symmetry

MDPI

Article

A Novel String Grammar Unsupervised Possibilistic C-Medians Algorithm for Sign Language Translation Systems

Atcharin Klomsae [1,2], Sansanee Auephanwiriyakul [1,3,*] and Nipon Theera-Umpon [3,4]

1 Computer Engineering Department, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand; atcharin.k@gmail.com
2 Graduate School, Chiang Mai University, Chiang Mai 50200, Thailand
3 Biomedical Engineering Institute, Chiang Mai University, Chiang Mai 50200, Thailand; nipon@ieee.org
4 Electrical Engineering Department, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand
* Correspondence: sansanee@ieee.org or sansanee@eng.cmu.ac.th; Tel.: +66-5394-2023

Received: 30 November 2017; Accepted: 14 December 2017; Published: 19 December 2017

Abstract: Sign language is a basic method for solving communication problems between deaf and hearing people. In order to communicate, deaf and hearing people normally use hand gestures, which include a combination of hand positioning, hand shapes, and hand movements. Thai Sign Language is the communication method for Thai hearing-impaired people. Our objective is to improve the dynamic Thai Sign Language translation method with a video captioning technique that does not require prior hand region detection and segmentation through using the Scale Invariant Feature Transform (SIFT) method and the String Grammar Unsupervised Possibilistic C-Medians (sgUPCMed) algorithm. This work is the first to propose the sgUPCMed algorithm to cope with the unsupervised generation of multiple prototypes in the possibilistic sense for string data. In our experiments, the Thai Sign Language data set (10 isolated sign language words) was collected from 25 subjects. The best average result within the constrained environment of the blind test data sets of signer-dependent cases was 89–91%, and the successful rate of signer semi-independent cases was 81–85%, on average. For the blind test data sets of signer-independent cases, the best average classification rate was 77–80%. The average result of the system without a constrained environment was around 62–80% for the signer-independent experiments. To show that the proposed algorithm can be implemented in other sign languages, the American sign language (RWTH-BOSTON-50) data set, which consists of 31 isolated American Sign Language words, is also used in the experiment. The system provides 88.56% and 91.35% results on the validation set alone, and for both the training and validation sets, respectively.

Keywords: string grammar unsupervised possibilistic C-medians; scale invariant feature Transform (SIFT); fuzzy K-nearest neighbour; sign language; big data

1. Introduction

Deaf persons are unable to discriminate speech through their ears, and therefore cannot use hearing for communication. Sign language is one communication method for deaf or hearing-impaired people that communicates information through hand gestures and other body actions. However, most hearing people cannot comprehend sign language, which can cause communication difficulties. To solve this issue, hand sign language translation may be able to help deaf or hearing-impaired people communicate with hearing people. There have been many research studies on hand sign translation in a variety of sign languages. However, in order to translate the hand sign, the system needs to

know both where the hands are, and what their movements are. To ease this problem, some research studies have utilised cyber-gloves (electronic gloves) to help detect the hand positions [1–12]. Other research works have used colour-coded gloves instead of cyber-gloves [13–17]. However, signers in these two cases had to wear extra equipment in order for the system to work properly, which might not fit well with daily life. Hence, there has also been other research on freehand (no extra equipment) translation systems. Some of these involved pre-processing the image frames, which included hand segmentation [18–32]. Others only pre-processed the portion of the image with hand parts (only hands were captured in the video data set) [33–37]. Other works utilised motion sensors or Kinect [38–40] to capture hand movements. The remaining methods were visual-based, without any segmentation, and only used cameras [41–46]. Although some of the visual-based methods provided impressively correct classifications, as shown in Table 1, they still suffered from errors that were caused by similar hand movements that had different finger movements for different words, similar hand gestures for different words, etc.

The above-mentioned methods all suffer from extra equipment usage, pre-processing, segmentation, or a capturing device that might not be practical in daily life. In this paper, we improve the method for Thai sign language translation using Scale Invariant Feature Transform (SIFT) and String Grammar Unsupervised Possibilistic C-Medians (sgUPCMed). The String Grammar Unsupervised Possibilistic C-Medians (sgUPCMed) algorithm is a new string grammar clustering method that is introduced in this paper for the first time. Since fuzzy clustering is able to cluster overlapping data samples and deal with noise or outliers, this method might better cope with the problems outlined above. Moreover, our system does not require hand region detection or hand segmentation for sign language translation. The system only uses a camera to record the movement, without any extra sensors or equipment on the signers. The Scale Invariant Feature Transform method is used to match the test frame with symbols in the signature library. The String Grammar Unsupervised Possibilistic C-Medians (sgUPCMed) algorithm is used for prototype generation, while the fuzzy k-nearest neighbour is utilised as a classifier. Ten isolated Thai sign words are used in our experiments: “elder”, “grandfather”, “grandmother”, “gratitude”, “female”, “male”, “glad”, “thank you”, “understand”, and “miss”. The experiments are implemented within signer-dependent, signer semi-dependent, and signer-independent scenarios. The subjects used in the signature library collection for the SIFT algorithm and in the string grammar clustering algorithms for the generation of prototypes are utilised in a subject-dependent case, while the subjects that are only presented in the blind test data set are utilised in a signer-independent case. However, there are two types of signer semi-independent cases. One is when subjects are only in the signature library collection, and the other is when subjects are only in the prototype generation process.

The first experiments were implemented with constraints: subjects were asked to wear a black shirt with long sleeves and stand in front of a dark background. The best system that emerged from these constraints was then implemented on signers in the blind test data set without any constraints, i.e., they were asked to wear a short-sleeve shirt and stand in front of various natural backgrounds. We implemented our proposed system with the RWTH-BOSTON-50 data set, which consists of 31 isolated American Sign Language (ASL) words as well, in order to show the ability of the system with other sign languages. We also compare the results with those from the existing algorithms. The remainder of this paper is organised as follows. Section 2 explains our proposed system, along with a review of the SIFT method and the String Grammar Unsupervised Possibilistic C-Medians (sgUPCMed) algorithm. The experimental results are shown in Section 3, and finally, we draw the conclusion in Section 4.

Table 1. Experiment results from previous algorithms working without extra equipment. SIFT: Scale Invariant Feature Transform; ASL: American Sign Language.

Sign Language	# of Recognised Words	Data Set	Instrument Used	Mode	Pre-Process with Segmentation	# of Signers	Classification Rate (%)
Taiwan Sign Language [19]	15	Test data set	None: free hand	Signer-independent	Yes	N/A	91
Japanese Sign Language [20]	6	Test data set	None: free hand	Signer-independent	Yes	20	93.5
Taiwan Sign Language [21]	20	Test data set	None: free hand	Signer-dependent	Yes	20	93.5
American Sign Language (ASL) [22]	39	Test data set	None: free hand	Signer-dependent	Yes	1	95
Malaysian Sign Language (MSL) [26]	66 (gestures not words)	Combined training and validation set)	None: free hand	Signer-dependent	No	1	80
Amefican sign language (RWTH-BOSTON-50) [27]	30	Combined training and test data set	None: free hand	Combined signer-dependent and signer-independent	Yes	3	89.1
Indian Sign Language [28]	36 (gestures not words)	Test data set	None: free hand	Signer-dependent	Yes	N/A	91.11
Chinese Sign Language [29]	8 (gestures not words)	Test data set	Kinect	Signer-dependent	Yes	8	82.79
Bangla Sign Language [31]	40 (alphabet)	Test data set	None: free hand	Signer-dependent	Yes	N/A	95.90
Indian Sign Language [32]	24	Test data set	None: free hand	Signer-dependent	Yes	N/A	90
Thai Sign Language (finger spelling) [33]	15	Test data set	None: free hand	Signer-dependent	Yes	5	91.20
Thai Sign Language (finger spelling) [34]	49	Test data set	None: free hand	Signer-dependent	Yes	2	72
Hand Gesture [35]	10	Test data set	None: free hand	Signer-dependent	Yes	15	97.62
Arabic Sign Language (ArSL) [36]	28	Test data set	None: free hand	Signer-dependent	Yes	N/A	93.21
American sign language (ASL) [37]	37 (gestures not words)	Test data set	None: free hand	Signer-dependent	Yes	5	94.32
Arabic Sign Language [41]	30	Test data set	None: free hand	Signer-dependent	Yes	8	97.4
				Signer-independent		10	94.2

Table 1. *Cont.*

Sign Language	# of Recognised Words	Data Set	Instrument Used	Mode	Pre-Process with Segmentation	# of Signers	Classification Rate (%)
Thai Sign Language Finger spelling words (with SIFT) [42]	10	Test data set	None: free hand	Signer-dependent	No	2	81.64 (on average)
			None: free hand	Signer-independent		2	33 (on average)
Thai sign language (with with Hidden Markov Model) [43]	10	Validation set	None: free hand	Signer-dependent	No	5	86–95 (on average)
			None: free hand	Signer semi-dependent		10	80 (on average)
			None: free hand	Signer-independent		5	75–76 (on average)
Arabic Sign Language [44]	23	Test data set	None: free hand	Signer-dependent	Yes	3	99.80
American Sign Language (RWTH-BOSTON-50) [45]	15	Test data set	None: free hand	Signer-dependent	Yes	3	93.33
American Sign Langage (RWTH-BOSTON-50) [46]	50	Test data set	None: free hand	Signer-dependent	No	3	82.8

2. System Description

The overview of the proposed Thai Sign Language translation system is shown in Figure 1. There are three parts to the system: string representation, string grammar clustering, and string grammar classification. In order to create string representation for each video, we first needed to collect 31 hand gestures [47] from 10 Thai hand sign words. Since fingers shapes, positions, and hand information are needed in the recognition system, we asked five subjects to wear a black shirt with long sleeves and stand in front of a dark background for the hand gesture collection process. Each subject was asked to perform each hand sign several times, and their movements were recorded in the form of video files. Then, we manually selected representative frames (Rframes) of each video file, and created the signature library [43] from these Rframes as a part of the training process. Each manual Rframe selection only captured a portion of the hand, which measured 190 $\times$ 190 pixels. Please note that this hand image is called a keyframe for the sake of simplicity. For each subject, we had 730 keyframes, and there were 3650 keyframes in total. Examples of hand gestures and their corresponding numbers of keyframes in the signature library are shown in Figure 2.

In the recognising process, we first chose F image frames with approximately equal spacing from each video sequence to generate a string from the video file. For each image frame, we utilised the Scale Invariant Feature Transform (SIFT) method [48] to extract interesting points. Then, we created descriptors that matched those in the signature library. Next, we selected a symbol representing that image frame. Finally the whole symbol sequence representing video files was generated. To create a prototype of each Thai hand sign word, we utilised the String Grammar Unsupervised Possibilistic C-Medians (sgUPCMed) algorithm. For the classification process, we utilised a modified version of the fuzzy k-nearest neighbour (FKNN) algorithm [49] to find the right Thai hand sign word.

