Recognition of Signed Expressions Using Cluster-Based Segmentation of Time Series

Mariusz Oszust and Marian Wysocki

Rzeszow University of Technology
Department of Computer and Control Engineering
W. Pola 2, 35-959 Rzeszow, Poland
{moszust, mwysocki}@prz-rzeszow.pl

Abstract. The paper considers automatic visual recognition of signed expressions. The proposed method is based on modeling gestures with subunits, which is similar to modeling speech by means of phonemes. To define the subunits a data-driven procedure is applied. The procedure consists in partitioning time series, extracted from video, into subsequences which form homogeneous groups. The cut points are determined by an evolutionary optimization procedure based on multicriteria quality assessment of the resulting clusters. In the paper the problem is formulated, its solution method is proposed and experimentally verified on a database of 100 Polish words.

Keywords: sign language recognition, time series segmentation, multiobjective clustering, evolutionary optimization, computer vision

1 Introduction

Automatic vision-based sign language recognition is an important prospective application of gesture-based human-computer interfaces. The aim of the research is a system that properly interprets gestures, e.g. translates them into written or spoken language. Most of such systems described in the literature (see e.g. [1], [2]) are based on word models where one sign represents one model in the model database. They can achieve good performance only with small vocabularies or gesture data sets. The training corpus and the training complexity increase with vocabulary size. So, large-vocabulary systems require the modeling of signed expressions in smaller units than words i.e. the words are modeled with subunits, which is similar to modeling speech by means of phonemes. The main advantage

Please cite this paper as follows: Oszust M., Wysocki M.: Recognition of Signed Expressions Using Cluster-Based Segmentation of Time Series, Choras R. (Ed.): Image Processing and Communications Challenges 2, Advances in Soft Computing Springer-Verlag Berlin / Heidelberg, pp. 167–174, 2010. The final publication is available at http://link.springer.com/chapter/10.1007%2F978-3-642-16295-4_19

of this approach is that an enlargement of the vocabulary can be achieved by composing new signs through concatenation of subunit models and by tuning the composite model with only small sets of examples. However, an additional knowledge of how to break down signs into subunits is needed.

Different vision-based subunit segmentation algorithms have been developed. Following Liddell and Johnson's movement-hold model the authors of [3] propose modeling each sign (word) as a series of movement and hold segments. Kraiss et al. in [1] present an iterative process of data-driven extraction of subunits using hidden Markov models (HMMs). Han et al. in [4] define the subunit boundary using hand motion discontinuity.

In this paper we propose a new approach where the subunits' boundary points are considered as decision variables in a multiobjective optimization problem. The problem consists in finding subunits which can be grouped in clusters of good quality. The quality is measured by two cluster validity indices, one based on entropy [5] and another the Dunn index [6], [7]. The indices are optimized simultaneously using lexicographic ordering [8] and an immune-based evolutionary algorithm [9], [10]. The approach refers to clustering of time series data [11], [12], multiobjective clustering [13], [14], and cluster-based time series segmentation [15]. The contribution of the paper lies in (1) formulation of the problem of determining subunits for sign language recognition as a multiobjective cluster optimization, (2) formulation of the problem of modeling signed expressions with the subunits, (3) proposition of solution methods, and verification of the approach by experiments on real data.

The rest of the paper is organized as follows. Section 2 formulates the problem of subunits extraction and describes the solution method. Section 3 gives details of the subunit-based recognition method. The results of experiments with recognition of 100 isolated words of the Polish Sign Language (PSL) are given in section 4. Section 5 concludes the paper.

2 A data-driven subsequence extraction method

2.1 The input data

Most of the sign gestures are two-handed and dynamic. Let $S = \{X_1, X_2, \ldots, X_n\}$ denote a data set, where a sequence $X_i = \{x_i(1), x_i(2), \ldots, x_i(T_i)\}$ of real valued feature vectors represents a signed word. All vectors $x_i(t)$, where $i \epsilon I = \{1, 2, \ldots, n\}$, and t is a time sampling point, $t \epsilon \mathcal{T}_i = \{1, 2, \ldots, T_i\}$, have identical structure. They contain features extracted from image sequences registered by a camera. For instance, we use seven manual features for the right hand and the same features for the left hand: the position of the hand with respect to the face (three spatial coordinates), the area, orientation, compactness and eccentricity of a hand (four features as a very simplified information about the hand shape). Two time sequences X_i and $X_{j \neq i}$ may represent different words or different realizations of the same word. In modeling signed expressions we should take into account that the features we are observing appear both sequentially

