Tanzania Journal of Science 47(2): 769-778, 2021
ISSN 0856-1761, e-ISSN 2507-7961

Sign Language Recognition Using Kinect Sensor Based on Color Stream and Skeleton Points

Isack Bulugu
University of Dar es Salaam, College of Information and Communication Technologies, Department of Electronics and Telecommunications, P. O. Box 33335 Dar es Salaam, Tanzania
E-mail: bulugu@udsm.ac.tz
Received 16 Feb 2021, Revised 25 May 2021, Accepted 26 May 2021, Published May 2021
DOI: https://dx.doi.org/10.4314/tjs.v47i2.32

Abstract

This paper presents a sign language recognition system based on color stream and skeleton points. Several approaches have been established to address sign language recognition problems. However, most of the previous approaches still have poor recognition accuracy. The proposed approach uses Kinect sensor based on color stream and skeleton points from the depth stream to improved recognition accuracy. Techniques within this approach use hand trajectories and hand shapes in combating sign recognition challenges. Therefore, for a particular sign a representative feature vector is extracted, which consists of hand trajectories and hand shapes. A sparse dictionary learning algorithm, Label Consistent K-SVD (LC-KSVD) is applied to obtain a discriminative dictionary. Based on that, the system was further developed to a new classification approach for better results. The proposed system was fairly evaluated based on 21 sign words including one-handed signs and two-handed signs. It was observed that the proposed system gets high recognition accuracy of 98.25%, and obtained an average accuracy of 95.34% for signer independent recognition.

Keywords: Sign language, Color stream, Skeleton points, Kinect sensor, Discriminative dictionary.

Introduction

In recent years, the use of hand movements, especially sign languages, serves as a motivating force for research in sign language modeling, analysis and recognition. Although sign languages are complicated to model since the meanings of signs depend on people and cultures, a set of specific sign language vocabulary can always be predefined in many applications, such as sign language systems, so that the ambiguity can be limited. Sign language is used intentionally bridge the gap to allow hearing impaired and dumb person and communities to communicate with the society. It is non-verbal language that translates one's meaning which involves positioning of the fingers, hands, arms, head and body, (Cheok et al. 2019). This automatically makes possible the communication between deaf-mute and normal people via computer. However, sign language recognition is still a challenging task since it involves both manual parameters such as hand-shapes, orientations, locations and movements, and non-manual parameters such as facial expression, head and body pose, and gaze. Signs performed by one hand are also known as one-handed signs, while the corresponding hand is the dominant hand; others performed by two hands are named with two-handed signs. Besides, the variations caused by different signers aggravate the difficulty of recognition.

Most of the previous systems use gloves which are attached with sensors to gather sign data and usually can get satisfactory results (Liang and Ouhyoung 1998, Gao et al. 2000). But the unnaturalness and complexity limit their wide-spreading. Vision-based methods (Fatmi et al. 2017, Raheja et al. 2016) encounter the dimension-reducing projection and occlusion problems and often lead to poor accuracy. Recently, Microsoft Kinect sensor, for its ability to capture depth information, is becoming an active trend in the sign language recognition (SLR) and gesture recognition (GR) community (Chai et al. 2013, Pöhlmann et al. 2016, Huang et al. 2018, Camgoz et al. 2018, Jing et al. 2019).

Since Microsoft launched the Kinect in 2010, owing to its affordable price and flexibility in data acquisition, many research projects and applications about it have begun to surface. Among the projects, SLR or GR is one of the main active topics. Chai et al. 2013) proposed a Kinect-based American Sign Language recognition system via Hidden Markov Model (HMM) method and compared to their existing system, CopyCat, results suggest that the Kinect may be a viable option for sign verification. Reyes et al. (2011) resented a gesture recognition approach based on a feature weighting Dynamic Time Warping (DTW) algorithm, they used the extracted skeleton joints from depth image as feature vectors (Camgoz et al. 2018). Using skeleton joints, Patsadu et al. (2012) made comparisons among different classification methods: back-propagation neural network (BPNN), Support Vector Machine (SVM), decision tree and naïve Bayes. The average accuracy they acquired is 93.72%, confirming the high potential of using Kinect in human body recognition applications (Pöhlmann et al. 2016).

Standard machine learning algorithms like HMM, DTW, and SVM are the common methods used in the SLR domain. Recently, sparse coding-based pattern recognition has been widely extended. Zhang and Li (2010) resented a sparse-representation-based face recognition scheme with a discriminative K-SVD (D-KSVD) method (Liu et al. 2016). An over complete dictionary was constructed using a set of spatio-temporal descriptors to model and identify human actions (Guha and Ward 2012). The sparse representation and compressed sensing has been applied into gesture recognition problems by Kemeng et al. (2015). Jiang et al. (2011) proposed a novel algorithm, LC-KSVD, for learning a reconstructive and discriminative dictionary. Experimental results show the learned dictionary, which can be small and single unified, is very effective in face recognition and object recognition.

The proposed solution has one new approach that uses sparse coding (SC). Then, for an input signal y, SC approximates it by a sparse linear combination from a learned or given dictionary D. When adding discriminative information into the dictionary learning procedure, sparse coding shows the property of being suitable for pattern recognition problems such as: face recognition, action recognition and gesture recognition (Guha and Ward 2012, Donahue et al. 2015, Mahmoud et al. 2016).