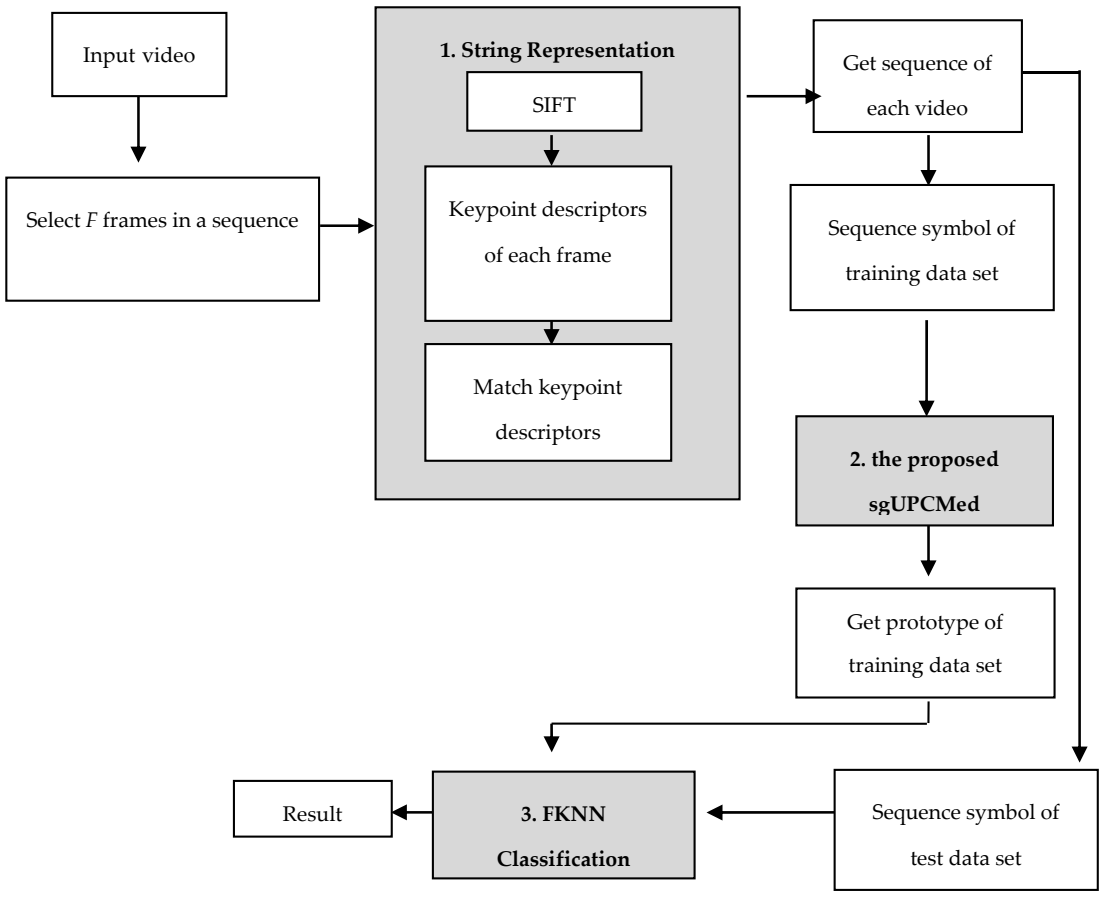

Figure 1. System overview of Thai sign language translation. FKNN: fuzzy k-nearest neighbour.

Now, let us briefly describe the SIFT algorithm [48]. This approach is used to detect and describe local features in training images called the keypoint. It is one of the most popular approaches in 2D image-features matching. The SIFT consists of four main steps to generate a keypoint: the detection of scale-space extrema, feature point localisation, orientation assignment, and feature point descriptor. In the detection of scale-space extrema, Gaussian scale-space is constructed. The input image is smoothed with the difference of Gaussian (DoG) function. Scale space is separated into octaves. To create a set of scale-space images, the initial image is repeatedly convolved with Gaussian masks on each octave. The difference between consecutive blur amounts is then output as one octave of the pyramid. The local extrema of difference of Gaussian in scale space are found by comparing an interest pixel to its 26 neighbours in 3×3 regions at the current and adjacent scales. The extremum is selected as a keypoint location if the value of the pixel is greater than or less than all of its neighbours. Now, the number of keypoints is less than the number of pixels. However, there are still plenty of points, and many of them are bad points. In the keypoint localisation step, it rejects points with low contrast and points with poor edges. Please note that the number of found keypoints for each image will depend on the characteristics of the image. To assign an orientation, the gradient histogram and a small point around the keypoint based on local image gradient directions are used. The magnitude and direction of the gradient are calculated for all of the pixels in a neighbouring area around the keypoint in the Gaussian-blurred image. A gradient histogram with 36 bins is created in this step. Any peak within 80% of the highest peak is used to create a keypoint with that orientation. Then, a 16×16 neighbourhood around the keypoint is found. It is divided into 16 sub-blocks, with the size of 4×4. For each sub-block, an eight-bin orientation histogram is created. Each keypoint descriptor is a vector of 128 dimensions that distinctively identifies the neighbourhood around the keypoint. Figure 3a shows the keypoints found in an example keyframe, and examples of keypoint descriptors of three hand gestures are shown in Figure 3b–d.

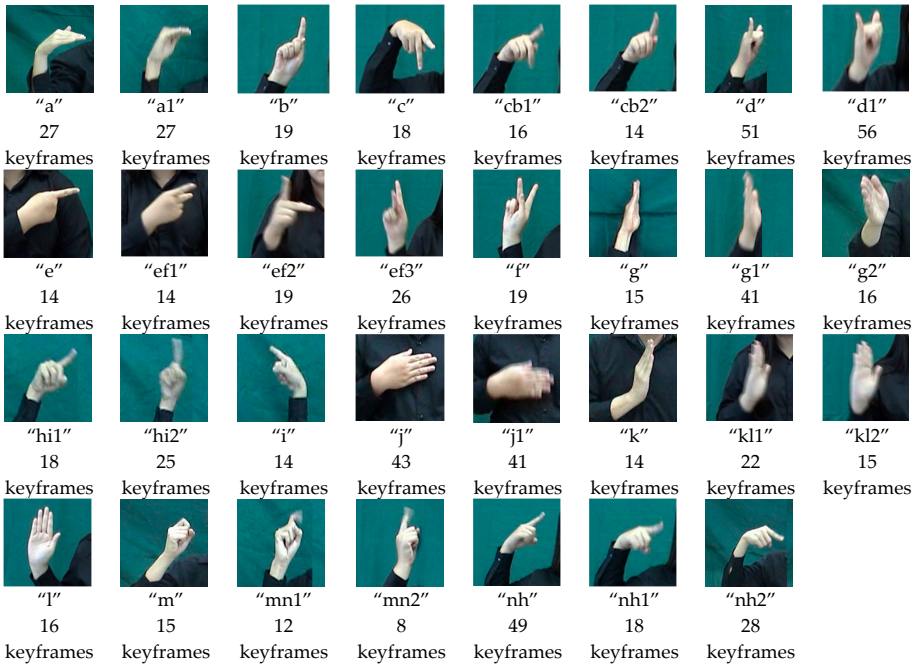

Figure 2. Examples of 31 hand gestures.

The keypoint matching process is performed by comparing the Euclidean distance between the two nearest neighbour keypoint descriptors in the signature library, and the current keypoint descriptor. If the ratio of the smallest distance to the second smallest distance is less than a given threshold, then the two keypoints match. Examples of keypoint matching are shown in Figure 3e–g.

Since the keyframes for each symbol in the signature library may have different numbers of keypoints, to identify the correct symbol for that image frame, the average number of matched keypoints per keyframe (*Avg_Match*) [43] of each symbol is computed as:

$$Avg\_Match = \frac{\text{Number of matched keypoints of the symbol}}{\text{Number of keyframes of the symbol}} \quad (1)$$

An example of this process is shown in Figure 4. We repeat these steps for F selected image frames of the video. Then, we obtained the sequence of symbols of each video, and then used this as the sequence of primitives in our string grammar fuzzy clustering algorithms.

In this paper, we propose the String Grammar Unsupervised Possibilistic C-Medians (sgUPCMed) algorithm to find multiprototypes of string data, i.e., Thai hand sign words in this work, from the training data set. We briefly describe the sgUPCMed algorithm here. Let $S = \{s_1, s_2, \ldots, s_N\}$ be a set of N strings. Each string (s_k) is a sequence of symbols (primitives). For example, $s_k = (x_1 x_2 \ldots x_l)$, a string with length l, where each x_i is a member of a set of defined symbols or primitives. Let $\mathbf{V} = (sc_1, sc_2, \ldots, sc_c)$ represent a C-tuple of string prototypes, each of which characterises one of the C clusters. Let $\mathbf{U} = [u_{ik}]_{C \times N}$ be a membership value of string k in cluster i. Let $\mathbf{T} = [t_{ik}]_{C \times N}$ be a possibilistic value of string k in cluster i. Since this is a string calculation, the numeric distance metrics cannot be used in this case. Hence, the distance metric used in the paper is the Levenshtein distance [50–53] between string s_j and string prototypes sc_i (Lev(sc_i, s_j)) (a smallest number of transformations needed to derive one string from the other) between input string j and cluster prototype i.

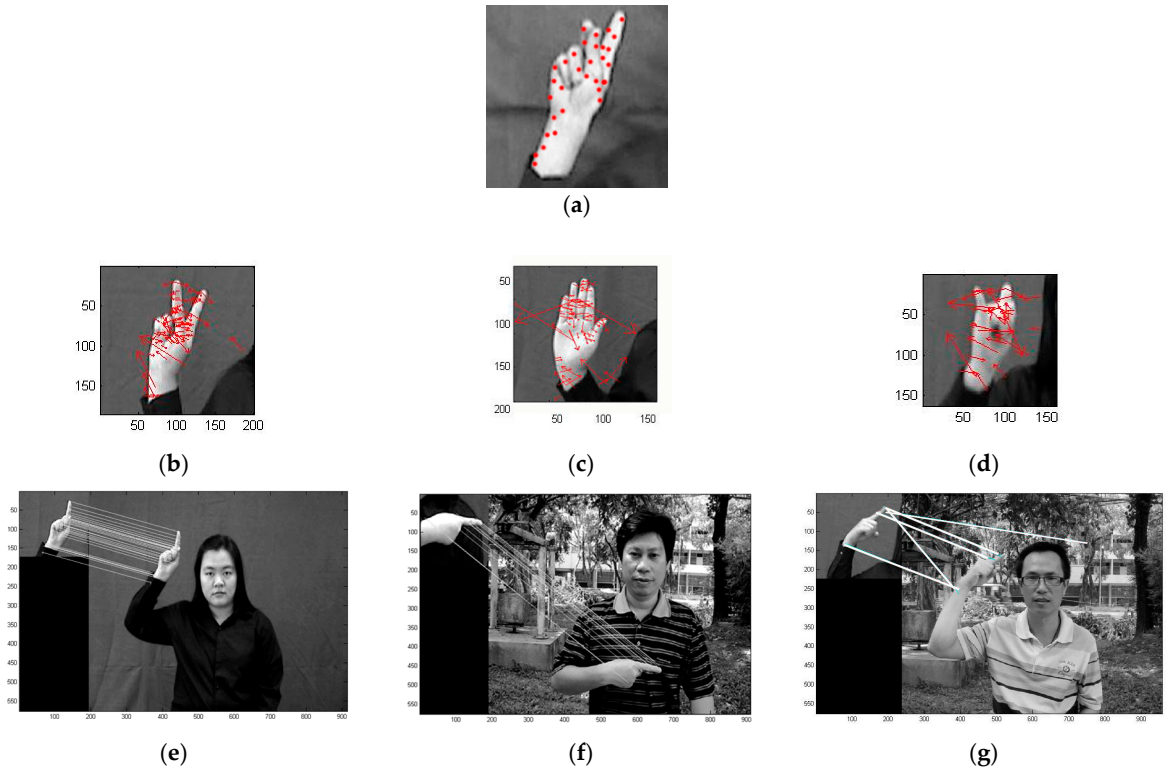

Figure 3. Keypoint descriptor generation (**a**) keypoints found on a keyframe; keypoint descriptors found on hand gestures (**b**) "f", (**c**) "l", and (**d**) "d1"; and the hand gesture (**e**) "b" assigned to a test image using the SIFT method and test images within a constraint environment, (**f**) "e" assigned to a test image using the SIFT method and test frames without a constraint environment, and (**g**) "nh" assigned to a test image using the SIFT method and test frames without a constraint environment.

