frame in one of the 32 categories. We tested three different machine learning algorithms : logistic regression, support vector machine (SVM) with a *linear* kernel and k-nearest neighbors (kNN) on different subsets of our descriptors. Figure 7 shows the accuracy (number of correct predictions divided by the total number of predictions) on the test set depending on the machine learning algorithm and the subset of distances selected. We can see that the ”29 distances” subset presented in Section 4.1. gives the best results with the 3NN approach (91.2%) while the SVM on the 325 distances have the overall best accuracy (92.3%) (but the duration of the classifier training is longer).

Some configurations are more sensible to confusion than others. For example, the ’K’ and ’V’ configurations are often mistaken (in the two configurations, the middle and index fingers are raised; in the ’K’, there is a contact between the thumb and the base of the middle finger while there is not in the ’V’). The ’B’ and ’Pi’ configurations are also confused as only the thumb position is discriminant between the two configurations.

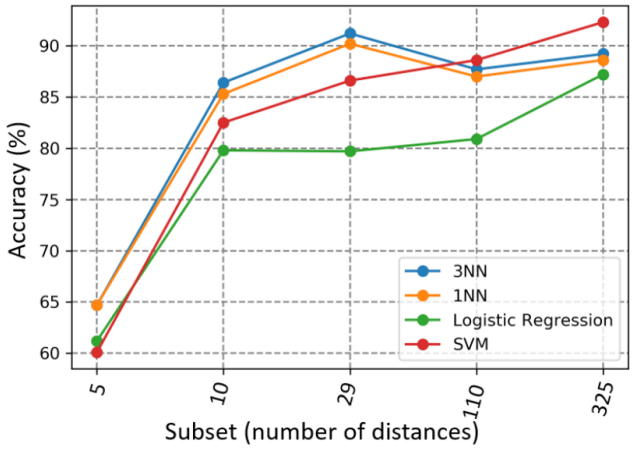


Figure 7: Accuracy on the test set for the hand configurations channel depending on the machine learning algorithm and selected distances.

Labeling the Affect Channel

The facial channel labeling is a supervised learning task aiming at identifying the correct class among the 7 defined in section 3.2.. Different methods were tested: kNN (*1NN* and *3NN*), SVM (*linear* and *RBF* kernels) and Random Forests (RF). The sequences recorded on each signer were processed separately. For each signer, the sequences IE and SE which represent roughly 50% of the data were used as the training set while the EU sequences were used as the test set. Whereas during the training phase, each frame of the training set with its corresponding manually annotated label was considered as a training sample, the test examples were constituted of the average along time of the frames composing each segment. These segments have been previously obtained according to the method presented in sec-

tion 4.2.. Each of these segments was classified as a whole represented by its average vector of blendshape weights:

$$\bar{\hat{W}} = \frac{\sum_{f=1}^{F} \hat{W}_f}{F} \quad (7)$$

with F the number of *frames* of the considered segment. Figure 5 shows an example of classification using this method; results are detailed in next section.

5. Results

The results of the automatic annotation of the hand configurations and of the facial expression channels are presented in this section.

5.1. Automatic Annotation of the Hand Configurations

To automatically annotate hand configurations during continuous signing, (i) the stream is segmented to distinguish *hand configuration* from *transitions* segments (section 4.2.), then, (ii) the hand configuration of each frame of the *hand configuration* segments is labeled (section 4.3.), and finally, (iii) each *hand configuration* segment is labeled according to the predominant class in the segment (see fig. 8).

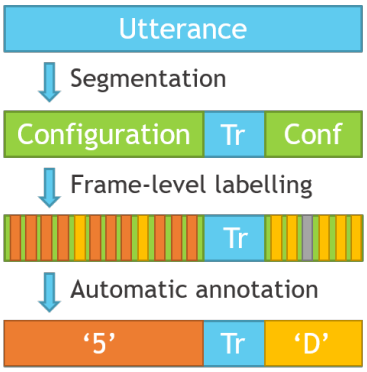


Figure 8: Overview of the automatic annotation of hand configurations.

