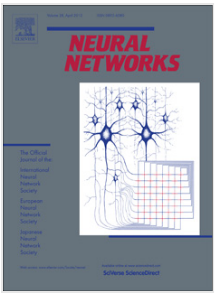

Non parametric, self organizing, scalable modeling of spatiotemporal inputs: The sign language paradigm

G. Caridakis, K. Karpouzis, A. Drosopoulos, S. Kollias

PII: S0893-6080(12)00262-6
DOI: 10.1016/j.neunet.2012.10.001
Reference: NN 3086

To appear in: *Neural Networks*

Received date: 26 July 2011
Revised date: 3 October 2012
Accepted date: 4 October 2012

Please cite this article as: Caridakis, G., Karpouzis, K., Drosopoulos, A., & Kollias, S. Non parametric, self organizing, scalable modeling of spatiotemporal inputs: The sign language paradigm. *Neural Networks* (2012), doi:10.1016/j.neunet.2012.10.001

Non parametric, self organizing, scalable modeling of spatiotemporal inputs: The Sign Language Paradigm

G. Caridakis, K. Karpouzis, A. Drosopoulos, S. Kollias
Intelligent Systems, Content and Interaction Lab
National Technical University of Athens
gcari, kkarpou, ndroso, stefanos@image.ntua.gr

Abstract

Modeling and recognizing spatiotemporal, as opposed to static input, is a challenging task since it incorporates input dynamics as part of the problem. The vast majority of existing methods tackle the problem as an extension of the static counterpart, using dynamics, such as input derivatives, at feature level and adopting artificial intelligence and machine learning techniques originally designed for solving problems that do not specifically address the temporal aspect. The proposed approach deals with temporal and spatial aspects of the spatiotemporal domain in a discriminative as well as coupling manner. Self Organizing Maps (SOM) model the spatial aspect of the problem and Markov models its temporal counterpart. Incorporation of adjacency, both in training and classification, enhances the overall architecture with robustness and adaptability. The proposed scheme is validated both theoretically, through an error propagation study, and experimentally, on the recognition of individual signs, performed by different, native Greek Sign Language users. Results illustrate the architecture's superiority when compared to Hidden Markov Model techniques and variations both in terms of classification performance and computational cost.

Keywords: Spatiotemporal pattern recognition, Self Organizing Maps, Markov Models, Probabilistic, Sign language recognition

2010 MSC: 68T10 Pattern recognition, 60J45 Probabilistic potential theory, 60J10 Markov chains

1. Introduction

Challenges and open issues concerning the applicability and extensibility of approaches that aim at tackling spatiotemporal problems include resistance to noise and variability (w.r.t. user/repetition gesture performance) in the input channel, computational efficiency of the recognition scheme adopted, large scale dictionary registration and recognition and dictionary extension without the need of extensive retraining. Additionally, fusion of multiple modalities and usage of arbitrary, or experimentally defined, initialization parameters, such as the number of HMM states significantly influences performance, generalization and adaptability of the majority of approaches. In the proposed scheme, dedicated per modality classifiers are trained in order to model different recognition aspects and are consequently fused at decision level. This approach resembles Boosting of weak classifiers; however, the classifiers used in the proposed scheme are suitable for tackling particular aspects of the recognition task and not weak, generic classifiers. Input variability is addressed through the flexibility, provided by state transition probability dispersion during Markov chains training and by optimal path search performed during classification, both based on SOM neighborhood properties. Spatial modeling is achieved using a SOM, trained with a representative sample of hand positions during signing. Spatial modeling is performed once, annulling the need for exhaustive retraining when an unknown class is introduced to the vocabulary. Given that the initial training set is representative in terms of signing space distribution, no additional training is required, since the signing space has been well modeled. The SOM nodes neighboring relation, formed during training, consists a crucial characteristic of the overall training and classification process. It is driving the adaptive nature of the overall approach, tackling large scale vocabulary application issues. A modified algorithm is used for calculating the Levenshtein distance, also taking into account the similarity of sequence's symbols, addressing the problem of potential variation or noise in the input channel.

Sign Language (SL) is the linguistic system used by the hearing disabled group, in order for the members of the group to communicate amongst themselves and also with hearing able people. Unlike spoken languages, sign languages are heavily based on iconicity to convey meaning. A morphosyntactic structure is employed to express linguistic relations in 3D space. Concepts are represented by signs, the basic grammatical unit of a sign language, forming a visual natural language. Sign language analysis and recognition can be viewed as a spatiotemporal problem incorporating the issues discussed previously, as well as a wide range of concepts. It includes pure image analysis tasks, ranging from locating and tracking the face and hands of the signer up to notions related with semantics and context, usually found in natural language processing paradigms.

Validation of the proposed approach is performed through its application to the SL analysis problem. At first we locate the head and hands of the signer in order to extract features related to both handshape and hand location. Then, hand positions are used to train Self Organizing Maps, so as to effectively represent the signing space, tackling the spatial aspect of the recognition task. First order Markov chains, which use the SOM units as states, are used to cope with the temporal aspect. Fusion of SOM and Markov chains is performed by a greedy algorithm seeking to make a locally optimal choice at each stage and converge to a global solution. Intra and inter user spatial or performance variation and random errors in the input stream are tackled by incorporating the neighborhood property of the models' states in the overall classification process, thus enhancing the overall architecture with robustness and adaptability. Separate classifiers, namely Markov models for hand position and movement and Hidden Markov Models for handshape features are fused on a decision level, in a committee-machine-like setup, further ensuring stability of the recognition process. Application of the proposed architecture has a low computational cost, making it therefore suitable for realtime applications. Experimental results, discussed in section 4.2 and performed on two datasets, a synthetic and a Greek SL corpus, illustrate the architecture's superiority both in terms of classification performance and computational cost over popular techniques such as Hidden Markov Models and their variations (Multi-Stream, Parallel and Product HMMs). Initial validation on the synthetic dataset have been presented on [9] and current work builds on this and enhances the approach with:

- transition probability spreading during training providing robustness against noise and variability in the input channel
- incorporation of a novel distance calculation algorithm based on the

Levenshtein distance metric which takes into consideration similarity of symbols

- a distributed approach, tackling individual aspects of the handshape such as boundary and region, aiming to model an extremely complex pattern, such as the handshape, especially for 2D projection, and challenging, in terms of automatic recognition, finger configurations

- incorporation of multiple modalities and appropriate multimodal fusion; the latter balances each stream's contribution to the final decision according to respective unimodal classification results

- extensive experimentation with datasets featuring native signers that illustrate the architecture's superiority to current state of the art schemes, both in terms of classification performance and computational cost

.

The remaining of the paper is organized as follows: section 2 discusses aspects, challenges and previous work related to automatic Sign Language recognition, by critically reviewing each approach, bringing forth the focus and the strong points of each article. Section 3 introduces the proposed architecture and is roughly divided into the learning process (section 3.2) and the classification process (section 3.6). The proposed approach is validated: a) theoretically in section 3.6.1, by studying the propagation of error when a random error is introduced in the input stream and b) experimentally in section 4, by applying the learning and classification scheme on the Greek Sign Language Corpus [17] featuring three native signers performing representative lemmata of the Greek Sign Language. Finally, the article is summarized in section 5 where future directions of the presented research work are also discussed.