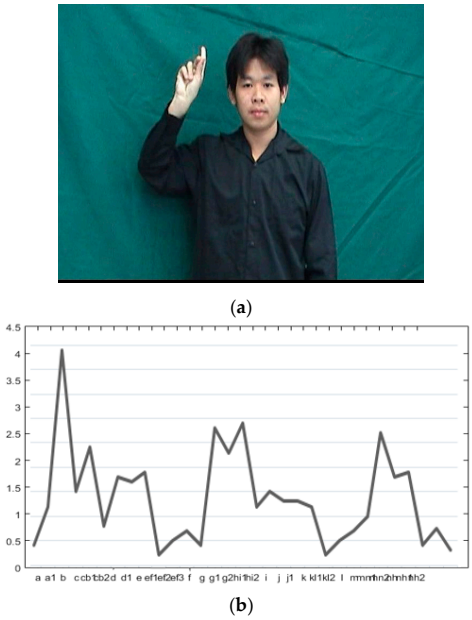

Figure 4. *Avg_Match* of symbol. The matched symbol is "b": (**a**) hand gestures "b"; (**b**) Graph of *Avg_Match* of symbol.

Our String Grammar Unsupervised Possibilistic C-Medians (sgUPCMed) algorithm is a modified version of the unsupervised possibilistic C-medians [54]. They are based on the same concept. That is, the objective function is based on the fuzzy C-means (FCM) clustering algorithm and two cluster validity indices, i.e., the partition coefficient (PC) and the partition entropy (PE). However, in our case, the feature vectors are not numeric vectors, but strings, and the distance is not the Euclidean distance, but the Levenshtein distance. Hence, the sgUPCMed's objective function is:

$$\min \sum_{k=1}^{N} \sum_{i=1}^{C} u_{ik}^{m} Lev(s_k, sc_i) + \frac{\beta}{m^2 \sqrt{c}} \sum_{i=1}^{C} \sum_{k=1}^{N} (u_{ik}^{m} \log u_{ik}^{m} - u_{ik}^{m}) \quad \text{for } k = 1, \ldots, N \text{ and } 0 \leq u_{ik} \leq 1. \tag{2}$$

where u_{ik} is the membership value of string s_k belonging to cluster i, sc_i is the string prototype of cluster i, m is the fuzzifier (normally, $m > 1$), β is a positive parameter, and N is the number of strings. Yang and Wu [19] defined β as the sample covariance based on the Euclidean distance. However, our data set is a string data set, and our β is calculated based on the Levenshtein distance as:

$$\beta = \frac{\sum_{k=1}^{N} Lev(Med, s_k)}{N} \tag{3}$$

where *Med* is the median string of the data set, i.e.,

$$Med = \underset{j \in S}{\operatorname{argmin}} \sum_{k=1}^{N} Lev(s_j, s_k) \tag{4}$$

Theorem 1 (sgUPCMed). *If $Lev(s_k, sc_i) > 0$ for all i and k, when m, η, k >1, and S contains C < N distinct string data, then $J_{m,\eta}$ is minimised only if the update equation of u_{ik} is:*

$$u_{ik} = \exp\left(-\frac{m\sqrt{c} Lev(sc_i, s_k)}{\beta}\right) \tag{5}$$

Proof. The reduced form of Equation (5) with **V** fixed for the kth column of **U** is:

$$\min\left\{ L_i(\mathbf{U},\lambda) = J_m^{ik}(\mathbf{U}) = u_{ik}^m Lev(s_k, sc_i) + \frac{\beta}{m^2\sqrt{c}} \sum_{i=1}^{C}\sum_{k=1}^{N} (u_{ik}^m \log u_{ik}^m - u_{ik}^m) \right\} \tag{6}$$

From the Lagrange multiplier theorem, the derivative of $L_i(\mathbf{U},\lambda)$ with respect to u_{ik} and setting it to zero leads to:

$$\begin{aligned} \frac{\partial L_i(\mathbf{U},\lambda)}{\partial u_{ik}} = m(u_{ik})^{m-1} Lev(s_k, sc_i) + \frac{\beta}{m^2\sqrt{c}}(m u_{ik}^{m-1} m \ln u_{ik} - m u_{ik}^{m-1}) = 0 \\ \frac{m\sqrt{c}Lev(s_k,sc_i) + \beta {u_{ik}}^{m-1} \ln u_{ik}}{\sqrt{c}} = 0 \\ u_{ik} = \exp\left(-\frac{m\sqrt{c}Lev(sc_i,s_k)}{\beta}\right) \end{aligned} \tag{7}$$

□

The fuzzy median string [55–58] is utilised as a cluster center update equation because of the utilisation of the Levenshtein distance in our string grammar clustering. Hence, the cluster center i update equation is:

$$sc_i = \underset{j\in S}{\operatorname{argmin}} \sum_{k=1}^{N} u_{ik}^m Lev(s_j, s_k) \quad \text{for } 1 \le i \le C \tag{8}$$

However, Ref. 56 and 57proved that the modified median string provides a better classification rate than the regular median string. Then, the modified method in [56–58] is also modified to calculate the fuzzy median. Let Σ^* be the free monoid over the alphabet set Σ and a set of strings $S \subseteq \Sigma^*$. Then, the modified fuzzy median, i.e., an approximation of fuzzy median using edition operations (insertion, deletion, and substitution) over each symbol of the string [56–58] will be:

$$sc_i = \underset{j\in\Sigma^*}{\operatorname{argmin}} \sum_{k=1}^{N} u_{ik}^m Lev(s_j, s_k) \text{for } 1 \le i \le C \tag{9}$$

The modified fuzzy median string algorithm of the sgUPCMed isshown in Algorithm 1:

Algorithm 1. The modified fuzzy median string algorithm of the sgUPCMed.

Start with the initial string s.
For each position i in the string s
1. Build alternative
Substitution: Set $z = s$. For each symbol $a \in \Sigma$
(a) Set z' to be the result of substituting i^{th} symbol with symbol a.
(b) If $\sum_{k=1}^{N} (u_{ik}^m) Lev(z', s_k) < \sum_{k=1}^{N} (u_{ik}^m) Lev(z, s_k)$.
then, set $z = z'$.
Deletion: Set y to be the result of deleting the i^{th} symbol of s.
Insertion: Set $x = s$. For each symbol $a \in \Sigma$
(a) Set x' to be the result of adding a at position i^{th} of s.
(b) If $\sum_{k=1}^{N} (u_{ik}^m) Lev(x', s_k) < \sum_{k=1}^{N} (u_{ik}^m) Lev(x, s_k)$.
then, set $x = x'$.
2. Choose an alternative
Select string s' from the set of strings {s,x,y,z} from step 1 using

$$s' = \underset{G\in\{s,x,y,z\}}{\operatorname{argmin}} \sum_{k=1}^{N} (u_{ik}^m) Lev(G, s_k).$$

Then, set $s = s'$.

Hence, the summary of sgUPCMed algorithm is shown in Algorithm 2.

Algorithm 2. sgUPCMed algorithm.

Store N unlabeled finite strings $S = \{s_k; k = 1, ..., N\}$
Initialise string prototypes for all C classes
Set m
Compute β using Equation (3)
Do {
    Update membership value using Equation (5)
    Update center string of each cluster i (sc_i) using Equations (8) and (9)
} **Until** (stabilise)

After, the multiprototypes, i.e., $SC = \left\{sc_1^1, \ldots, sc_{N_1}^1, sc_1^2, \ldots, sc_{N_2}^2, sc_1^C, \ldots, sc_{N_C}^C\right\}$ where sc_k^j is string prototype k of class j, are created. The fuzzy k-nearest neighbour [48] is utilised as a classifier. The FKNN is similar to the k-nearest neighbour (KNN), except that each data point can belong to multiple classes, with different membership values associated to these classes. For each string s, the membership value u_i in class i can be calculated as the following:

$$u_i(s) = \left.\frac{\sum_{j=1}^{K} u_{ij}\left(\frac{1}{Lev\left(sc_j^q, s\right)}\right)^{1/(m-1)}}{\sum_{j=1}^{K}\left(\frac{1}{Lev\left(sc_j^q, s\right)}\right)^{1/(m-1)}}\right\} \quad (10)$$

where u_{ij} is the membership value of the jth prototype from class q (sc_j^q) in class i, c is the number of classes, and K is the number of nearest neighbours. The decision rule for the test string s is:

$$s \text{ is assigned to class } i \text{ if } u_i(s) > u_j(s) \text{ for } j \neq i \quad (11)$$

In the experiment, since we know the class that the prototype string sc_j^q represents, we set $u_{iq} = 1$ for sc_j^q in class q and 0 for all of the other classes. The parameter m is used to determine how heavily the distance is weighted when calculating each neighbour's contribution to the membership value, and its value is chosen for our experiment as $m = 2$.

To summarise our algorithm as shown in Figure 1, we first need to create the multiprototypes training process with the SIFT and the sgUPCMed algorithms. The Levenshtein distance and the FKNN are used to recognise the sign language words. The computational complexity of the training process will be $O(F{\cdot}(ot{\cdot}m{\cdot}n + kp + m{\cdot}n + m{\cdot}n{\cdot}b^2{\cdot}kp)) + O(F{\cdot}nS{\cdot}dl^2) + O((l^2{\cdot}N^2) + (l^2{\cdot}N^2) + (l^3{\cdot}c{\cdot}\mid\Sigma\mid))$, where m and n are the width and height of an image, ot and kp are the number of octaves and the number of keypoints, dl is the SIFT descriptor length, F is the number of video image frames, and nS is the number of keyframes in the signature library. The remaining parameters are string length (l) (equals to F), the number of data samples (N), the number of clusters (c), and the alphabet set (Σ). For the recognising process, the computational complexity is $O(F(ot{\cdot}m{\cdot}n + kp + m{\cdot}n + m{\cdot}n{\cdot}b^2{\cdot}kp)) + O(F{\cdot}nS{\cdot}dl^2) + O(N{\cdot}(l^2 + N\log N + K))$, where K is the number of nearest neighbours used.

3. Experimental Results

An experiment data set (training and test video data sets) was collected from 25 subjects at different times of day for several days. Subjects 1–20 were asked to wear a black shirt with long sleeves and stand in front of a dark background. In contrast, subjects 21–25 were asked to wear short sleeves and were in front of various complex natural backgrounds. The data set consisted of 10 hand sign words (classes), i.e., "elder", "grandfather", "grandmother", "gratitude", "female", "male", "glad", "thank you", "understand", and "miss". The number of samples for each hand sign is shown in

Table 2. We first collected the keyframes in the signature library from subjects 1–5. We manually selected a portion of the hand in each frame that measured 190 × 190 pixels. After that, we computed keypoint descriptors for each frame using SIFT, and then stored it in the signature library database. The test videos were recorded for subjects 1–20 (with constraint) and for subjects 21–25 (without any constraints). Each video was decimated, which left only 14 frames. Each frame was matched to a representative symbol in the signature library using SIFT, and the threshold values for the experiment varied between 0.65, 0.7, and 0.75.