Figure 9 shows three utterances of French Sign Language manually and automatically annotated. The segmentation threshold has been fixed to 0.2 and the machine learning algorithm used here is 3NN (*3-nearest neighbors*). We worked with the 29 distances described in section 4.1.. While the results can differ from one utterance to another, the recognized labels and segments are mainly consistent with the manual annotation. The results given by the metrics (i.e. accuracy, recall and precision) are computed with respect to the manual annotation and are therefore limited by the 80% overlap of the automatic segmentation with the manual segmentation. There are very few errors in terms of recognition of hand configurations (accuracy of 90%). A perceptual evaluation could give a better assessment of our results.

5.2. Automatic Annotation of the Affect channel

The sequences are first segmented according to the methods described in section 4.2.. Each segment is then labeled

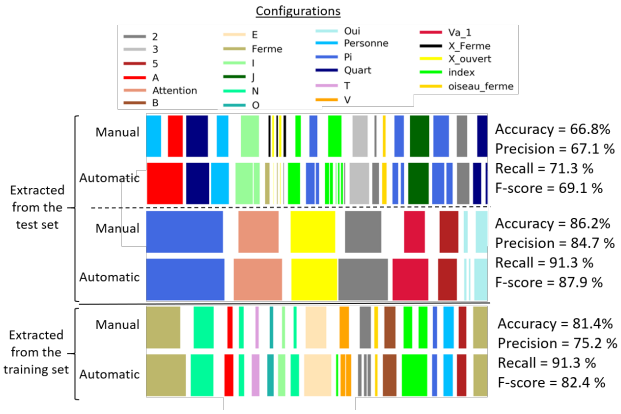


Figure 9: Result of the automatic annotation of hand configurations on three different utterances.

independently of the others, according to the following procedure. For each segment, we compute the average vector descriptor over time: $\bar{\hat{W}}$. This vector is used as input of the classification model (kNN or SVM or Random Forest) previously trained on the basis of the learning set. The classifier then returns the label associated with this segment. In order to evaluate the error due to the segmentation, the automatic annotation is performed on both the automatically detected segments and the manually defined ones. For both segmentations, Figure 10 gives the accuracy of the classifier for each tested algorithm. It shows that the best results are obtained with the Random Forest algorithm.

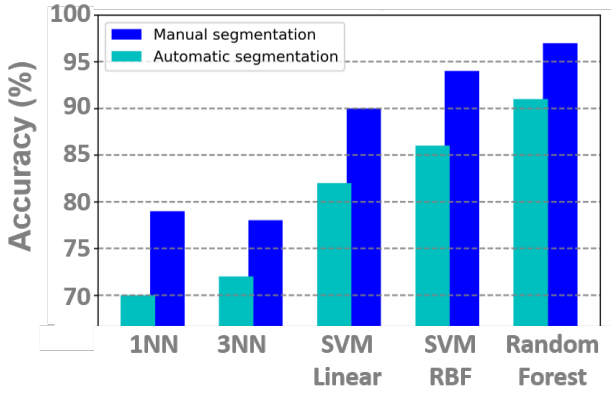


Figure 10: Accuracy on the test set for the affect channel depending on the machine learning algorithms and the segmentation.

6. Conclusion

We designed an approach to automatically annotate sign language MoCap data by processing each channel separately. We detailed the specific examples of hand configuration and facial expression annotation.

There are still many challenges to overcome. Using machine learning models, the automatic annotation could be significantly improved by increasing the size of the dataset, so that the training phase would be more efficient. In addition, following the approaches developed in language processing, we could also use models that learn the dynamics of the sequences, such as Hidden Markov Models, Con-

ditional Random Fields, or Recurrent Neural Networks. However, these methods require large databases.

Another challenge concerns the evaluation of the annotation results. Indeed, for the manual annotation, we rely on a ground truth which may be subject to errors or imprecision. This problem occurs for most recognition or annotation tasks. One solution could be to define a ground truth from a set of previously trained annotators, following strict instructions. In the near future, we plan to validate our annotations by defining quantitative metrics, both for hand configurations and facial expressions. As a complement to assess the quality of the annotation, we also plan to perceptually evaluate the results.