2. Machine learning and SL recognition

An abundance of automatic sign language recognition techniques can be found in the literature, differentiating in terms of input streams, extracted features, vocabularies, signer dependence, isolated or continuous recognition [39] and [1]. The input stream can be either based on the use of motion capture (direct-measure device) data gloves [53, 63] or consist of visual signals. Datagloves are quite expensive and intrusive, however they constitute a robust and accurate way of capturing 3D hand location and finger flexion in real time. Motion capture is used in [49] while time-of-flight camera is employed in [21] and visual and device inputs are fused in [8]. In all approaches features are extracted from the gestured input stream mainly based on position of the dominant right hand. Usually, when motion capture is employed, the 3D position is included in the features set, but for vision based approaches, only the 2D projection of the hand position can be extracted and 3D can only be calculated in conjunction with stereo vision. The position of the hand is relative to some reference point, for example, the head of the user or his/her back in case of data capturing by placing an additional sensor on the back of the signer. Another important issue on which automatic sign language recognition is based on, is the vocabulary size of the experimental corpora. In most cases, the experimental dataset is composed of a quite restricted number of signs (≈ 50); only in a small number of cases this is extended [15, 22, 21, 63]. Additionally, signer dependence and sign variation are decisive aspects when trying to implement architectures into real world. Signer independence, and ways to tackle both intersigner variation in the performance of the signs and grammatical idioms of signer groups, are vital for achieving good recognition rates in an arbitrary setting by an unregistered user.

An important issue in the wide research area of automatic sign language recognition is the vocabulary size of the experimental corpora used to verify the robustness and generalization capabilities of the proposed systems, as can be seen in Table 1. Most articles construct their experimental dataset using a quite restricted number of signs, varying from 10 to 65, while others extend their vocabulary, to what would be a more representative sample of the respective sign language, but still use between 164 and 274 signs. The articles that approximate a universal recognizer are those who reach impressive vocabulary sizes that enumerate up to more than 5000 signs. Publications belonging to the last group focus mainly on large vocabulary recognition and issues related to a complete, real time system that could support automatic sign language recognition. The number of repetitions performed for each vocabulary entry is also important, as is the training/testing sample ratio. Typically each sign is repeated 5-10 times e.g.[7, 11, 12, 32, 36, 54, 52, 65], while there are cases where more [34, 26, 29] or less [3, 18, 30, 44, 55] repetitions are performed for each lemma in the restricted, experimental vocabulary.

Residing in the heart of each recognition system, the classification architecture is considered the most important one. Although a plethora of works propose a single off the shelf classifier, there is also a significant number of approaches that utilize a combination of classification schemes. Such schemes

Size	Work
$10 < Size < 65$	[2, 5, 4, 6, 8, 13, 14, 26, 27, 28, 30, 34, 40, 42, 43, 44, 45, 48, 50, 51, 53, 55, 56, 59, 62]
$164 < Size < 274$	[3, 7, 11, 12, 18, 25, 32, 41, 54, 65]
$Size > 5000$	[15, 19, 20, 24, 23, 22, 21, 31, 36, 52, 64, 63]

Table 1: Vocabulary Size

include HMM and variants, Neural Networks, Boosting Techniques, Linear Models, Tree structures, Clustering and State Sequence Comparison.

2.1. HMM and variants

HMMs can model spatiotemporal information in a natural way. These models have the ability to compensate time and amplitude variances, as has been proven in speech and character recognition. As a result, this approach dominates, in terms of popularity in Sign Language Recognition. The drawback in this case is the need to collect extensive amounts of data and the demanding processing time to estimate corresponding HMM parameters. Additionally, the architecture's sensitivity to arbitrary or experimentally defined architectural decisions (e.g. number of states) constitutes another weakness of the approach. Finally, in signer independent scenarios, HMMs seem to fail to adapt and generalize well [65].

Conventional HMMs [6, 8, 13, 65] have been used extensively while interesting variants have been also presented. Fang, Gao et al. have been very active and published several articles on sign language recognition adopting HMM variants. Temporal clustering with k-means has been proposed to cluster the temporal sequence of the vectors. Dynamic Time Warping is employed as the distance computation criterion since it can measure the distance between two temporal sequences. This is achieved by aligning different time signals and normalizing them to a warping function through search for the minimal accumulating distance and the associated warping path. A SOFM/SRN/HMM model [22] has been used for signer-independent continuous SLR. Wang et al. also deal with the issue of scaling with increasing vocabulary sizes based on phoneme recognition process; in [55] they cope with this issue utilizing a homography like scheme where each sign is represented as a series of tiny hand motions and segmented into atomic units of 3 consecutive frames. Zhang et al. in [64] incorporate a Tied-Mixture Density Hidden Markov Model which speeds up recognition without significant loss of recognition accuracy. Factorial HMMs and Coupled HMMs model several processes occuring in parallel, while Parallel HMMs [49] model parallel processes independently. In the above, training times are polynomial to the number of states, and linear to the number of parallel processes and the decoding algorithm is a token passing instead of the standard Viterbi algorithm.

2.2. Boosting

Boosting is a general method that can be used for improving the accuracy of a given learning algorithm. More specifically, it is based on the principal that a highly accurate or ”strong” classifier can be produced through linear combination of many inaccurate or ”weak” classifiers. In general, the performance of an individual weak classifier may be only slightly better than random. Cooper and Bowden in [11] present an approach to large lexicon sign recognition that does not require tracking, based on the AdaBoost boosting algorithm to detect present visemes. The same authors in [12] experiment with different boosting classifiers, such as AdaBoost and AdaPlusBoost, using volumetric features as a natural extension of haar like features into the temporal domain. Bowden, Ong, Kadir et al. published several articles [7, 38, 32] on hand detection, hand shape and sign language recognition, focusing on boosted classifiers, minimal training and large vocabulary generalization. In [10] the same researchers present an unsupervised method to recognise signs from subtitles.

2.3. Neural Networks

Neural Networks have been used extensively within Sign Language recognition. Vamplew and Adams in [48] introduced SLARTI, a modular architecture consisting of multiple feature-recognition neural networks and a nearest-neighbour classifier. Yang et al. in [59] presented a feature extraction algorithm based on multiscale segmentation and used the resulting trajectories to classify gestures based on a time-delay neural network. Shanableh et al. [42] employed k-nearest neighbor and Bayesian classifier to recognize isolated Arabic Sign Language, while Yang [61] added fuzziness to a BP network. Wang and Gao [54], considering the speed and performance of isolated word recognition systems, presented a Semi-Continuous Dynamic Gaussian Mixture Model recognition technique.