Table 2. Number of words in the training data set with subjects 1–15 and the test data set with subjects 1–25.

Data Set	Subjects	Elder	Grandfather	Grandmother	Gratitude	Female	Male	Glad	Thank You	Understand	Miss
Training data set	1a	36	36	36	36	12	36	36	36	36	36
	2a–15a	32	32	32	32	32	32	32	32	32	32
Test data set	1b	12	12	12	12	12	12	12	12	12	12
	2b–15b	8	8	8	8	8	8	8	8	8	8
	16–19	20	20	20	20	20	20	20	20	20	20
	20	10	10	10	10	10	10	10	10	10	10
	21–25	5	5	5	5	5	5	5	36	5	5

Our experiment was divided into three parts, i.e., training with 1a, training with 1a–5a, and training with 1a–15a. There were three groups of blind test: data set 1b–15b, data set 16–20 (used to represent the signer-independent cases), and 21–25 (with various complex natural backgrounds). For all of the training and test data sets, we assigned symbols to each frame in the training data set using the SIFT method. We use multiprototypes created from the sgUPCMed algorithm to classify 10 hand sign words, in which the lengths of each string representation were 14. Afterwards, we created multiprototypes in terms of a sequence of primitives, and the test string was assigned to the word that the closest prototype belonged to, according to the FKNN algorithm with the Levenshtein distance.

We implemented four-fold cross validation on the training set and implemented the sgUPCMed with four, eight, and 12 clusters on each class separately to create multiprototypes for each class. Then, the FKNN with K = 1, 3, 5, 7 and 9 were implemented as classifiers. Figures 5–7 show the best and average correct classification of the validation set of training with 1a, training with 1a–5a, and training with 1a–15a for the FKNN with K = 1, 3, 5, 7, and 9, respectively. We can see that the best classification rate was at 98.81%, when trained with 1a and 12 prototypes for each class with 0.75 SIFT threshold and K = 9. Whereas, when trained with 1a–5a, the best classification rate that system provided was at 94.55%, with 12 prototypes, 0.65 SIFT threshold, and K = 9. For the 1a–15a training data set, we obtained 88.37% correct classification with 12 prototypes, 0.7 SIFT threshold, and K = 9. From all of the experiments, we can see that if we increase the number of prototypes in the process of string grammar clustering, there is a chance that the classification rates of all of the types of signer will also increase. From the results in Figures 5–7, the 12-prototype string grammar clustering with 9-FKNN gave a classification rate that was higher than the other prototypes. Hence, we used 12-prototypes string grammar clustering with 9-FKNN to test the blind test data set, as shown in Figures 8–11.

From Figure 8, the classification rates for 1a and 1b (signer-dependent cases) were 97.92% and 90.56%, respectively, since this system was trained with the first subject. The average classification rate from the signer semi-independent cases (2a–5a and 2b–5b) was approximately 59.50%. Meanwhile, the average classification results from subjects 6a–15a, 6b–15b, and 16–20 (the signer-independent cases) was around 54.32%.

The classification results on the test set when we trained on the data set 1a–5a are shown in Figure 9. The best average classification rate of the blind test data sets of the signer-dependent cases (subjects 1–5) was around 95.24%. The best average classification rates of the blind test data sets of the signer-independent cases, subjects 6–15 and subjects 16–20, were 67.64% and 71.8%, respectively. Therefore, we can see that the classification rates from all of the types of signer are increased, if we increase the number of signers in the training process of the sgUPCMed method.

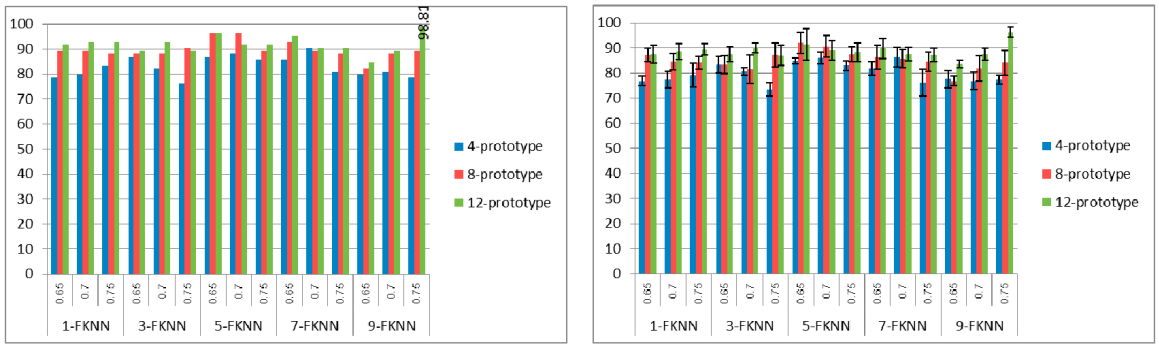

Figure 5. (**a**) The best and (**b**) average ± standard deviation classification rate (%) of the validation set from four-fold cross-validation when trained with data set 1a for FKNN with K = 1, 3, 5, 7, and 9.

From the blind test results from the training process with 1a–15a, as shown in Figure 10, we can see that the best average classification rates for the signer-dependent cases (subjects 1–5) was 90.99%, whereas that of the signer semi-dependent cases (subjects 6–15) was 85.14%. Meanwhile, the best average classification rate of the signer-independent cases (subjects 16–20) was 79.90%. Again, the greater the number of training subjects, the more accurate the system.

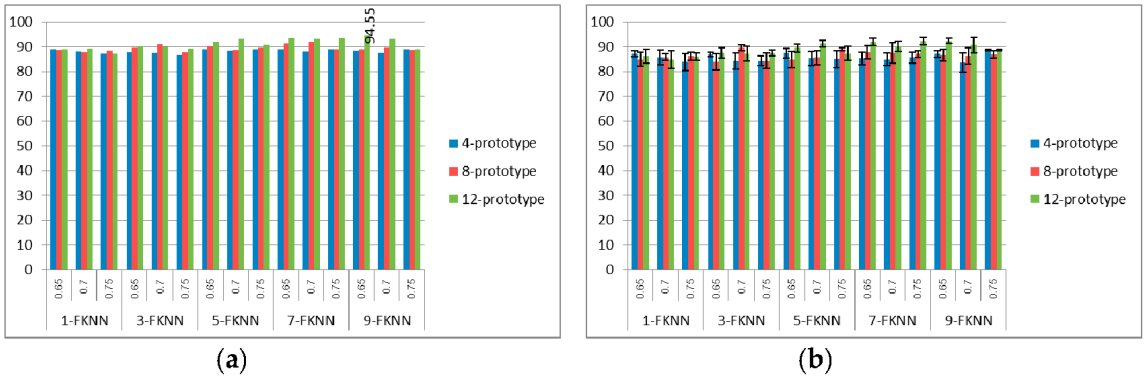

Figure 6. (**a**) The best and (**b**) average ± standard deviation classification rate (%) of the validation set from four-fold cross-validation when trained with data set 1a–5a for FKNN with K = 1, 3, 5, 7, and 9.

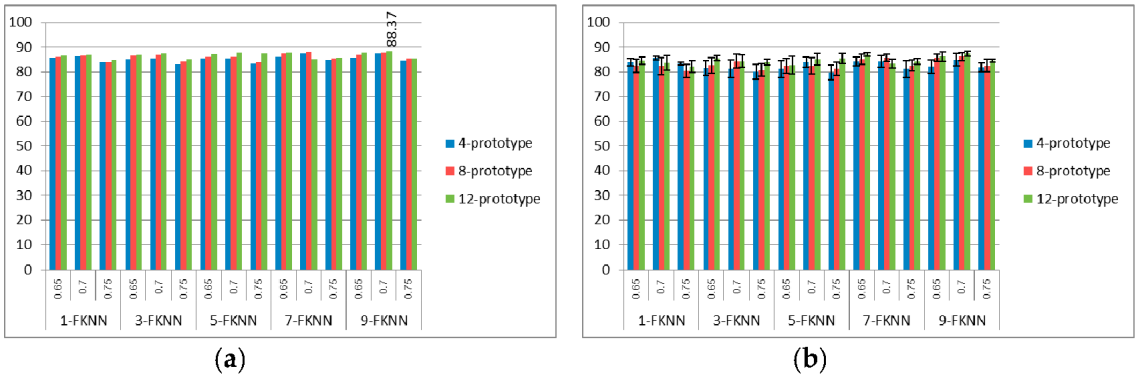

Figure 7. (**a**) The best and (**b**) average ± standard deviation classification rate (%) of the validation set from four-fold cross-validation when trained with data set 1a–15a for FKNN with K = 1, 3, 5, 7, and 9.

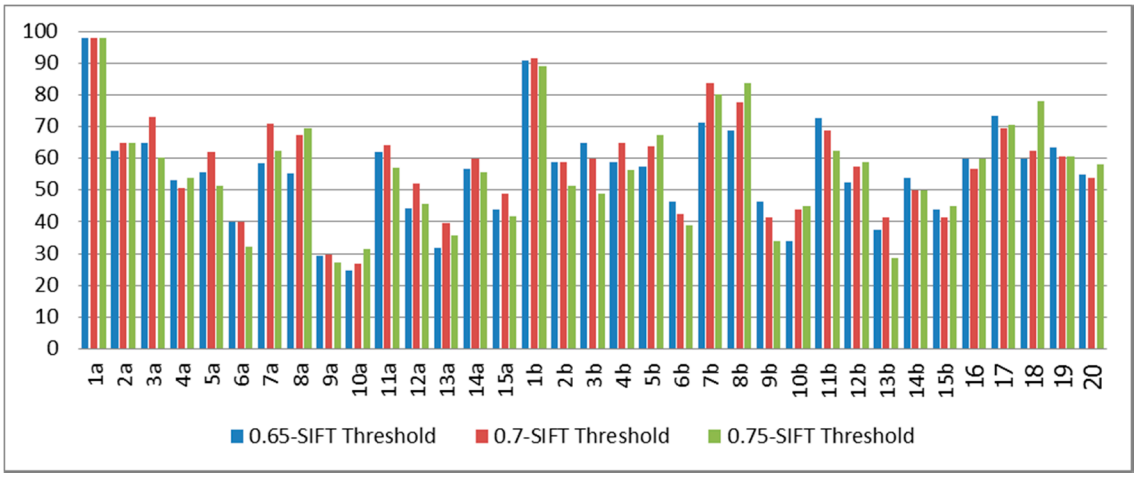

Figure 8. Classification rate on test sets when trained with data set 1a.

Hence, we selected the system trained with subjects 1a–15a, and tested it on subjects 21–25, who were asked to wear any type of shirt and stand in front of natural backgrounds. Each subject performed each sign five times at any time of day. The classification results are shown in Figure 11. We can see that the best result for subjects 21–25 were 76% with 0.70 SIFT threshold, 80% with 0.70 SIFT threshold, 66% with 0.70 SIFT threshold, 68% with 0.65 and 0.70 SIFT threshold, and 62% with 0.75 SIFT threshold for the five subjects, respectively. Since the signers of this test set (subjects 21–25) were different signers from the training data set and the signature library, the results of this experiment provided low classification. Furthermore, when we used SIFT with the unconstrained system and complex natural backgrounds, the matched keypoints might be incorrectly matched, as shown Figure 3g. We can use Equation (1) to find the correct symbol for each test frame, even though it has some mismatched keypoints from the SIFT process. However, our algorithm cannot find the right symbol if there are too many mismatched keypoints.