7. Acknowledgements

Part of the observations of this paper are based on the motion capture data and annotations of the Sign3D project (Gibet et al., 2015).

8. References

Aifanti, N., Papachristou, C., and Delopoulos, A. (2010). The mug facial expression database. In 11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10, pages 1–4, April.

Battison, R. (1978). Lexical borrowing in American sign language. ERIC.

Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M., Pfister, H., and Gross, M. (2007). Multi-scale capture of facial geometry and motion. In ACM SIGGRAPH 2007 Papers, New York, NY, USA.

Botsch, M. and Sorkine, O. (2008). On linear variational surface deformation methods. IEEE Trans. on Visualization and Computer Graphics, 14(1):213–230, January.

Busso, C., Bulut, M., Lee, C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J., Lee, S., and Narayanan, S. (2008). Iemocap: Interactive emotional dyadic motion capture database. Language Resources and Evaluation, 42(4):335–359, 12.

Deng, Z., Chiang, P.-Y., Fox, P., and Neumann, U. (2006). Animating blendshape faces by cross-mapping motion capture data. In Proce. of the 2006 Symposium on Interactive 3D Graphics and Games, pages 43–48, New York, NY, USA.

Dilsizian, M., Yanovich, P., Wang, S., Neidle, C., and Metaxas, D. (2014). A new framework for sign language recognition based on 3d handshape identification and linguistic modeling. In Nicoletta Calzolari (Conference Chair), et al., editors, Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Dreuw, P. and Ney, H. (2008). Towards automatic sign language annotation for the elan tool. In 3rd Workshop on the Representation and Processing of Sign Languages.

Gibet, S., Lefebvre-Albaret, F., Hamon, L., Brun, R., and Turki, A. (2015). Interactive editing in French sign language dedicated to virtual signers: requirements and challenges. Universal Access in the Information Society, 15(4):525–539.

Héloir, A., Gibet, S., Multon, F., and Courty, N. (2005). Captured motion data processing for real time synthesis of sign language. In Motion in Games.

Johnson, R. E. and Liddell, S. K. (2011). A segmental framework for representing signs phonetically. Sign Language Studies, 11(3):408–463.

Kim, J.-B., Park, K.-H., Bang, W.-C., and Bien, Z. (2002). Continuous korean sign language recognition using gesture segmentation and hidden markov model. In Proceedings of the 2002 IEEE International Conference on Fuzzy Systems.

Lefebvre-Albaret, F., Dalle, P., and Gianni, F. (2008). Toward a computer-aided sign segmentation. In Language Resources and Evaluation Conference (LREC). European Language Resources Association.

Lin, J. F. S., Karg, M., and Kulić, D. (2016). Movement primitive segmentation for human motion modeling: A framework for analysis. IEEE Transactions on Human-Machine Systems.

Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, pages 94–101, June.

Max Planck Institute for Psycholinguistics. (2017). Elan v.4.9.4. `http://tla.mpi.nl/tools/tla-tools/elan/`.

Naert, L., Larboulette, C., and Gibet, S. (2017). Coarticulation analysis for sign language synthesis. In International Conference on Universal Access in Human-Computer Interaction.

Stokoe, W. C. (1960). Sign language structure: An outline of the visual communication systems of the american deaf. Studies in Linguistics, Occasional Papers, 8.

Vogler, C. and Metaxas, D. (2001). A framework for recognizing the simultaneous aspects of american sign language. Computer Vision and Image Understanding, 81(3):358–384.

Yang, R. and Sarkar, S. (2006). Detecting coarticulation in sign language using conditional random fields. In Proceedings of the 18th International Conference on Pattern Recognition.

Yin, L., Chen, X., Sun, Y., Worm, T., and Reale, M. (2008). A high-resolution 3d dynamic facial expression database. In 2008 8th IEEE International Conference on Automatic Face Gesture Recognition, pages 1–6, Sept.

Zhang, X., Yin, L., Cohn, J. F., Canavan, S., Reale, M., Horowitz, A., Liu, P., and Girard, J. M. (2014). Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. Image and Vision Computing, 32(10):692 – 706. Best of Automatic Face and Gesture Recognition 2013.