and simultaneously. For example, the hand shape and hand position can change independently at the same time [1]. To model parallel processes we will distinguish N groups of features (channels). This is based on the assumption that the separate processes evolve independently from one another with independent output. So, we will write $x_i(t) = [x_i^1(t), x_i^2(t), \ldots, x_i^N(t)]$ and, in accordance with it, we will use an upper index to indicate time series related to a group: $X_i^l = \{x_i^l(1), x_i^l(2), \ldots, x_i^l(T_i)\}, S^l = \{X_1^l, X_2^l, \ldots, X_n^l\}, l\epsilon\mathcal{N} = \{1, 2, \ldots, N\}$. During extraction of subunits all elements in a group will be considered jointly, whereas different groups will be considered separately. For instance, one can assign one channel to one hand ($N = 2$) or one channel to one of the 14 features mentioned earlier ($N = 14$). In general, the number N as well as the assignment of the features to groups can be a subject of further research.

2.2 Sequence partitioning problem

Let us consider a time decomposition D^l, which, for each $i\epsilon I$, defines a number $k_i^l = k_i^l(D^l) \geq 1$ and k_{i-1}^l cut points $t_{ij}^l = t_{ij}^l(D^l)$, where $1 < t_{i1}^l < t_{i2}^l < \ldots < t_{i,k_i^l-1}^l < T_i$. The decomposition means that X_i^l is partitioned into k_i^l subsequences. The first subsequence $s_{i1}^l(D^l)$ starts at $t = 1$ and ends at $t = t_{i1}^l$, the next subsequence $s_{i2}^l(D^l)$ starts at $t = t_{i1}^l$ and ends at $t = t_{i2}^l$, and so on until the last subsequence $s_{i,k_i^l}^l(D^l)$ which starts at $t = t_{i,k_i^l-1}^l$ and ends at T_i. The resulting data set $S'^l(D^l) = \{s_1'^l(D^l), s_2'^l(D^l), \ldots, s_n'^l(D^l)\}$, where $s_i'^l(D^l) = \{s_{i1}^l(D^l), \ldots, s_{i,k_i^l}^l(D^l)\}, i\epsilon I$, contains $n'^l = n'^l(D^l) = \sum_{i=1}^n k_i^l(D^l)$ subsequences. The length of each subsequence is constrained by the minimal L_{min} and the maximal L_{max} number of points. We propose determining a good decomposition into subsequences by solving a multicriteria decision problem, based on the following main steps: (i) partition the set $S'^l(D^l)$ into m^l (a given number) clusters, i.e. $S'^l(D^l) = \{C_1^l(D^l), C_2^l(D^l), \ldots, C_{m^l}^l(D^l)\}$, (ii) evaluation of the decomposition D^l using a vector of two criteria (indices) $J(D^l) = [J_1(D^l), J_2(D^l)]$ which characterizes the quality of the resulting clusters. The first criterion is the conditional entropy minimized by the minimum entropy clustering (MEC) algorithm described in [5]. Experiments presented in [5] show that MEC performs significantly better than k-means clustering, hierarchical clustering, SOM and EM. Moreover, it can correctly reveal the structure of data and effectively identify outliers simultaneously. To compare discrete sequences we use dynamic time warping (DTW) [6],[16]. DTW aligns two sequences while attempting to achieve the minimal difference. The warping path with the optimal distance d_{DTW} can be obtained by dynamic programming. The second criterion is the Dunn index DI [6], [7]. It is defined by two parameters: the diameter $diam(C_i^l)$ of the cluster C_i^l and the set distance $\delta(C_i^l, C_j^l)$ between C_i^l and C_j^l, where

$$diamC_i^l = max_{x,y\epsilon C_i^l}\{d(x,y)\}, \delta(C_i^l, C_j^l) = min_{x\epsilon C_i^l, y\epsilon C_j^l}\{d(x,y)\} \qquad (1)$$

and $d(x, y)$ indicates the distance between points x, y.

$$DI = min_{1 \leq j \leq m^l} \{ min_{1 \leq i \leq m^l, i \neq j} \{ \frac{\delta(C_p^l, C_q^l)}{max_{1 \leq k \leq m^l} diam C_k^l} \} \} \tag{2}$$

Larger values of DI correspond to good grouping with compact and well separated clusters.