2.4. Discussion on SL automatic recognition

While HMMs are widely used in Sign Language and gesture recognition, such an approach has been proven inadequate in many cases [39]. In HMM training, each word model is estimated separately, using the corresponding labeled training observation sequences, without considering data that are close, but do not match the patterns exactly (i.e. other models with similar behavior). Moreover, arbitrarily or experimentally defined design parameters, such as the number of states, make HMMs unstable and sensitive to modifications of these parameters. Additionally, Dynamic Time Warping and HMMs are intrinsically related with each other, while based on own features. DTW searches for the best alignment path while the HMMs likelihood function sums the density along all possible alignment paths. DTW can provide a higher level of granularity in the movement path compared to HMMs. Lichtenauer et al. in [35] present an interesting combination DTW and HMMs with discriminative classifiers.

It should be mentioned that recognition rates of either isolated or continuous signing are merely an indicative and quite subjective criterion for method comparison due to the many different parameters and uncontrolled variables used. Furthermore, the use of different datasets by researchers makes the comparisons more difficult. Even in cases when the same corpora are used, such corpora are not in the public domain or they are published under some copyright restrictions.

3. The proposed scheme

The automatic spatiotemporal recognition scheme proposed here and tested on the Sign Language application domain is described in detail in the following. Section 3.1 introduces the overall approach while sections 3.2-3.5 and 3.6 discuss the modeling and classification process respectively. Intermediate transformations and novel algorithms, aiming to tackle challenges that are introduced by the spatiotemporal domain in general and SL in particular, are presented in sections 3.3 to 3.5. Finally, the scheme is validated through experimental results on a synthetic and a Greek Sign Language corpus of 118 representative lemmata, presented in section 4.

3.1. Feature extraction

Features required for training and testing the proposed approach are extracted based on the methodology described in [16]. The feature extraction process uses monocular vision input and Geodesic Active Regions models, enhanced with color and motion cues, which evolve to fit hand regions. The extracted feature set includes hand coordinates (hand trajectories), shape and region descriptors. Shape features are both boundary-based (Fourier descriptors, Curvature features) and region-based (Moments, Moment-Based Features). Region based features include the shape area, its eccentricity, compactness, orientation and the minor to major axis length ratio. These features are used to train classifiers, depicting position, direction and state transition, the outputs of which are fused through appropriate boosting until a final classification decision is reached. In case of an unlabeled sign instance, all trained models are tested against the instance and resulting probabilities are fused using weights calculated according to the isolated recognition rates, providing the final recognition outcome.

3.2. Modeling Sign Language spatial and temporal aspects

In Sign Language analysis the input space can be modeled as a cube surrounding the signer in which all signs take place. In the following, this space is modeled using a Self Organizing Map, for each hand. The trained SOM is using a representative sample of the hand positions during signing. We utilize SOM as a clustering tool to derive a more abstract representation of the signing space and define neighboring relations between the map's nodes based on training. Neurons are allowed to alter their weights representing similarities in the map. Learning refers to both weights and node neighboring relations. This neighboring relation consists a crucial characteristic of the overall training and classification process which is the basis of the adaptive nature of the overall approach.

Hand coordinates are first normalized with respect to size and position (head diagonal and center respectively), independently of the sign class they belong to and the order during sign performance. Then they are used to train an hexagonal, two-dimensional grid SOM, through batch mode learning. Normalization also ensures that the input information stream is invariant to both the position of the signer in front of the camera and to the anatomical individuality of the signer. SOM training is performed only once and not individually for every class. This feature is quite advantageous because it reduces the training time, the required storage resources and adds to the system design's simplicity. Assuming that an adequate number of sign instances have been used to train the SOM to represent the signing space, no additional training is required whenever a new class is introduced to the vocabulary.

The Unified distance matrix, or U-matrix, is one of the most popular methods of displaying SOMs and visualizing the distance between adjacent units in the map. When the samples are not similar, the distance between the corresponding map units is shown on the U-matrix display with warm colors. Trained SOMs U-matrices for the right and left hand are shown in Figure 1. The U-matrix for the left hand has been mirrored, in an attempt to make the representation more intuitive, since hand coordinates are relative to the head position in the specified frame. It is worth noticing that the signing area for each hand has been uniformly mapped, especially in regions that the appearance of the hand is frequent and coincide with areas of great uniformity (blue areas).

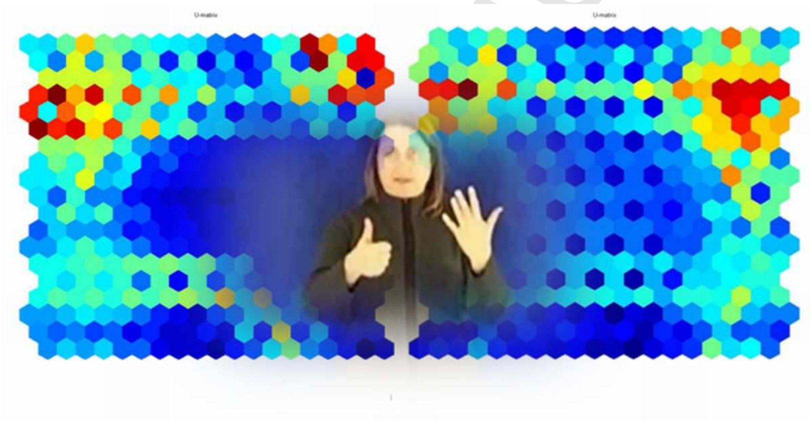

Figure 1: Signing space as modeled by SOMs for the GSLC described in section 4.2. U-matrices for right and left hand are illustrated.

3.3. Transformations and transitions

With the signing space modeled, each hand coordinate is assigned to a respective Best Matching Unit (BMU) on the map, transforming a sign instance G to a sequence of map units. The transformation is defined as:

$$T(G) = (u_1, u_2, \ldots, u_l) : u_i = W_r(BMU(x_i, y_i)) \tag{1}$$

where W_r is a median filter over a window of map units with length r. G is transformed to a new space model $G^{'} = N(T(G))$, N being a function that removes consecutive u values that are equal to each other. An additional transformation $OF(G) = \{v_1, v_2, ..., v_{l-1}\}$ is also adopted, based on the optical flow sequence of each sign, v denoting direction vectors defined by consecutive sign trajectory points. Vectors $OF(G)$ are first quantized, in order to provide distinct direction states, and then smoothed using a median filter, as in the position counterpart, resulting in transformation $G^{''} = N(OF(G))$.

In the proposed scheme, each sign instance G contributes to the training of two sets of Markov Models M^{som} and M^{of} for position and direction modeling respectively. The states of the Markov models belong to SOM states u for M^{som} and quantized direction vectors v for M^{of}. These result in calculation of transition probabilities and of respective initial state probabilities vectors (π^{som} and π^{of}).