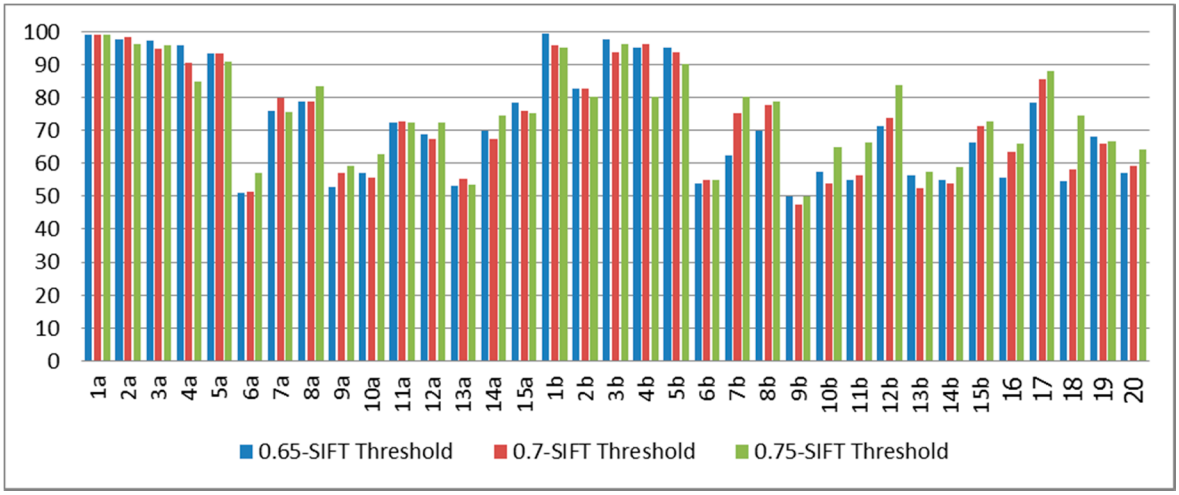

Figure 9. Classification rate on test sets when trained with data set 1a–5a.

Now, we compare the performance of our algorithm with the reported classification rate results of the Thai Sign Language (TSL) translation system [43] using Hidden Markov Model (HMM) on the same data set as shown in Table 3. We can see that the best results for the signer-dependent, signer semi-dependent and signer-independent cases from the TSL with HMM were 88.60%, 80.55%,

and 76.75%, respectively. Whereas those from our proposed algorithms were 90.85%, 85.14%, and 79.90%, respectively. A comparison can be done between this method and the best average of our translation system. Our system yields a pretty good result that is comparable with TSL [43] in all of the experiments. HMM may create higher misclassification than our method because the HMM model that gives the maximum value is not the right one. Meanwhile, our method not only chooses the maximum one, it also utilises string grammar fuzzy clustering to find the multiprototypes, and after that, the FKNN algorithm will choose the closest string prototypes using the k-nearest neighbours.

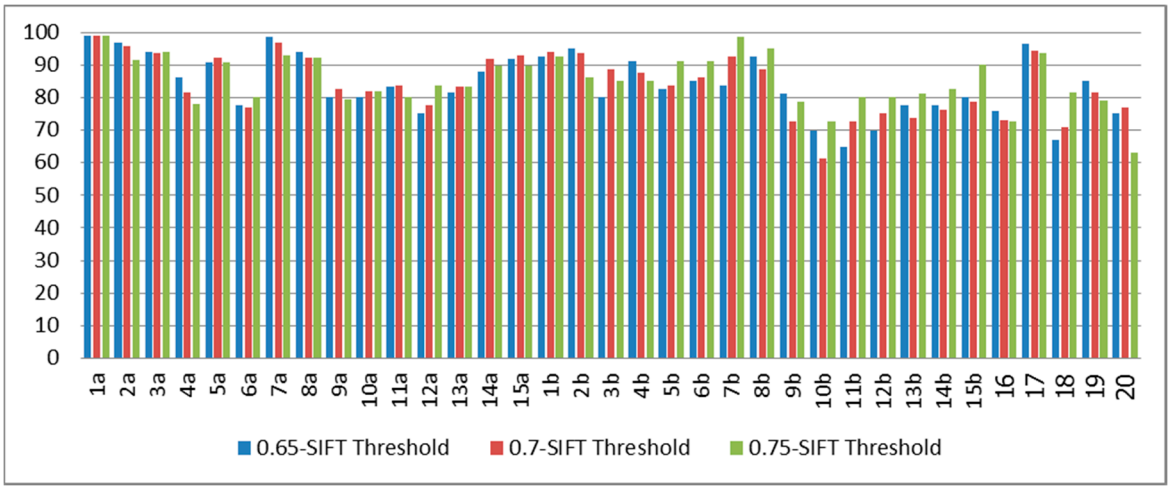

Figure 10. Classification rate on test sets when trained with data set 1a–15a.

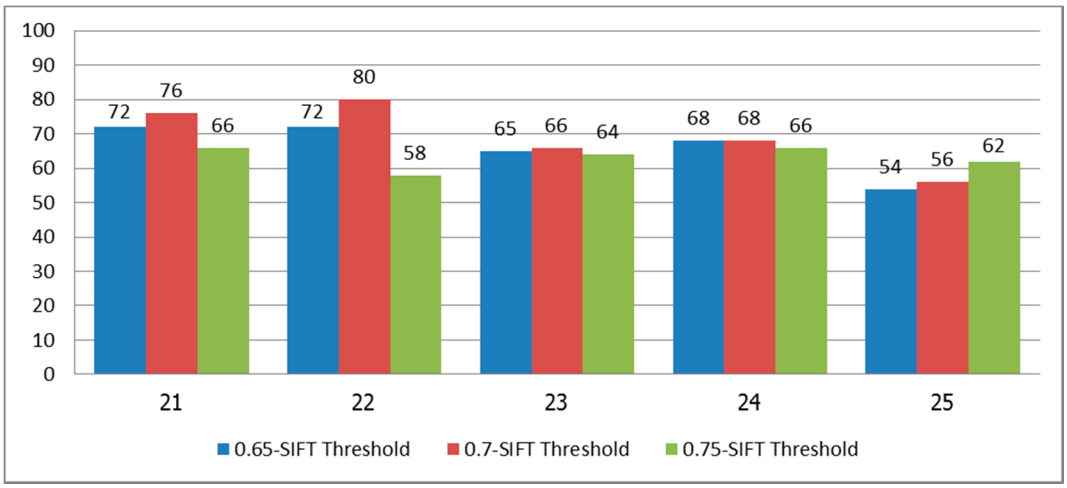

Figure 11. Classification rate on test sets when trained with data set 1a–15a, and tested with unseen signers against various complex natural backgrounds.

In order to consider how this system performs when implemented on other sign languages; we implemented our system on the RWTH-BOSTON-50 data set [46]. Although this data set has 50 words from three signers, there are 19 words in which the total number of video sequences for each word is around one to three sequences. Hence, we only used the 31 words that have more than three sequences for each word. The details of the words used and the number of sequences and signers performing the words are shown in Table 4. Hence, for 31 words, there are 437 sequences in total. Again, for this data set, we collected keyframes to create a signature library. Since each signer did not perform the same amount of sequences for each word, the numbers of repetitions selected manually for each

keyframe were not the same. There were 81 keyframes; hence, there are 319 keyframes in total in the signature library. An example of keyframes and the corresponding number of keyframes is shown in Figure 12. In this case, to generate a string for each word, the minimum number of frames of each word is used as the number of symbols F of the word sequence, because each word contains a different number of frames, as shown in Table 4. For example, for the word "ARRIVE", we created a string with a length of seven, whereas for "BOX", the created string had a length of nine. Again, we chose F image frames with approximately equal spacing from each video sequence to generate a test string for each word. This shows that our sgUPCMed does not require the strings to have the same length in order to perform the string clustering.

Table 3. Comparison of classification rates on test sets of our proposed method with Thai Sign Language (TSL) (with HMM).

Method	Mode	The Best Average of Classification Rate (%)		
		SIFT Threshold		
		0.65	**0.70**	**0.75**
TSL (with HMM) [43]	signer-dependent	<u>88.60</u>	88.29	87.82
	Signer semi-dependent	80.35	80.45	<u>80.55</u>
	signer-independent	<u>76.75</u>	76.32	75.23
Proposed Method	signer-dependent	<u>90.85</u>	90.99	89.32
	Signer semi-dependent	81.67	81.88	<u>85.14</u>
	signer-independent	<u>79.90</u>	79.40	77.90

Note: The best classification rate is indicated with bold and underline character.