2.3 Optimization method

As follows from subsection 2.2 our problem is a multiobjective optimization problem (MOP) with two criteria. To solve MOPs evolutionary algorithms are often used. Evolutionary algorithms deal simultaneously with a set of possible solutions (the so–called population) which allow us to find several members of the Pareto optimal set in single run of the algorithm [9]. Our approach to solve the MOP adopts the immune-based algorithm CLONALG originally used for ordinary optimization [9], [10]. We use lexicographic ordering [8]. Here the single objective J_1 (considered the most important) is optimized without considering J_2 Then the J_2 is optimized but without decreasing the quality of the solution obtained for J_1. In the sequel we shortly describe the algorithm, the encoding method, and the mutation operator.

CLONALG. The main loop (repeated *gen* times, where *gen* is the number of generations) consists of four main steps: one initial step where all the elements of the population are evaluated and three transformation steps: clonal selection, mutation, apoptosis.

1. Evaluation. For each element D^l in the population P compute $J_i(D^l), i = 1, 2$ and perform lexicographic ordering of the elements.
2. Clonal selection. Choose a reference set $P_a \subset P$ consisting of h elements at the top of the ranking obtained in step 1.
3. Mutation.
   (a) For each $D^l \epsilon P_a$ make c mutated clones $Dc_j, j = 1, 2, \ldots, c$, compute their values $J_1(Dc_j), J_2(Dc_j)$, and place the clones in the clonal pool CP.
   (b) Lexicographically order the elements of $P \cup CP$, choose a subset $P_c \subset P \cup CP$ containing B best elements, where B denotes the size of P.
4. Apoptosis. Replace b worst elements in P_C by randomly generated elements.
5. Set $P \subset P_C$.

In the algorithm the current population P is mixed with the clonal pool CP and the predefined number of best elements (i.e. at the top of the ranking) is picked up to form new population. The last step of the main loop replaces b worst solutions by randomly generated elements.

Encoding and mutation. Each element of the population P represents a decomposition D^l of the set S^l into a set S'^l (see section 2.2). It has the form of the integer valued vector $D^l = [t_{11}^l, t_{12}^l, \ldots, t_{1,k_1^l-1}^l, t_{21}^l, t_{22}^l, \ldots, t_{2,k_2^l-1}^l, \ldots,$
$t_{n1}^l, t_{n2}^l, \ldots, t_{1,k_n^l-1}^l]$ composed of the cut points of the original sequences.

The mutation process consists of a given number M of mutations conducted on a population element. The mutation means an operation randomly chosen from the following variants: (a) add cut point (probability 1/4), (b) remove cut point (probability 1/4), (c) move cut point (probability 1/2). In all cases a subsequence is randomly selected and, depending on a drown variant, it is: (a) divided into two shorter subsequences, (b) joined together with its preceding subsequence, (c) made shorter or longer by shifting its initial point. New cut point in (a) and (c) is placed in a position randomly chosen from the corresponding set of feasible points, i.e. the points for which the resulting subsequences satisfy the length constraints. Similarly, the union in (b) is accepted if the resulting subsequence is not too long.

The optimization results in obtaining a good decomposition D^l_{opt}. We can use it to transform each sequence X^l_i to a string of labels $\tilde{X}^l_i = \{e^l_{i1}, e^l_{i2}, \ldots, e^l_{i,ki}\}$, where $e^l_{ik} \epsilon E^l = \{\alpha^l_1, \alpha^l_2, \ldots, \alpha^l_{m^l}\}$, α^l_k denotes the label assigned to the cluster C^l_k, and e^l_{ik} is a label of the cluster the subsequence $s^l_{ik}(D^l_{opt})$ belongs to. Let us denote by $\tilde{X}_i$ the string representation of X_i, i.e. $\tilde{X}_i = \{\tilde{X}^1_i, \tilde{X}^2_i, \ldots, \tilde{X}^N_i\}$ and, consequently, by $\tilde{S}$ the counterpart of S.

3 Subunit–based recognition

Let us assume that a word to be classified is represented by a sequence $Y = \{y(1), y(2), \ldots, y(T_y)\}$. The feature vectors $y(.)$ have the same structure as $x(.)$ and therefore the sequences $Y^l = \{y^l(1), y^l(2), \ldots, y^l(T_y)\}$, where $l \epsilon \mathcal{N}$, will be considered separately. Two problems have to be solved. The first problem consists in finding an appropriate string representation of Y^l, i.e. $\tilde{Y}^l = \{e^l_{y1}, e^l_{y2}, \ldots, e^l_{y,k^l_y}\}$, where $e^l_{ik} \epsilon E^l$ and, consequently, the string representation $\tilde{Y}$ of Y. The second problem is to find $NN(\tilde{Y})$ – the nearest neighbor of $\tilde{Y}$ in the set $\tilde{S}$. Then the unknown word is assigned to the class which $NN(\tilde{Y})$ belongs to.