During training, the Markov transition probability matrix depends additionally on the neighboring relation between states. Let us assume that an actual transition between u_i and u_j occurs for a sign instance used for training. Then, a number of transitions synapses are created from u_i to u_j, as would be the obvious case, but additionally to all the neighbors of u_j. Figure 2 illustrates this process. Green colored node denotes u_i, red one u_j and orange ones neighbors of u_j. The weights of these synapses, which finally affect the transition probability, are proportional to the neighboring associations. Thus for the actual transition $u_i \rightarrow u_j$ the transition matrix is updated using the following equation:

$$T_{tr}(i,x) = T_{tr}(i,x) + NF_{u_j}(x) \tag{2}$$

T_{tr} is the transition matrix for the respective Markov model, NF the SOM neighboring function and x the neighbors of u_j. For the implementation of the SOM, SOMTOOLBOX [47] was adopted, which features a som_neighf function that returns a MxM matrix containing neighborhood function values between all map units. Initially, Euclidean inter-unit distances between the SOM units (U_d) of the map grid are calculated. A gaussian function $e^{-U_d/(2*radius)}$ is then applied to these distances (U_d). T_{tr} is updated as in 2 and is finally normalized so that row values sum to 1 for all row values corresponding to states that are included in the training instances; self transitions are excluded by assigning a value of zero to the respective cells of T_{tr}.

This procedure ensures that if the state sequence in the testing dataset contains transitions that have not appeared in the training sequences, they will still be included in the search for the best scoring S_i (see section 3.6, Equation 5) since they will have a non zero transition probability. This enhances architecture's robustness against user variability and minimizes the requirement for extensive training sets.

3.4. A modified Levenshtein distance

An additional model that is used in the proposed method for each sign class is the Generalized Median M_g. It is defined as a sequence that consists of a combination of set's symbols that minimizes the sum of distances to every string of the set [33]. In order to define M_g we employ a modified version of the Levenshtein distance L as the distance metric, $M_g = \arg\min_g \sum L(g, G')$,

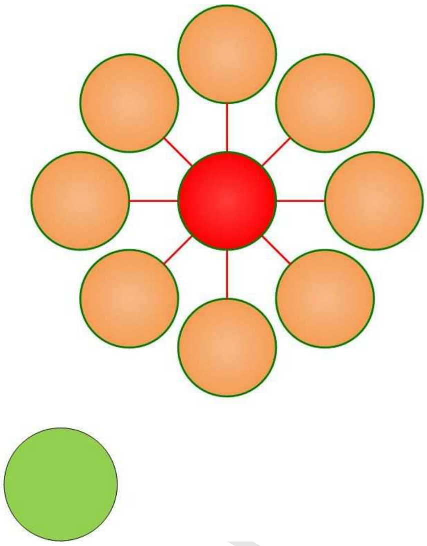

Figure 2: Transitions based on neighboring relations. Transitions from the Green colored node (source node u_i) to the Orange nodes, who are neighbors of the Red node (target node u_j) will now have a non-zero probability in T_{tr}.

g being a gesture belonging to the training set for a specific sign class. This modified version of the Levenshtein distance L is also employed during classification, as will be discussed in section 3.6.

This variation of the Levenshtein distance calculation algorithm (Algorithm 2) incorporates neighboring relations between SOM nodes. The latter are the symbols of the two sequences in question and are used to assign a cost for each symbol substitution and is also employed during the classification stage. The original cost assignment algorithm is shown in Algorithm 1. It takes place during the comparison of each symbol $(str1[i], str2[j])$ of sequences $str1$ and $str2$ in order to decide which action has minimal cost. Finally, the Levenshtein distance is defined by the bottom-right cell of the constructed matrix d.

Algorithm 1 Original Levenshtein distance cost calculation

```
if $str1[i] = str2[j]$ then
    $cost := 0$
else
    $cost := 1$
end if
$d[i,j] := minimum(\ d[i-1,j]+1, //deletion\ d[i,j-1]+1, //insertion$
$d[i-1,j-1]+cost//substitution\ )$
```

Algorithm 1 does not, however, take into account how similar symbols $str1[i], str2[j]$ are, when such a similarity measurement actually exists in the symbol set. In the case of the SOM, which is trained to map hand coordinates, this similarity exists between the nodes that are the actual symbols of the set constituting each sequence. The cost for the substitution action should be lower, in such cases that the two symbols participating in the substitution are close in terms of the SOM neighboring function; it should be higher if the two nodes are not neighbors. In our approach Algorithm 2 is proposed in order to tackle issues similar to the one described above:

According to this algorithm the cost for a substitution is proportional to the neighboring relation of the two participating nodes (symbols in the sequences). A similar modification can be applied to the Damerau-Levenshtein distance metric, for the case of transposition of two symbols; however, this is a case quite rare in sign language analysis. It should be mentioned that the neighboring relation of the nodes influences two other actions in the Levenshtein distance calculation: deletion and insertion. The cost of each action is

Algorithm 2 Modified Levenshtein distance cost calculation for symbol substitution

```
if $str1[i] = str2[j]$ then
    $cost := 0$
else
    $cost := 1 - NF_{str1[i]}(str2[j])$
end if
```

equal to the neighboring relationship between the i^{th} node, being inserted or deleted, and the preceding and successive nodes, $i-1$ and $i+1$ respectively:

$$cost := 1 - \frac{NF_i(i-1) + NF_i(i+1)}{2} \tag{3}$$

The mean Levenshtein distance (L_m) between the members of the set and M_g is also calculated. This constitutes a way to measure variation within the members of the set that will be used accordingly in the classification stage (section 3.6) and is defined as:

$$L_m = \frac{\sum_{i=1}^{n} L(G'_i, M_g)}{n} \tag{4}$$

n being is the number of training vectors for a specific class.

3.5. Modeling the handshape

Modeling the handshape during sign language analysis can be quite complex, especially when the corresponding features are derived through monocular visual processing. This feature set refers to three sifferent aspects: palm orientation, fingertip direction and finger joints arrangement. The set can however describe the fusion of all these three aspects and not each one separately. It is, therefore, emphasized that this approach does not refer solely to the arrangement of fingers, as is the case with most of the approaches based on motion capture.

To tackle this problem, in our approach, we train continuous (mixtures of three Gaussians), left-to-right Hidden Markov Models based on features describing the handshape [46], which are: features describing the area of the extracted hand (HMM^{hs_a}), Fourier descriptors (HMM^{hs_f}), moments (HMM^{hs_m}) and coefficients of the Curvature Cepstrum (HMM^{hs_c}), as can be shown in Figure 3. These are utilized to model different combinations of finger joint angles, palm orientation and fingertip direction.

Every HMM trained with different feature sets has a different number of states. The number of states per aspect specific HMM was defined experimentally. Indicative state numbers are: Position 7, RegionBased 5, FourierDescriptors 3, Moments 5, CepstrumCoeff 3. Again, this enforces the argument about HMMs inadequacy in terms of arbitrarily defined parameters.

3.6. Fusion of classifiers

During classification, the models presented earlier compute corresponding participation probabilities. The latter are then fused at decision level, following a weak classifier boosting approach. Let us consider a test sign pattern g, belonging to an unknown class, transformed to g' and g'' as described earlier in Section 3.3.