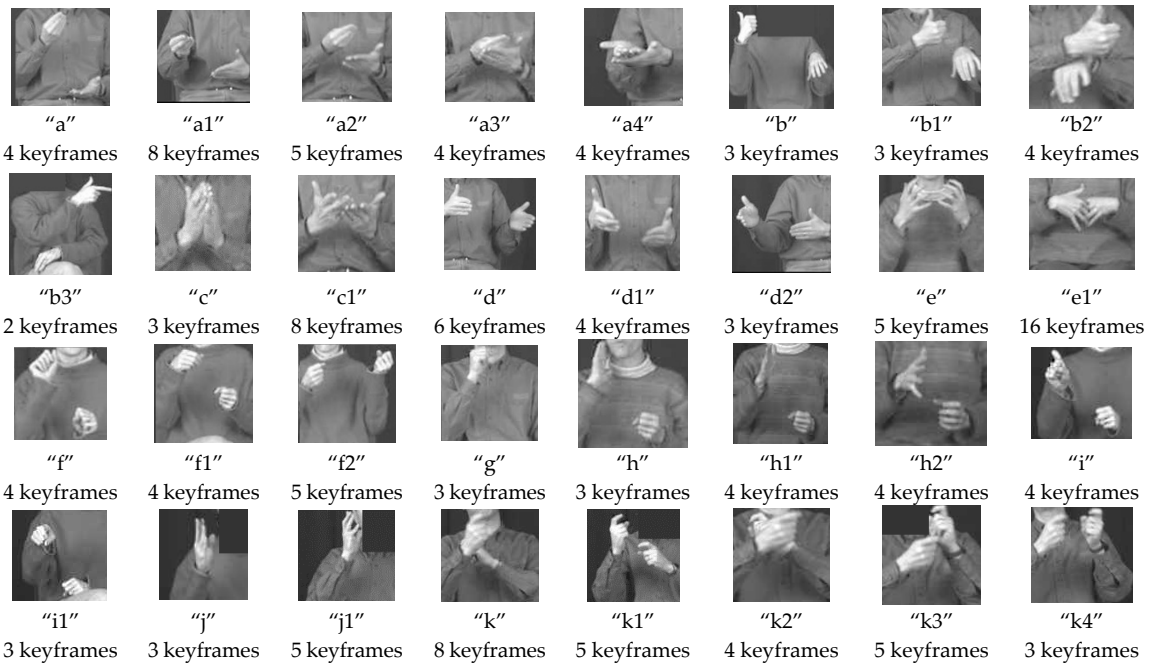

Figure 12. *Cont.*

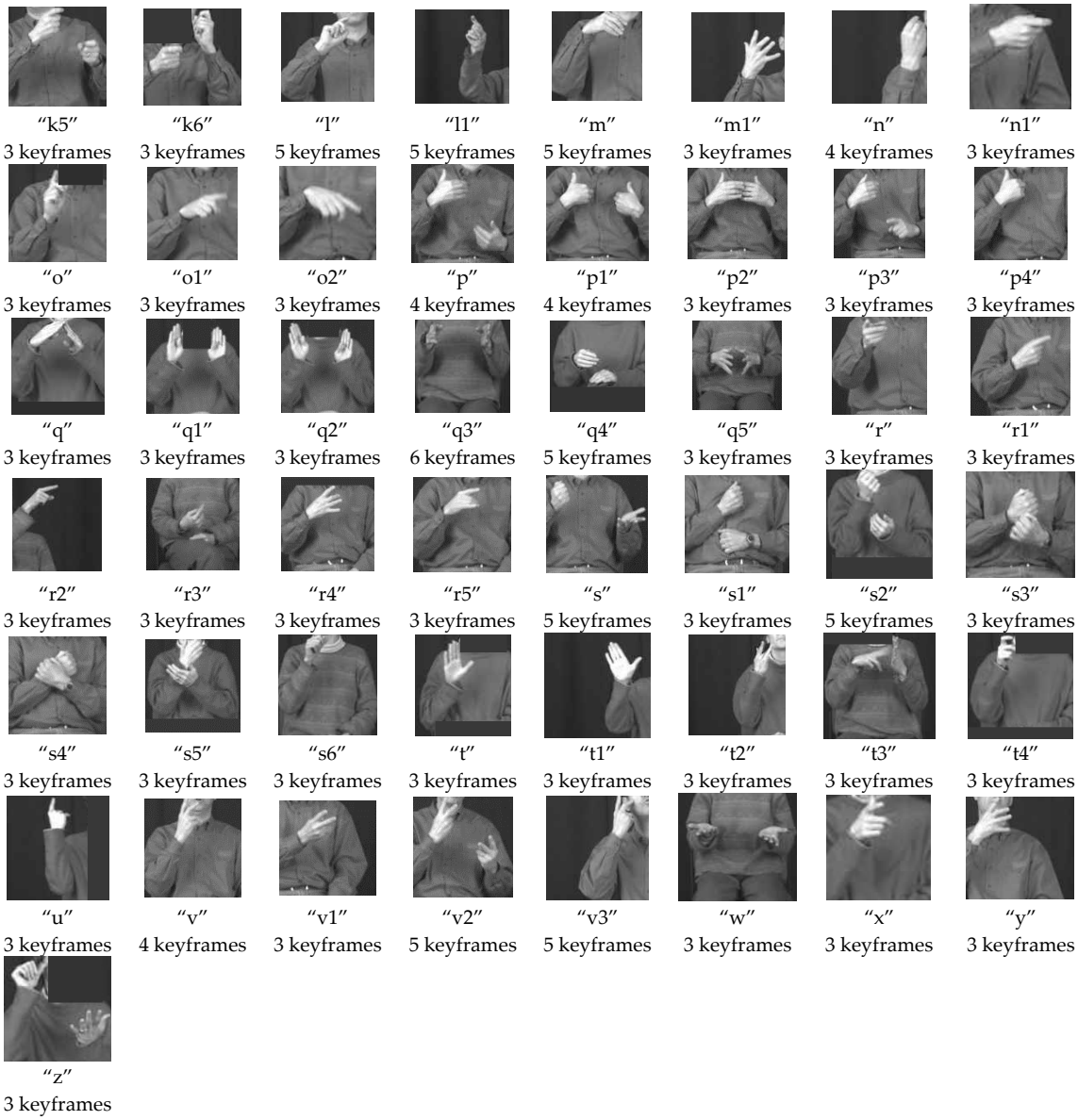

Figure 12. Example of 81 hand gestures for the RWTH-BOSTON-50 data set.

Table 4. Details of RWTH-BOSTON-50 [46] used in the experiment.

Word	ARRIVE	BLAME	BOOK	BOX	BREAK-DOWN	BUY	CAN	CAR
# sequences	15	6	23	7	5	31	19	19
Minimum sequence length	7	8	5	9	14	5	4	4
# of signers	3	3	3	1	2	3	3	3
Word	**FINISH**	**FUTURE**	**GIVE**	**GO**	**HAVE**	**HOUSE**	**IX_i**	**IX_i far**
# sequences	7	21	24	19	6	12	37	12
Minimum sequence length	3	6	6	6	5	15	5	15
# of signers	2	3	2	1	2	2	3	2
Word	**IX_1p**	**LIKE**	**LOVE**	**NEW**	**NOT**	**POSS**	**PREFER**	**READ**
# sequences	8	6	16	7	7	12	5	4
Minimum sequence length	3	5	5	6	5	5	5	6
# of signers	3	2	3	2	2	3	2	1
Word	**SHOULD**	**SOMETHING GONE**	**VISIT**	**WHAT**	**WHO**	**WOMAN**	**YESTERDAY**	
# sequences	10	12	18	24	25	8	12	
Minimum sequence length	4	4	6	14	6	6	6	
# of signers	3	3	3	3	3	2	3	

We implemented leave-one-out cross-validation for this data set, meaning that we only selected one sequence for each word to be our validation set, and used all of the other sequences to train our sgUPCMed algorithm. The numbers of prototypes created by our sgUPCMed algorithm were not the same, as shown in Table 5, because of the different numbers of word repetitions of the data set. We then used FKNN with only one nearest neighbour to find the best word match. The SIFT threshold used in this data set varied from 0.4 to 0.75, with a step size of 0.5. The results from the validation data set and the combined training and validation set are shown in Table 6. The SIFT threshold used in this data set varied from 0.4 to 0.75, with a step size of 0.5. The best result of the validation set was 88.56%, while the best result of the combined training and validation set was 91.35%.

Table 5. Number of prototypes for the RWTH-BOSTON-50 data set.

Word	ARRIVE	BLAME	BOOK	BOX	BREAK-DOWN	BUY	CAN	CAR
# prototypes	5	3	4	2	2	6	3	4
Word	**FINISH**	**FUTURE**	**GIVE**	**GO**	**HAVE**	**HOUSE**	**IX_i**	**IX_i far**
# prototypes	2	3	6	4	2	2	4	2
Word	**IX_1p**	**LIKE**	**LOVE**	**NEW**	**NOT**	**POSS**	**PREFER**	**READ**
# prototypes	3	2	3	2	2	3	2	2
Word	**SHOULD**	**SOMETHING GONE**	**VISIT**	**WHAT**	**WHO**	**WOMAN**	**YESTERDAY**	
# prototypes	3	3	3	3	3	2	3	

Table 6. Classification rate on the validation set, and the classification rate on the combined training and validation sets of the RWTH-BOSTON-50 data set.

Data Set	SIFT Threshold							
	0.4	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**
Validation set	86.27	88.56	86.73	87.19	86.73	85.81	84.21	84.21
Combined training and validation set	91.05	91.35	90.81	90.68	90.55	90.38	90.18	89.94

Note: The best classification rate is indicated with bold and underline character.

We compared our results with the existing algorithms to show the performance of our proposed method. Tables 7 and 8 show the performance of the proposed algorithm and that of the existing algorithms on the Thai sign language data set and the RWTH-BOSTON data set, respectively. However, on the RWTH-BOSTON data set, we merely show the performance of different algorithms ([27,45,46]) on the same data set without any comparison analysis, because it could be thought of as an unfair comparison. Table 8 shows that our result is in the same range (88.56% correct classification rate) of those in [27,45,46] with different experiment settings. Also, it is difficult to directly/indirectly compare our method with the other methods, because the sign languages of other countries are different from Thai Sign Language. Hence, we can only say that our algorithm can be used in different sign languages, not just in Thai Sign Language, and it can provide a reasonable result that is within the same range as the existing algorithms shown in Table 1. Moreover, in order to implement our proposed algorithm in a different sign language, we need to create a signature library for each sign language, since it is not the same for different sign languages. We also need to train our system according to that separately.

Table 7. Indirect comparison of the classification rate of the test sets for our proposed method and the other methods on the Thai Sign Language data set.

Method	# of Recognised Words	Data Set	Instrument Used	Mode	Pre-Process with Segmentation	# of Signers	Classification Rate (%)
TSL (with HMM) [43]	10	Validation set	None: free hand	signer-dependent	No	5	86–95 (on average)
			None: free hand	signer-semi-dependent		10	80 (on average)
			None: free hand	signer-independent		5	75–76 (on average)
Proposed Method on Thai Sign Language data set	10	Validation set	None: free hand	signer-dependent	No	5	89–91 (on average)
			None: free hand	signer-semi-dependent		10	81–85 (on average)
			None: free hand	signer-independent		5	77–80 (on average)

Table 8. Indirect comparison of the classification rate of the test sets for our proposed method and the other methods on the RWTH-BOSTON-50 data set.

Method	# of Recognised Words	Data Set	Instrument Used	Mode	Pre-Process with Segmentation	# of Signers	Classification Rate (%)
ASL_RWTH-BOSTON-50 [27]	30	Combined training and test data set	None: free hand	Combined signer-dependent and signer-independent	Yes	3	89.09
ASL_RWTH-BOSTON-50 [45]	15	Test data set	None: free hand	signer-dependent	Yes	3	93.33
ASL_RWTH-BOSTON-50 [46]	50	Test data set	None: free hand	signer-dependent	No	3	82.8
Proposed Method on RWTH-BOSTON-50	31	Validation set (Leave-one-out strategy)	None: free hand	Combined signer-dependent and signer-independent	No	3	88.56
		Combined training and validation data set	None: free hand				91.35

One might also ask about the difference between the proposed method with that in [42,43]. The method in [42] can be used to translate Thai finger-spelling, but not Thai hand sign. The method only uses the SIFT method to find a matching alphabet. If the composed alphabets match with any spelled words, then the system reports that word. The one in [43] used the SIFT method to extract this feature as in this proposed method. However, it used HMM as a classifier, while we use the sgUPCMed and the FKNN algorithms as our classifiers here. In addition, from Table 7, our proposed method is better than the results shown in [43].

One of the advantages of our algorithm is that it provides a good result for recognising isolated sign language words that have similar hand gestures. Examples of Rframes representing the Thai Sign Language words of "grandmother" and "grandfather" are shown in Figure 13. The symbol sequence of the word "grandmother" is "mmmbbbbbmbbbbb", while that of the "grandfather" is "mmmmcb2cb2bbbbbbbb". We can see that these two words have similar hand gestures in the sequence. The blind test recognition rates of "grandmother" and "grandfather" at a SIFT threshold of 0.65 and FKNN with $K = 9$ are 72% and 84%, respectively. Examples of frames from "GO" with the symbol sequence of "ooooo1o2" and "SHOULD" with the symbol sequence of "ooo1o2" from the RWTH-BOSTON-50 are shown in Figure 14. Again, the recognition rates with a SIFT threshold of 0.45 with one nearest neighbour (FKNN) of these "GO" and "SHOULD" are 73.68% and 80%, respectively. These are examples of sign language words with similar hand gesture. However, for the whole data set, there were some other words with similar hand gestures as well. With this condition, our system can still provide a good recognition rate for the whole data set. However, some misclassifications have occurred in our system, which might be the result of some hand gestures that are very similar, yet represent different symbols in the signature library. An example of similar hand gestures for symbols "g", "g2", and "k" in Thai Sign Language are shown in Figure 15. Meanwhile, Figure 16 shows an example of the similar hand gestures for symbols "t", "t1", "v", and "v3" in RWITH-BOSTON-50. If this occurs for different words, it might cause the system to think that they are the same word.