The string representation can be found by solving an optimization problem with respect to cut points of Y^l for each $l \epsilon \mathcal{N}$. Let $D^l_y = [t^l_{y1}, t^l_{y2}, \ldots, t^l_{y,k^l_y-1}]$ characterizes a decomposition. As opposed to the previous optimization, now the criterion to be minimized is $J(D^l_y) = \sum_{k=1}^{k^l_y} d_{DTW}(k)$, where $d_{DTW}(k)$ denotes the DTW distance between the k-th subsequence $s^l_{y,k}(D^l_y)$ of Y^l and its nearest neighbor $NN(s^l_{y,k}(D^l_y))$ in the set $S'^l(D^l_{opt})$. The optimization task can be solved by CLONALG. Then e^l_{jk} is a label of the cluster the $NN(s^l_{y,k}(D^l_{y,opt}))$ belongs to. The procedure is repeated for each $l \epsilon \mathcal{N}$. The second problem is also an optimization task. Here the so called edit distance [6] is used as a measure of the difference between two strings. The method resembles DTW. It uses dynamic programming to find a minimum number of operations (insert, delete, replace) required to transform one string into the other. Let us denote by d^l_i the edit distance $d_{ED}(\tilde{Y}^l, \tilde{X}^l_i)$ between the string $\tilde{Y}^l$ and a string $\tilde{X}^l_i$. The similarity measure between the sequence Y and a sequence X_i is the sum $d_i = \sum_{l=1}^{N} w^l d^l_i$, where w^l denotes a weight assigned to the l-th component of the feature vector.

In particular, all the weights are equal to one. The sequence Y becomes assigned to the class X_j belongs to, where $j = arg\ min_{i \epsilon I}(d_i)$.

4 Experiments

In this section we present the results of experiments based on real sequences obtained for signed Polish words. The sequences represent 100 words which can be used at the doctor's and in the post office. Each word is characterized by a vector of 14 features mentioned in section 2.1. We used a data set consisting of sequences of feature vectors for 40 realizations of each of the 100 words. Gestures have been performed by two signers. One person is a PSL teacher, the other has learnt PSL for purposes of this research. Each signer repeated each word 20 times. The data have been registered with the rate 25 frames/s. To perform cross–validation we divided the data set into four disjoint subsets. Each subset consisted of data corresponding to 10 repetitions of each word (5 repetitions performed by each signer). We performed four experiments using three subsets as the training set S and the remaining subset as the test set. The experiments have been labeled with $A, \ldots, D$. In each experiment the data in S were used to extract the subsequences as described in section 2 and the remaining elements were classified by the method described in section 3. Sample results of recognition are given in table 1. Subunits for each feature were extracted independently ($N = 14$), thus 14 symbolic transcriptions were assigned to each word in S. Parameters used by immune algorithm were as follows: $B = h = 10; c = 5; b = 2; M = 2; gen = 5; L_{min} = 4; L_{max} = 8$. We solved the optimization task for $m^l = 10$ clusters. Exemplary subsequences, obtained for the horizontal placement of the right hand center, extracted from words in the training set, with related symbolic transcriptions are shown in fig. 1. Automatically determined subsequences' boundaries are marked with crosses. Resulting symbolic transcriptions based on ten subunits are given in brackets. Immune algorithm for creating symbolic transcriptions of the sequences from the test set used identical parameters as those for extracting subunits. Table 1. shows the recognition rates. Results are quite promising and comparable with the results we obtained on the same dataset using heuristic definitions of visually-oriented subunits and related parallel hidden Markov models [18].

Table 1. Results of the cross–validation test. Recognition rate in %. Because of randomness of the optimization algorithms each test was repeated ten times.