Based on the M^{som} and M^{of}, the probabilities $P(g^{'})$ and $P(g^{''})$ of g

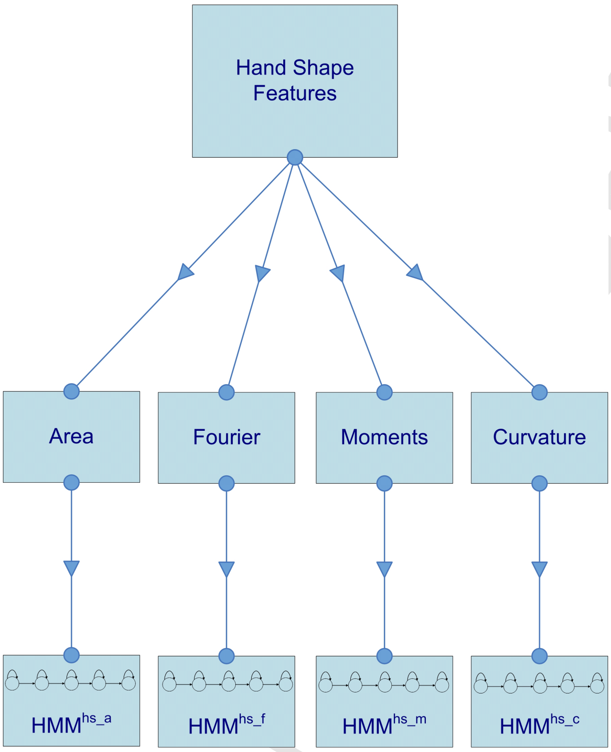

Figure 3: Hidden Markov Models based on features describing the handshape

belonging to class j are calculated as shown below:

$$\begin{aligned} P(g^{'}|M_j^{som}) &= \frac{\prod\limits_i S_i^{som}}{|g^{'}|} \\ P(g^{''}|M_j^{of}) &= \frac{\prod\limits_i S_i^{of}}{|g^{''}|} \end{aligned} \tag{5}$$

where S_i are values representing an evaluation factor for each u or v value with respect to M_j^{som} and M_j^{of} Markov models. Let us examine the case of S_i^{som} where a search is performed across all units of the map, for a unit that combines a considerable transition probability from the previous state with a relatively small distance onto the map grid from the current state

(Equation 6). Intuitively, this search corresponds to the greedy aspect of the classification algorithm since it constantly seeks the locally optimal, in terms of proximity and transition probability, choice at each stage.

$$S_i^{som} = \max_z (NF_{u_i}^{som}(z) P(z|u_{i-i}, M_j^{som})) \tag{6}$$

where NF^{som} is derived in SOM training. NF^{of}, used in S_i^{of} of Equation 5 is arbitrarily defined: a value of 1/2 is given to the closest direction neighbor and 1/4 for the second closest neighbor in both directions. The winner unit derived from the search is used as the previous state, in the next step.

According to the modified Levenshtein distance, an additional similarity measurement M_{g_j} is introduced which corresponds to the distance of g' to the generalized median of each class:

$$P(g^{'}|M_{g_j}) = \frac{L_{m_j}}{L(g^{'}, M_{g_j})} \tag{7}$$

This can tackle the partial sign problem, where if the whole of a sign instance is the starting part of a sign class then it would get high ranking using just M^{som} and M^{of}.

Classification fusion is implemented by a weighted summation of the individual modalities of the handshape, as shown in Equation 8. a, f, m, c correspond to different handshape feature sets, as shown in Figure 3. For example HS_g^a corresponds to area feature vector for sign instance g and $HMM_j^{hs_a}$ corresponds to the HMM trained with area features for sign class j.

$$P(HS_g|HMM_j) = \sum_{q \in [a,f,m,c]} w_q P(HS_g^q|HMM_j^{hs_q}) \tag{8}$$

The overall winner class is determined by the sum of the evaluation of Equation 9 for both dominant and non-dominant hands. Each of the four terms in the equation represents the participation of each separate modality to classification of sign g in j ($g \in j$), one of the classes in the dictionary. Each of the information channels is assigned a weight ($w_{som}, w_{of}, w_L, w_a, w_f, w_m, w_c$) which derives from the recognition rate each channel achieves individually. The notation of the weights is as following: som and of for the position and direction Markov models respectively, L for the Levenshtein distance with the Generalized Median and a, f, m, c for area, fourier, moments and cepstrum HMMs respectively. The non-dominant hand is also assigned a weight (< 1) since its participation in the decision making process is inferior to that of the dominant hand.

$$\arg\max_j (w_{som} P(g^{'}|M_j^{som}) + w_{of} P(g^{''}|M_j^{of}) + w_L P(g^{'}|M_{g_j}) + P(HS_g|HMM_j)) \tag{9}$$

3.6.1. Error Analysis

SOM uses several error measurement metrics to determine the mapping quality. The simpler and most commonly evaluated measurement is the cumulative quantization error e, which is the average distance D between N input samples d_i and their respective best matching unit BMU.

$$e = \frac{E}{N} : E = \sum_{i=1}^{N} D(d_i, bmu_i) \tag{10}$$

An alternative SOM error measurement, including the neighborhood information, is distortion, defined in Equation 11, in terms of the neighborhood function $h(bmu(i), j)$ and prototype-data vector distance $m_j - x(i)$. In case of a fixed neighborhood and discrete data, the distortion measure can be interpreted as the energy function of the SOM which is minimized approximately. [57] proposes another entropy based SOM error metric EH defined in 12 k is the total number of input samples and M_{total} is the total number of SOM units.

$$D^{SOM} = \sum_i \sum_j h(bmu(i), j) \parallel m_j - x(i) \parallel^2 \tag{11}$$

$$\begin{aligned} EH &= -\sum_{i=1}^{M_{total}} e_i \log e_i \\ E_n &= \sum_{j=1}^{k} distance(d_j, m_n) \\ e_n &= \frac{E_n}{E} \end{aligned} \tag{12}$$

To perform an error analysis of the proposed system, let us focus on the SOM decoding stage, and particularly on evaluating $P(g^{'}|M_j^{som}$ and $\prod_i S_i^{som}$. Let us investigate the effect of a random error $\delta x, \delta y$, of the hand point x, y in trajectory g, on the evaluation of Equation 6. The trajectory point is now $x + \delta x, y + \delta y$. a) If $\delta x, \delta y$ is small so that $BMU(x, y) = BMU(x + \delta x, y + \delta y)$, no error is introduced, and thus propagated, in the decoding stage since it is absorbed by the SOM. b) If $\delta x, \delta y$ are large, then $u^{'} = BMU(x + \delta x, y + \delta y) \neq BMU(x, y) = u_i$. Then S_i changes and $\delta S_i^{SOM} \approx NF_{u_i}^{som}(u^{'})$. Since, however, $u^{'}$ constitutes the new u_i in the next transition, the error is not propagated to the next steps of the recognition process.