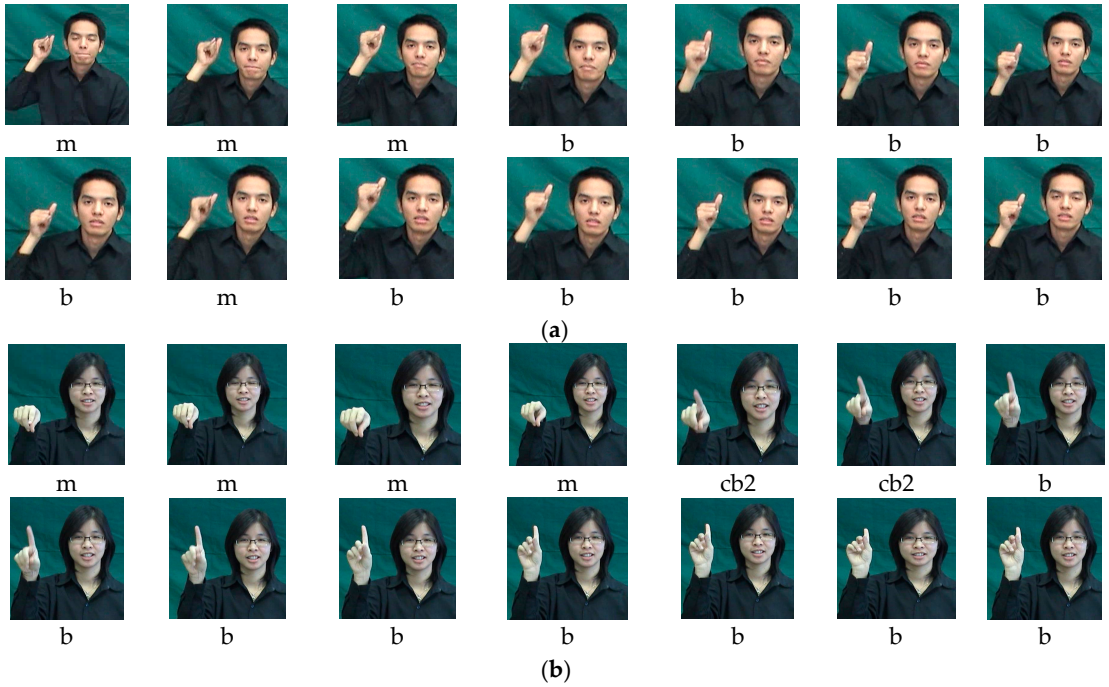

Figure 13. Representative frames (Rframes) of Thai sign language words (**a**) "Grandmother" and (**b**) "Grandfather".

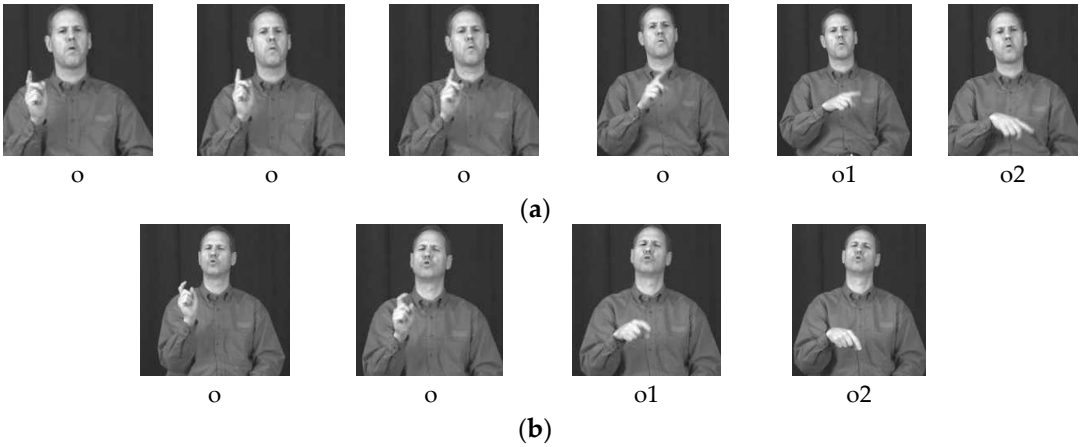

Figure 14. All frames of RWTH-BOSTON-50 word (**a**) "GO" and (**b**) "SHOULD".

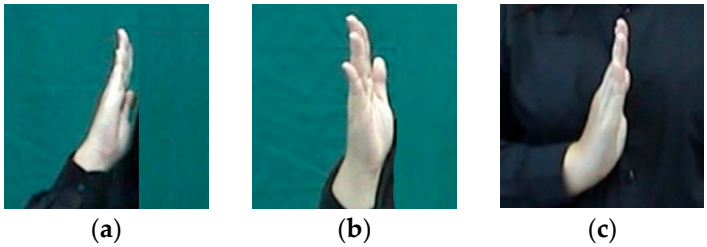

Figure 15. The hand part of similar keyframes (**a**) "g", (**b**) "g2", and (**c**) "k" in Thai Sign Language.

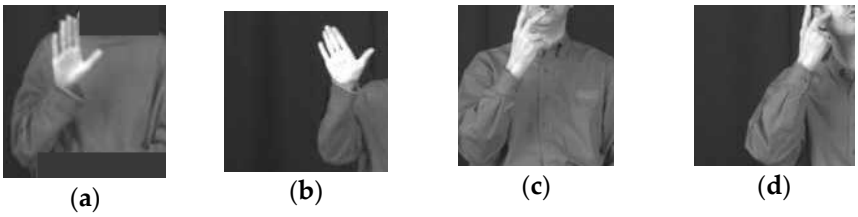

Figure 16. The hand part of similar keyframes (**a**) "t", (**b**) "t1", (**c**) "v", and (**d**) "v3" in RWTH-BOSTON-50.

To implement this system in real time, one might wonder how fast the algorithm will be when implemented online. One of the parameters that might influence the real-time processing is the number of keypoints found on each image frame. The number of minimum, maximum, and average number of keypoints from the Thai Sign Language data set were 56, 153, and 72, respectively. Whereas, those from the RWTH-BOSTON-50 data set were 32, 112, and 41 keypoints, respectively. Of course, the greater the number of keypoints, the slower the algorithm. However, the average recognising processing times of both data sets was approximately one to two seconds per sign word, respectively. This part of the experiment was implemented on a 3.6 GHz Intel Core i7 with 8 GB 2400 MHz DDR4 RAM.

4. Conclusions

In this paper, we improved the dynamic Thai Sign Language translation system with video caption without prior hand region detection and segmentation using the Scale Invariant Feature Transform (SIFT) method and the proposed String Grammar Unsupervised Possibilistic C-Medians (sgUPCMed) algorithm. The SIFT method was used to match test frames with symbols in the signature library, whereas the proposed sgUPCMed algorithm was used to generate multiprototypes for each sign. The fuzzy k-nearest neighbour (FKNN) algorithm was utilised to find the matched sign words. Please

note that because of the different signs in various languages, it is necessary to train the system with several SIFT thresholds and several cluster numbers from sgUPCMed in order to find the best SIFT threshold and the best cluster number. Also, the best number of K in the FKNN for the recognising process can be found by trial and error on several testing data sets before using the system in real-time applications. We found that the best result for the blind Thai Sign Language (isolated sign word) data sets of signer-dependent cases was in between 89% and 91% on average, and the average for the signer semi-independent cases (where the same subjects were used in the string grammar clustering) was around 81–85%. Whereas, the best average classification rate of the blind data sets of the signer-independent cases was 77–80%. Moreover, our system could perform translations for each video without the need for any pre-processing techniques, i.e., segmentation and hand detection. The SIFT method provides more informative information of the position, shape, and orientation of the hand and fingers. This allows the system to be able to recognise hand sign words that have similar gestures. However, when we tested our algorithm with the test subject without any constraint on five signers (subjects 21–25), who were asked to stand in front of various complex backgrounds and could wear any shirt, the best correct classification rate in this case was around 70–80% on average.

To prove our generality over sign languages, we also implemented our proposed algorithm with the RWTH-BOSTON-50 data set, which consists of 31 isolated American Sign Language words. The result showed that the sign language word recognition was 88.56% and 91.35% on the validation set only and for both the training and validation sets, respectively. This shows that our algorithm is flexible enough to be implemented on any sign language.

Since the objective function of our sgUPCMed algorithm is based on the validity indices PC and PE as in the Unsupervised Possibilistic C-Means (UPCM), the exponential membership functions were used to describe the degree of belonging. This is an advantage of our system, because the system can detect the outliers (too far from prototypes) of Thai Sign Language by yielding very low or close to zero membership values for those outliers. However, one constraint of the sgUPCMed algorithm was that it sometimes generated coincident clusters when we ran multiprototype clustering. It might generate prototypes with very close locations, because of the relaxing constraint of the columns and rows of the independent possibilistic values.

We also compared our method with the HMM method, and demonstrated that the best classification of our method is better than HMM on all of the experiments. The HMM method may create higher misclassification rates than our method because of the chance that the selected probability-based HMM model is not the actually the best one. Although our system provides better classification rates than previous methods for Thai Sign Language, there are still some issues regarding its performance that we could improve. For instance, the string representation process could be improved by using other features, because the SIFT method could not extract some interesting points from images with complex natural backgrounds. Hence, the classification rates in these cases were low.

Although our system performs very well in translating hand sign, the final goal is to translate continuous sign language. Some research studies are already investigating continuous sign language translation [59–61]. Our future plan is to embed our system into a continuous sign language translation system.

One of the disadvantages of the system might be from the transformation from 3D images to 2D images. Our future work will consider including 3D information in the acquisition process of the data set before implementing it into the translation system.

Acknowledgments: The authors would like to thank Thailand Research Fund under the Royal Golden Jubilee Ph.D. Program and the Chiang Mai University (Grant No. PHD/0044/2555) for financial support.

Author Contributions: All authors conceived and designed the experiments; Atcharin Klomsae performed the experiments; and all authors contributed to the writing of the paper.