	A	B	C	D	Cross–validation mean
Minimum	82,0	92,4	91,8	89,4	
Maximum	85,7	94,2	95,2	92,1	
Mean	84,2	93,0	93,0	90,3	90,1
StDev	1,6	0,7	1,3	1,0	

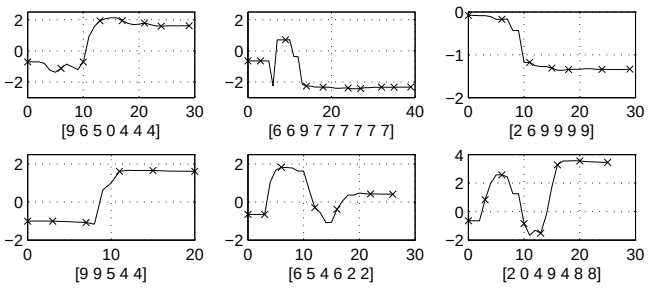

Fig. 1. Sequences representing six chosen words from a training set (from left to right, from top to bottom: *pharmacy, audiogram, angina, ambulance, parcel, man*)

5 Conclusions

Large-vocabulary systems of sign language recognition require the modeling of signed expressions in smaller units than words. However, an additional knowledge of how to break down signs into subunits is needed. In vision-based systems the subunits are related to visual information. As linguistic knowledge about the useful partition of signs in regard of sign recognition is not available, the construction of an accordant partition is based on a data–driven process when signs are divided into segments that have no semantic meaning – then similar segments are grouped and labeled as a subunit. In this paper we propose a new approach to determining the subunits. Subunits' boundaries are considered as decision variables in a multiobjective optimization problem. We use two objective functions, entropy and the Dunn index, as measures of cluster quality. These functions are optimized simultaneously. The method has been successfully verified using a database of 100 Polish words, but there remain some open questions concerning eg. the number of clusters, cluster validity indices, optimization methods etc. We will consider these issues in future research. A next step will be related to more advanced experimentation including recognition words and sentences of PSL. Interesting questions concern the choice of features and assignment of the features to groups during definition of subunits (section 2.1), as well as dependence of new words recognition on signer and size of training sets.

Acknowledgement

This research was supported by the Polish Ministry of Higher Education under grant N N516 369736.

References

1. Kraiss K.F (2006) Advanced man-machine interaction. Springer, Berlin

2. Ong SCW, Ranganath S (2005) Automatic sign language analysis: a survey and the future beyond lexical meaning. IEEE Trans. PAMI 27:873–891
3. Vogler C, Metaxas DA (2001) Framework for recognizing the simultaneous aspects of american sign language. Computer Vision and Image Understanding 81:358–384
4. Han J, Awad G, Sutherland A (2009) Modelling and segmenting subunits for sign language recognition based on hand motion analysis. Pattern Recognition Letters 30:623–633
5. Li H, Zhang K, Jiang T (2004) Minimum entropy clustering and applications to gene expression analysis. In: 3rd IEEE Computational Systems Bioinformatics Conference, 142–151
6. Xu R, Wunsch DC (2009) Clustering. J. Wiley and Sons, Inc., Hoboken, New Jersey
7. Maulik U, Bandyopadhyay S (2002) Performance evaluation of some clustering algorithms and validity indices. IEEE Trans. PAMI 24:1650–1654
8. Miettinen KM (1998) Nonlinear multiobjective optimization. Kluwer Acad. Publ.
9. De Castro LN, Von Zuben FJ (2002) Learning and optimization using the clonal selection principle. IEEE Trans. on Evolutionary Computation 6:239–251
10. Trojanowski K, Wierzchon S (2009) Immune-based algorithms for dynamic optimization. Information Sciences 179:1495–1515
11. Bicego M, Murino V, Figueiredo MAT (2004) Similarity-based classification of sequences using hidden markov models. Pattern Recognition 37:2281–2291
12. Liao TW (2005) Clustering of time series data – a survey. Pattern Recognition 38:1857–1874
13. Handl J, Knowles J (2007) An evolutionary approach to multiobjective clustering. IEEE Trans. on Evolutionary Computation 11:56–76
14. Saha S, Bandyopadhyay S (2010) A symmetry-based multiobjective clustering technique for automatic evolution of clusters. Pattern Recognition 43:738–751
15. Tseng VS, Chen CH, Huang PC, Hong TP (2009) Cluster-based genetic segmentation of time series with DWT. Pattern Recognition Letters 30:1190–1197
16. Ratanamahatana CA, Keogh E (2005) Three myths about dynamic time warping data mining. In: SIAM Int. Conf. on Data Mining 506–510
17. Minimum Entropy Clustering Java package, `http://www.cs.ucr.edu/~hli/mec/`
18. Kapuscinski T, Wysocki M (2007) Recognition of signed Polish words using visually oriented subunits. In: 3rd Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. Poznan, 202–206