The error analysis focuses on the SOM since the latter comprises the main contribution of the proposed architecture. HMM aspects, as error propagation, is not discussed throughout this article, since we do not present a novel HMM methodology. Any errors produced by the HMM module will introduce a respective, weighted error in the following fusion procedure.

4. Experimental Results

In order to validate the proposed mechanism we performed experiments on two datasets: a synthetic one containing only 2D spatial information of one hand and an actual Greek Sign Language corpus described in [17].

4.1. Synthetic corpus

This dataset consists of 30 gestures and 10 repetitions each. The set of coordinates formed a dataset containing categories varying in gesture complexity. For the synthetic dataset [9], experiments were conducted in order to evaluate the recognition performance of the proposed method based only on the position of one hand. Using all instances, for both training and testing phases of the system, in an attempt to validate the system's learning capabilities, resulted in 100% recognition percentages. For an evaluation of the generalization capabilities of the proposed method, another experiment was executed using the 10-fold cross validation strategy. In this case the average recognition rate was 93%. In order to compare the results of our system with the most commonly used approach in the literature we implemented a HMM based classifier. We trained one HMM per class. We used continuous left-to-right models and a mixture of 3 Gaussian probability density functions. During classification an instance was tested against all models and the one with the highest log-likelihood value was selected as the winner resulting an average recognition rate of 86,36%.

4.2. Greek Sign Language Corpus

The corpus used for the second set of experimentation was the Greek Sign Language Corpus (GSLC). All aspects of the corpus including corpus design, content definition, recording and quality control, annotation, etc. are described in detail by Efthimiou and Fotinea in [17]. From the GSLC dataset, 3 native signers were selected performing 3 repetitions of 118 representative lemmata of the Greek Sign Language under controlled recording conditions performing.

Initially, we have tested the proposed architecture against standard left-to-right continuous HMMs with the number of states defined experimentally so as to maximize the recognition rate. It is worth noting that the proposed approach does not require such an arbitrary design decision, such as the experimentally defined number of states in the HMM approach, and the topology and transition matrix of the Markov models is determined automatically without the need for manual, trial and error processes. The GSLC dataset is by far more complicated than the first synthetic dataset, described in section 4.1, since many of the signs differ only in the dominant handshape and have the same spatial and movement characteristics. The recognition rates for the two approaches can be seen in table 2.

The contribution of each stream in the final recognition rate can be analyzed as depicted in Table 3, while the handshape's recognition rate analysis using HMMs for each of the streams is shown at table 5. Additionally, table 4 demonstrates recognition rates on a dataset consisting a selection (over 90%)

Feature set	HMM	SOMM
Dominant Hand	61.12	73.41
Both Hands + direction	79.18	91.10

Table 2: Position based recognition rates

of the signs in GSLC. Table 2 shows the improvement caused by the incorporation of both hands and direction features compared to single handed, position based features. Symbol set comparison (Levenshtein) is used both for single and two handed recognition. The improvement due to the incorporation of Levenshtein is illustrated in Table 3 where the contribution of the novel symbol set distance metric is evident by comparing respective rows (second and third).

Feature set	Recognition rate
Position (Dominant hand)	73.41
Position (Both hands)	83.30
Position (Both hands) + Direction + Levenshtein	91.10
Position (Both hands) + Direction + Levenshtein + Handshape	97.80

Table 3: Analysis of recognition rates for different feature sets

During an informal evaluation of the dataset in terms of how easily each class would be recognized, two reviewers, one from the Greek deaf community and the other being one of the authors of the article who had a good understanding of the functionality of the proposed approach, inspected all the repetitions of the performed signs by watching the recorded video section and plotting the extracted hand positions by the feature extraction process. Consequently, each sign is assigned a rating depicting the recognition difficulty caused by either signer inconsistent performance or by errors introduced by the feature extraction process. It is worth noting that the 12 signs that performed worst in terms of automatic recognition rate belonged to the top 15 of the rating given manually by the reviewers. The respective recognition rates on the reduced dataset of 106 (118-12) are shown in table 4.

Feature set	Recognition rate
Position (Dominant hand)	74.77
Position (Both hands)	88.10
Position (Both hands) + Direction + Levenshtein	95.05
Position (Both hands) + Direction + Levenshtein + Handshape	99.54

Table 4: Performance on a subset of GSLC

Feature set	Recognition rate
Area	47.44
Fourier	36.67
Moments	36.82
Curvature	26.35
Fusion	55.1

Table 5: Analysis of recognition rate based on handshape features

Furthermore, we are willing to test how the proposed fusion process performs against popular HMM approaches such as Multi-Stream, Parallel and

Product variations designed to cope with multimodality and information from multiple streams. The results for these HMM variations as well as all of the experiments and results dealing with HMMs on the GSLC originate from [46]. SOMM's superiority in terms of efficiency, especially in demanding problems such as signer-independent sign language recognition in GSLC, is based on its adaptability both during training, due to the incorporation of the neighboring characteristic as discussed in section 3.3, and during the decoding stage where the classification algorithm, presented in 3.6, constantly seeks local maxima, in terms of proximity and transition probability.

Scheme	Recognition rate
Multi-Stream HMM	92.27
Parallel HMM	92.45
Product HMM	93.64
SOMM	97.80

Table 6: Fusion Performance

All of the above experiments were performed using Matlab and SOM-TOOLBOX [47] on a regular PC (2GHz Dual Core, 3GB RAM), using the leave-one-out cross-validation method and the processing time required for each step is shown in table 7. SOM size was defined taking into account problem complexity, classification performance, processing cost and spatial quantization and resolution requirements based on linguistic signing space modeling, resulting in a 10x10 and 20x20 SOM for the synthetic and the GSLC corpus respectively. Modalities weights were assigned as described in section 3.6. More specifically, $w_{som} = 0.83$ for the GSL corpus, and $w_{som} = 0.88$ for the selected GSLC subset, denoting the position information channel importance according to unimodal recognition results (Table 3 and 4 respectively). SOM training is the most demanding process in terms of processing time, but this process is only performed once regardless of the number of classes. The decoding stage varies depending on the sequence length but the average was 1.2 msec per instance per class, a performance which establishes the overall architecture suitable for real time applications. The proposed architecture is proven to be significantly faster than other dominant approaches as can be seen in table 8. It is worth mentioning that Product HMMs, which proves to be the most effective HMM variant (table 6), requires 15 times greater processing time for decoding.