Conflicts of Interest: The authors of the paper do have any conflict of interest with any companies or institutions. This work was supported by the Thailand Research Fund under the Royal Golden Jubilee Ph.D. Program and

the Chiang Mai University (Grant No. PHD/0044/2555). This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Vamplew, P. Recognition of sign language gestures using neural networks. In Proceedings of the 1st European Conference on Disability, Virtual Reality and Associated Technologies, Maidenhead, UK, 8–10 July 1996.
2. Waldron, M.B. Isolated ASL sign recognition system for deaf persons. *IEEE Trans. Rehabil. Eng.* **1995**, *3*, 261–271. [CrossRef]
3. Liang, R.H.; Ouhyoung, M. A real-time continuous gesture recognition system for sign language. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 558–565.
4. Vogler, C.; Metaxas, D. Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Orlando, FL, USA, 12–15 October 1997; pp. 156–161.
5. Fels, S.S.; Hinton, G.E. Glove-talk: A neural network interface between a dataglove and a speech synthesizer. *IEEE Trans. Neural Netw.* **1993**, *4*, 2–8. [CrossRef] [PubMed]
6. Su, M.C.; Jean, W.F.; Chang, H.T. A static hand gesture recognition system using a composite neural network. In Proceedings of the IEEE 5th International Conference on Fuzzy Systems, New Orleans, LA, USA, 11 September 1996; pp. 786–792.
7. Wu, J.Q.; Gao, W.; Song, Y.; Liu, W.; Pang, B. A Simple sign language recognition system based on data glove. In Proceedings of the International Conference on Signal Processing Proceedings (ICSP), Beijing, China, 12–16 October 1998; pp. 1257–1260.
8. Gao, W.; Fang, G.; Zhao, D.; Chen, Y. *A Chinese Sign Language Recognition System Based on SOFM/SRN/HMM*; Pattern Recognition: New York, NY, USA, 2004; Volume 37, pp. 2389–2402.
9. Ibarguren, A.; Maurtua, I.; Sierra, B. Layered architecture for real time sign recognition: Hand gesture and movement. *Eng. Appl. Artif. Intell.* **2012**, *23*, 1216–1228. [CrossRef]
10. Oz, C.; Leu, M.C. American sign language word recognition with a sensory glove using artificial neural networks. *Eng. Appl. Artif. Intell.* **2011**, *24*, 1204–1213. [CrossRef]
11. Min, B.W.; Yoon, H.S.; Soh, J.; Yang, Y.M.; Ejima, T. Hand gesture recognition using hidden Markov models. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Orlando, FL, USA, 12–15 October 1997; pp. 4232–4235.
12. Mohamed, M.; Mohamed, D. Arabic sign language recognition by decisions fusion using dempster-shafer theory of evidence. In Proceedings of the Computing, Communications and IT Applications Conference (ComComAp), Hong Kong, China, 1–4 April 2013.
13. Holden, E.J.; Owens, R.; Roy, G.G. Adaptive fuzzy expert system for sign recognition. In Proceedings of the International Conference on Signal and Image Processing (SIP'2000), Las Vegas, NV, USA, November 1999.
14. Grobel, K.; Assan, M. Isolated sign language recognition using hidden Markov models. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Orlando, FL, USA, 12–15 October 1997; pp. 162–167.
15. Alon, J.; Athisos, V.; Yuan, Q.; Sclaroff, S. Simultaneous localization and recognition of dynamic hand gestures. In Proceedings of the Seventh IEEE Workshops on Application of Computer Vision, Breckenridge, CO, USA, 5–7 January 2005; pp. 254–260.
16. Auephanwiriyakul, S.; Chaisatian, P. Static hand gesture translation using string grammar hard C-means. In Proceedings of the Fifth International Conference on Intelligent Technologies, Houston, Texas, USA, December 2004.
17. Just, A.; Marcel, S. A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition. *Comput. Vis. Image Underst.* **2009**, *113*, 532–543. [CrossRef]
18. Starner, T.; Weaver, J.; Pentland, A. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1271–1375. [CrossRef]
19. Huang, C.L.; Huang, W.Y. Sign language recognition using model-based tracking and a 3D Hopfield neural network. *Mach. Vis. Appl.* **1998**, *10*, 292–307. [CrossRef]

20. Kobayashi, T.; Haruyama, S. Partly-hidden Markov model and its application to gesture recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Proceessing (ICASSP), Munich, Germany, 21–24 April 1997; pp. 3081–3084.
21. Chen, F.S.; Fu, C.M.; Huang, C.L. Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image Vis. Comput.* **2003**, *21*, 745–758. [CrossRef]
22. Yang, R.; Sarkar, S. Coupled grouping and matching for sign and gesture recognition. *Comput. Vis. Image Underst.* **2009**, *113*, 663–681. [CrossRef]
23. Prisacariu, V.A.; Reid, I. 3D hand tracking for human computer interaction. *Image Vis. Comput.* **2012**, *30*, 236–250. [CrossRef]
24. Han, J.; Awad, G.; Sutherland, A. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognit. Lett.* **2009**, *30*, 623–633. [CrossRef]
25. Kelly, D.; McDonald, J.; Markham, C. A person independent system for recognition of hand postures used in sign language. *Pattern Recognit. Lett.* **2012**, *31*, 1359–1368. [CrossRef]
26. Gamage, N.; Kuang, Y.C.; Akmeliawati, R.; Demidenko, S. Gaussian process dynamical models for hand gesture interpretation in sign language. *Pattern Recognit. Lett.* **2011**, *32*, 2009–2014. [CrossRef]
27. Zaki, M.M.; Shaheen, S.I. Sign language recognition using a combination of new vision based features. *Pattern Recognit. Lett.* **2011**, *32*, 572–577. [CrossRef]
28. Adithya, V.; Vinod, P.R.; Usha, G. Artificial neural network based method for Indian sign language recognition. In Proceedings of the 2013 IEEE Conference on Information and Communication Technologies (ICT 2013), Thuckalay, Tamil Nadu, India, 11–12 April 2013.
29. Lubo, G.; Xin, M.; Haibo, W.; Jason, G.; Yibin, L. Chinese sign language recognition with 3D hand motion trajectories and depth images. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014.
30. Ching-Hua, C.; Eric, R.; Caroline, G. American sign language recognition using Leap motion sensor. In Proceedings of the 13th International Conference on Machine Learning and Applications, Detroit, MI, USA, 3 December 2014.
31. Md, A.U.; Shayhan, A.C. Hand sign language recognition for Bangla alphabet using support vector machine. In Proceedings of the International Conference on Innovations in Science, Engineering and Technology (ICISET), Dhaka, Bangladesh, 28–29 October 2016.
32. Washef, A.; Kunal, C.; Soma, M. Vision based hand gesture recognition using dynamic time warping for Indian sign language. In Proceedings of the International Conference on Information Science (ICIS), Kochi, India, 12–13 August 2016.
33. Pariwat, T.; Seresangtakul, P. Thai finger-spelling sign language recognition using global and local features with SVM. In Proceedings of the 9th International Conference on Knowledge and Smart Technology (KST), Chonburi, Thailand, 1–4 February 2017.
34. Sriboonruang, Y.; Kumhom, P.; Chamnongthai, K. Hand posture classification using wavelet moment invariant. In Proceedings of the IEEE International Conference on Virtual Environments, Human–Computer Interfaces and Measurement Systems, Boston, MA, USA, 12–14 July 2004; pp. 78–82.
35. Shen, X.; Hua, G.; Williams, L.; Wu, Y. Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields. *Image Vis. Comput.* **2012**, *30*, 227–235. [CrossRef]
36. Fatma, G.; Tahani, B.; Olfa, J.; Mourad, Z.; Chokri, B.A. Arabic sign language recognition system based on wavelet network. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016.
37. Md., M.I.; Sarah, S.; Jawata, A. Real time hand gesture recognition using different algorithms based on American sign language. In Proceedings of the 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition, Dhaka, Bangladesh, 13–14 February 2017.
38. Lee, Y.H.; Tsai, C.Y. Taiwan sign language (TSL) recognition based on 3D data and neural networks. *Expert Syst. Appl.* **2009**, *36*, 1123–1128. [CrossRef]
39. Wenwen, Y.; Jinxu, T.; Changfeng, X.; Zhongfu, Y. Sign language recognition system based on weighted hidden Markov model. In Proceedings of the 2015 8th International Symposium on Computational Intelligence and Design, Hangzhou, China, 12–13 December 2015.

40. Sérgio, B.C.; Edson, D.F.M.S.; Talles, M.A.B.; José, O.F.; Symone, G.S.A.; Adson, F.D.R. Static gestures recognition for Brazilian sign language with Kinect sensor. In Proceedings of the 2016 IEEE Sensors, Orlando, FL, USA, 30 October–3 November 2016.
41. Al-Rouson, M.; Assaleh, K.; Tala'a, A. Video-based signer-independent Arabic sign language recognition using hidden Markov models. *Appl. Soft Comput.* **2009**, *9*, 990–999. [CrossRef]
42. Phitakwinai, S.; Auephanwiriyakul, S.; Theera-Umpon, N. Thai sign language translation using fuzzy C-means and scale invariant feature transform. *Lect. Notes Comput. Sci.* **2008**, *5073*, 1107–1119.
43. Auephanwiriyakul, S.; Phitakwinai, S.; Suttapak, W.; Chanda, P.; Theera-Umpon, N. Thai sign language translation using scale invariant feature transform and hidden Markov models. *Pattern Recognit. Lett.* **2013**, *34*, 1291–1298. [CrossRef]
44. Abdelbaky, A.; Saleh, A. Appearance-based Arabic sign language recognition using hidden Markov models. In Proceedings of the 2014 International Conference on Engineering and Technology (ICET), Cairo, Egypt, 19–20 April 2014.
45. Kian, M.L.; Alan, W.C.T.; Shing, C.T. Block-based histogram of optical flow for isolated sign language recognition. *J. Vis. Commun. Image Represent.* **2016**, *40*, 538–545.
46. Zahedi, M.; Keysers, D.; Deselaers, T.; Ney, H. Combination of tangent distance and an image distortion model for appearance-based sign language recognition. *Lect. Notes Comput. Sci.* **2005**, *3663*, 401–408.
47. Office of the Basic Education Commission. *Thai Hand Sign Language Handbook under the Initiatives of Her Royal Highness Princess Maha Chakri Sirindhorn*; Department of General Education: Bangkok, Thailand, 1997.
48. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
49. Keller, J.M.; Gray, M.R.; Givens, J.A. A fuzzy K-nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.* **1985**, *4*, 580–585. [CrossRef]
50. Fu, K.S.; Lu, S.Y. A clustering procedure for syntactic patterns. *IEEE Trans. Syst. Man Cybern.* **1977**, *7*, 734–742. [CrossRef]
51. Juan, A.; Vidal, E. On the use of normalized edit distances and an efficient k-NN search technique (k-AESA) for fast and accurate string classification. In Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 3–7 September 2000; pp. 676–679.
52. Bezdek, J.C.; Keller, J.; Krishnapuram, R.; Pal, N.R. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1999.
53. Fu, K.S. *Syntactic Pattern Recognition and Applications*; Prentice-Hall: Bergen, NJ, USA, 1982.
54. Yang, M.S.; Wu, K.L. Unsupervised possibilistic clustering. *Pattern Recognit.* **2006**, *39*, 5–21. [CrossRef]
55. Kohonen, T. Median strings. *Pattern Recognit. Lett.* **1985**, *3*, 309–313. [CrossRef]
56. De la Higuera, C.; Casacuberta, F. The topology of strings: Two np-complete problems. *Theor. Comput. Sci.* **2000**, *230*, 39–48. [CrossRef]
57. Martinez, C.D.; Juan, A.; Casacuberta, F. Use of median string for classification. In Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 3–7 September 2000; pp. 903–906.
58. Klomsae, A.; Auephanwiriyakul, S.; Theera-Umpon, N. A String grammar fuzzy-possibilistic C-medians. *Appl. Soft Comput.* **2017**, *57*, 684–695. [CrossRef]
59. Camgoz, N.C.; Hadfield, S.; Koller, O.; Bowden, R. SubUNets: End-to-end hand shape and continuous sign language recognition. In Proceedings of the 3rd International Workshop on Observing and Understanding Hands (International Conference on Computer Vision (ICCV)), Venice, Italy, 22–29 October 2017; pp. 222–227.
60. Koller, O.; Zargaran, S.; Ney, H.; Bowden, R. Deep Sign: Hybrid CNN-HMM for continuous sign language recognition. In Proceedings of the British Machine Vision Conference, York, UK, 19–22 September 2016.
61. Schmidt, C.A. Handling Multimodality and Scarce Resources in Sign Language Machine Translation. Ph.D. Thesis, Computer Science Department, RWTH Aachen University, Aachen, Germany, 2016.