Process	Right	Left	Total
SOM training	5.6151	3.2025	8.8176
Position Models	1.3345	2.0800	3.4145
Direction Models	1.1311	0.9159	2.0470
Decoding	0.0655	0.0803	0.1458

Table 7: Required time for training and classification (average times in seconds)

5. Conclusions and future work

Current work proposes an architecture for solving spatiotemporal problems and validates it by applying the proposed scheme to an extremely challenging problem of automatic Sign Language recognition. Dynamics, spatiotemporal variation and random errors and noise in the input stream are

Classifier	Training	Testing
SOMM	14.279	0.145
Multistream HMM	28.416	0.870
Product HMM	53.938	2.280

Table 8: Training and classification times for SOMM, Multistream HMM and Product HMM

tackled by a greedy algorithm constantly seeking the locally optimal choice at each stage and converging to a global solution and by incorporating the neighboring characteristic amongst the models' states both in the learning as well as the classification process. The proposed scheme is validated both theoretically, by performing an error propagation study, and experimentally on Greek Sign Language recognition proving the architecture's superiority, in terms of classification performance and computational cost, against popular techniques such as Hidden Markov Models and variations. SOMM's superiority in terms of efficiency, adaptability and generalization is based on the incorporation of proximity, according to the SOM representation, both in training and in classification. Regarding design issues, the self organizing feature of the proposed architecture eliminates the need for arbitrarily or experimentally defined initialization parameters, such as defining the number of HMM states which influences significantly their performance and hinder their generalization and adaptability. Modality fusion is performed at decision level, in Boosting-like, relaxed but none the less targeted manner based on weak classifiers who are suitable for tackling a particular aspect of the problem.

Concerning future directions of the proposed research work these would include continuous input and inter-segment analysis. Isolated (sign level) recognition, which is the application domain of the proposed architecture, can be extended to continuous (sentence level) recognition by incorporating a temporal segmentation module that automatically detects sign boundaries [37, 60, 58]. Boundary detection is performed in various ways:

- by detecting local minima in hand velocity or glove finger flexure values
- by detecting maxima in motion trajectory angle derivative or ratio between minimum acceleration and maximum velocity
- HMMs trained for implicit sign segmentation or to model transitions and epenthesis and matching probability drop.

Such an approach would provide segment boundaries and could be considered a preprocessing step of the continuous input stream. Finally, adding a layer of domain knowledge (e.g. linguistic knowledge in a Sign Language sentence) in order to assist intra-segment recognition, consists a research direction worth investigating. Such a domain knowledge layer would enhance the architecture's assertion in continuous input. Continuous recognition also introduces coarticulation phenomena, where each sign, of the sentence, is affected by the preceding and the subsequent sign and Linguistic NLP knowledge is certainly needed.

References

[1] U. von Agris, J. Zieren, U. Canzler, B. Bauer, K.F. Kraiss, Recent developments in visual sign language recognition, Universal Access in

the Information Society 6 (2008) 323–362.

[2] K. Assaleh, M. Al Rousan, Recognition of arabic sign language alphabet using polynomial classifiers, Journal of Applied Social Psychology (2005) 2136–2145.

[3] M. Assan, K. Grobel, Video-based sign language recognition using hidden markov models, in: Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction, Springer-Verlag, London, UK, 1998, pp. 97–109.

[4] B. Bauer, H. Hienz, Relevant features for video-based continuous sign language recognition, Automatic Face and Gesture Recognition 00 (2000) 440.

[5] B. Bauer, H. Hienz, K.F. Kraiss, Video-based continuous sign language recognition using statistical methods, International Conference on Pattern Recognition 02 (2000) 2463.

[6] B. Bauer, K.F. Kraiss, Video-based sign recognition using self-organizing subunits, International Conference on Pattern Recognition (2002) 24–34.

[7] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, M. Brady, A linguistic feature vector for the visual interpretation of sign language, in: European Conference on Computer Vision, ECCV, Springer, 2004, pp. 390–401.

[8] H. Brashear, T. Starner, P. Lukowicz, H. Junker, Using multiple sensors

for mobile sign language recognition, in: Wearable Computers, Seventh IEEE International Symposium on, pp. 45–52.

[9] G. Caridakis, K. Karpouzis, A. Drosopoulos, S. Kollias, Somm: Self organizing markov map for gesture recognition, Pattern Recognition Letters 31 (2010) 52–59.

[10] H. Cooper, R. Bowden, Learning signs from subtitles: A weakly supervised approach to sign language recognition, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 2568 –2574.

[11] H.M. Cooper, R. Bowden, Large lexicon detection of sign language, in: IEEE Workshop Human Computer Interaction, ICCV2007, volume 4796, pp. 88–97.

[12] H.M. Cooper, R. Bowden, Sign language recognition using boosted volumetric features, in: IAPR Conference on Machine Vision Applications (IAPR MVA 2007), May 16-18, 2007, Tokyo, Japan. pp 359-362.

[13] G. Cortes, L. Garcia, C. Benitez, J.C. Segura, Hmm-based continuous sign language recognition using a fast optical flow parameterization of visual information, in: INTERSPEECH-2006, paper 1543.

[14] Y. Cui, J. Weng, Appearance-based hand sign recognition from intensity image sequences, Computer Vision and Image Understanding: CVIU 78 (2000) 157–176.

[15] K. Derpanis, R. Wildes, J. Tsotsos, Hand Gesture Recognition within a

Linguistics-Based Framework, Computer Vision, ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14 (2004).

[16] O. Diamanti, P. Maragos, Geodesic Active Regions For Segmentation and Tracking of Human Gestures in Sign Language Videos, in: 15th IEEE International Conference on Image Processing, pp. 1096–1099.

[17] E. Efthimiou, S.E. Fotinea, GSLC: Creation and annotation of a greek sign language corpus for HCI, in: Universal Acess in Human Computer Interaction. Coping with Diversity, pp. 657–666.

[18] G. Fang, W. Gao, J. Ma, Signer-independent sign language recognition based on sofm/hmm, Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on (2001) 90–95.

[19] G. Fang, W. Gao, D. Zhao, Large vocabulary sign language recognition based on fuzzy decision trees, Systems, Man and Cybernetics, Part A, IEEE Transactions on 34 (2004) 305–314.

[20] G. Fang, X. Gao, W. Gao, Y. Chen, A novel approach to automatically extracting basic units from chinese sign language, Pattern Recognition (2004) 454–457.

[21] K. Fujimura, X. Liu, Sign recognition using depth image streams, Automatic Face and Gesture Recognition, 7th International Conference on (2006) 381–386.

[22] W. Gao, G. Fang, D. Zhao, Y. Chen, A chinese sign language recognition system based on SOFM/SRN/HMM, Pattern Recognition 37 (2004) 2389–2402.

[23] W. Gao, G. Fang, D. Zhao, Y. Chen, Transition movement models for large vocabulary continuous sign language recognition, Automatic Face and Gesture Recognition (2004) 553–558.

[24] W. Gao, J. Ma, J. Wu, C. Wang, Sign language recognition based on hmm/ann/dp, nternational Journal on Pattern Recognition and Artificial Intelligence vol. 14, no. 5 (2000) 587–602.

[25] K. Grobel, M. Assan, Isolated sign language recognition using hidden markov models, Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'., 1997 IEEE International Conference on 1 (1997) 162–167 vol.1.

[26] J. Hernandez-Rebollar, N. Kyriakopoulos, R. Lindeman, A new instrumented approach for translating american sign language into sound and text, Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on (2004) 547–552.

[27] H. Hienz, B. Bauer, K. Kraiss, Hmm-based continuous sign language recognition using stochastic grammars, in: Gesture Workshop, p. 185.

[28] C.L. Huang, W.Y. Huang, Sign language recognition using model-based tracking and a 3D Hopfield neural network, Machine Vision and Applications Volume 10, Numbers 5-6 / April, 1998 (1998) 292–307.

[29] K. Imagawa, H. Matsuo, R. Taniguchi, D. Arita, S. Lu, S. Igi, Recognition of local features for camera-based sign language recognition system, International Conference on Pattern Recognition, volume 04, IEEE Computer Society, Los Alamitos, CA, USA, 2000, p. 4849.

[30] I. Infantino, R. Rizzo, S. Gaglio, A framework for sign language sentence recognition by commonsense context, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37 (2007) 1034–1039.

[31] F. Jiang, H. Yao, G. Yao, Multilayer architecture in sign language recognition system, in: 6th international conference on Multimodal interfaces.

[32] T. Kadir, R. Bowden, E. Ong, A. Zisserman, Minimal training, large lexicon, unconstrained sign language recognition, British Machine Vision Conference (2004).

[33] Y. Lee, S. Kassam, Generalized median filtering and related nonlinear filtering techniques, Acoustics, Speech and Signal Processing, IEEE Transactions on 33 (2003) 672–683.

[34] Y.H. Lee, C.Y. Tsai, Taiwan sign language (tsl) recognition based on 3d data and neural networks, Expert Systems with Applications (2007).

[35] J.F. Lichtenauer, E.A. Hendriks, M.J. Reinders, Sign language recognition by combining statistical DTW and independent classification, Pattern Analysis and Machine Intelligence, IEEE Transactions on 30 (2008) 2040–2046.

[36] J. Ma, W. Gao, R. Wang, A parallel multistream model for integration of sign language recognition and lip motion, in: ICMI, pp. 582–589.

[37] L. Morency, A. Quattoni, T. Darrell, Latent-dynamic discriminative models for continuous gesture recognition, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 1–8.

[38] E.J. Ong, R. Bowden, A boosted classifier tree for hand shape detection, Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on (2004) 889–894.

[39] S. Ong, S. Ranganath, Automatic sign language analysis: a survey and the future beyond lexical meaning, Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 873–891.

[40] V. Pashaloudi, K. Margaritis, A performance study of a recognition system for greek sign language alphabet letters, in: International Conference "Speech and Computer".

[41] H. Sagawa, M. Takeuchi, A method for recognizing a sequence of sign language words represented in a japanese sign language sentence, Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on (2000) 434–439.

[42] T. Shanableh, K. Assaleh, M. Al-Rousan, Spatio-temporal feature-extraction techniques for isolated gesture recognition in arabic sign language, Systems, Man, and Cybernetics, Part B, IEEE Transactions on 37 (2007) 641–650.

[43] T. Starner, J. Weaver, A. Pentland, Real-time american sign language recognition using desk and wearable computer based video, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 1371–1375.

[44] M.C. Su, A fuzzy rule-based approach to spatio-temporal hand gesture recognition, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 30 (2000) 276–281.

[45] N. Tanibata, N. Shimada, Y. Shirai, Extraction of hand features for recognition of sign language words, International Conference on Vision Interface, in: International Conference on Vision Interface, pp. 391–398.

[46] S. Theodorakis, Isolated Greek Sing Language Recognition using Hidden Markov Models, Master's thesis, School of Electrical and Computer Engineering of the National Technical University of Athens, 2008.

[47] S. Toolbox, J. Vesanto, Neural network tool for data mining: Som toolbox (2000).

[48] P. Vamplew, A. Adams, Recognition of sign language gestures using neural networks, Australian Journal of Intelligent Information Processing Systems 5 (1998) 94–102.

[49] C. Vogler, American Sign Language Recognition: Reducing the Complexity of the Task with Phoneme-Based Modeling and Parallel Hidden Markov Models, Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania, 2002.

[50] C. Vogler, D. Metaxas, Parallel hidden markov models for american sign language recognition, International Conference on Computer Vision 01 (1999) 116.

[51] C. Vogler, D.N. Metaxas, Handshapes and movements: Multiple-channel american sign language recognition, in: Gesture Workshop, pp. 247–258.

[52] C. Wang, W. Gao, S. Shan, An approach based on phonemes to large vocabulary chinese sign language recognition, Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, in: Automatic Face and Gesture Recognition, pp. 393–398.

[53] H. Wang, M.C. Leu, C. Oz, American sign language recognition using multi-dimensional hidden markov models, Journal of Information Science and Engineering 22, 5 (2006) 1109–1123.

[54] J. Wang, W. Gao, A fast sign word recognition method for chinese sign language, Lecture Notes in Computer Science, Advances in Multimodal Interfaces ICMI 2000 1948/2000 (2000) 599–606.

[55] Q. Wang, X. Chen, C. Wang, W. Gao, Sign language recognition from homography, Multimedia and Expo, 2006 IEEE International Conference on, in: IEEE International Conference on Multimedia and Expo, pp. 429–432.

[56] A. Wilson, A. Bobick, Parametric hidden markov models for gesture recognition, IEEE Trans. Pattern Analysis and Machine Intelligence 21(9) (1999).

[57] Y. Wu, M. Takatsuka, The Geodesic Self-Organizing Map and its error analysis, in: Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38, Australian Computer Society, Inc. Darlinghurst, Australia, Australia, pp. 343–351.

[58] H.D. Yang, S. Sclaroff, S.W. Lee, Sign language spotting with a threshold model based on conditional random fields, Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (2009) 1264 –1277.

[59] M.H. Yang, N. Ahuja, M. Tabb, Extraction of 2d motion trajectories and its application to hand gesture recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (2002) 1061–1074.

[60] R. Yang, S. Sarkar, B. Loeding, Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (2010) 462 –477.

[61] X. Yang, Study on sign language recognition fusion algorithm using fnn, in: B.y. Cao, G.j. Wang, S.z. Guo, S.l. Chen (Eds.), Fuzzy Information and Engineering 2010, volume 78 of *Advances in Soft Computing*, Springer Berlin, Heidelberg, 2010, pp. 617–626.

[62] M. Zahedi, D. Keysers, H. Ney, Appearance-based recognition of words in american sign language, in: Pattern Recognition and Image Analysis.

[63] C.X. Zhang, H.X. Yao, F. Jiang, D.B. Zhao, X.T. Sun, Multilayer method based on multi-resolution feature extracting and MVC dimension reducing method for sign language recognition, Machine Learning

and Cybernetics, 2005. Proceedings of 2005 International Conference on 7 (2005) 4452–4457 Vol. 7.

[64] L.G. Zhang, Y. Chen, G. Fang, X. Chen, W. Gao, A vision-based sign language recognition system using tied-mixture density hmm, International Conference on Machine Intelligence (2004) 198–204.

[65] J. Zieren, K. Kraiss, Robust person-independent visual sign language recognition, Pattern Recognition and Image Analysis (2005) 520–528.