In this paper, the proposed approach takes advantage of the classifications properties of sparse coding, and employs them into a sign language recognition framework. Figure 1 shows the overview of the proposed system. The Kinect sensor was used as a sign data acquisition device. The color image was obtained as well as the skeleton frame from depth image by Kinect SDK. Then a representative feature vector composed of both hand trajectories and hand shapes were extracted. In the training phase, a discriminative sparse dictionary learning algorithm was applied, Label Consistent K-SVD (LC-KSVD) (Liang and Ouhyoung 1998), and then proposed a new classification approach which was proven to get better results. The main contributions of this paper are:

- Combining the sparse coding based recognition and RGB-D sensor into SLR domain.
- A representative sign descriptor.

- An accurate classifier designed for the learned dictionary.

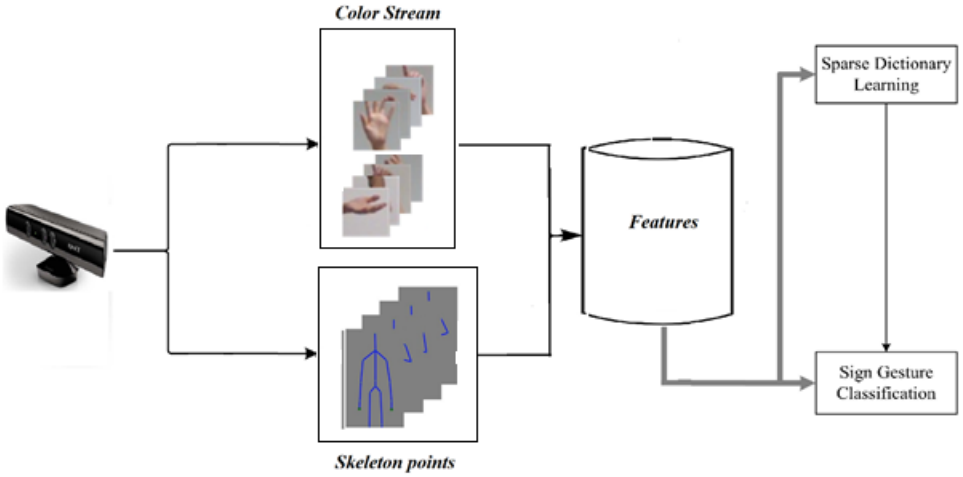

Figure 1: System overview.

Materials and Methods

Data acquisition

Data streams from Kinect SDK include color stream, depth stream and skeleton stream. The skeleton stream is estimated from the depth stream (Azis et al. 2015). A skeleton stream frame can track skeleton joints up to six persons. In the proposed system, it was assumed that in the system there was only one person standing in front of the Kinect to perform signs, which was quite reasonable. Every single skeleton can include a maximum of 20 skeleton joints, which consist of 3D Cartesian coordinates, at a frame rate of approximately 30 FPS. Besides, the color image obtained from Kinect is 480 × 480 pixels at the same frame rate as skeleton image.

Feature extraction

Sign linguists generally distinguish the basic components (or phoneme subunits) of a sign gesture as consisting of the hand shape, hand orientation, location and movement (Angelopoulou et al. 2019). Followed by this conception, the feature description in the proposed system is composed of 3D hand trajectory and hand shape information.

Hand trajectory

Among all the 20 skeleton joints, only two hand joints were particularly interested: Right Hand (RH) and Left Hand (LH). Combine these two hand points, a primitive 3D trajectory vector can be obtained for a given sign

$$\boldsymbol{t_0} = [x_1^R, y_1^R, z_1^R, \dots, x_{N_0}^R, y_{N_0}^R, z_{N_0}^R, x_1^L, y_1^L, z_1^L, \dots, x_{N_0}^L, y_{N_0}^L, z_{N_0}^L]^T \quad (1)$$

where N_0 denotes total frames from the sign beginning to the end of a sign. Triplet x_i^R, y_i^R, z_i^R represents the Cartesian coordinate of the right hand in the i^{th} frame and similarly, x_i^L, y_i^L, z_i^L, corresponds to the left hand. Operator $[\dots]^T$ denotes transposition operation. The primitive hand trajectory vector cannot be directly used in further steps, it is first necessary to perform normalization. Trajectory is normalized in three steps: speed, scale and position.

Speed normalization: To eliminate the difference of hand speed caused by different signers, a linear resampling (Jing et al. 2019) was applied. For a trajectory $\boldsymbol{t_0}$ of $2N_0$ points, both hand were linear resampled to $2N_f$ points

in 3D space, where N_f denotes the fixed resampling points number and in the experiment was set to 100, leading to a moderate sampling accuracy and complexity. The distance between each two adjacent points is equal to L/N_f after the resampling, where L is the accumulated length of $\boldsymbol{t_0}$.

Scale normalization: Since signers vary in different heights and different positions or angles from Kinect, huge variations arise in trajectories' scales and positions. Here, scale normalization was achieved by a coordinate converting method (Liang and Ouhyoung 1998). To the resampled trajectory, it was first shifted to body coordinate system. That is to say, the spine position was taken as an origin of coordinate. Then trajectory points were converted from Cartesian coordinates to spherical coordinates, making triplet (x, y, z) to(d, θ, φ). The converting formulas were

$$d = \sqrt{x^2 + y^2 + z^2}, \quad (2)$$

$$\theta = \arccos(\tfrac{z}{d}), \quad (3)$$

$$\varphi = \arctan(\tfrac{y}{x}), \quad (4)$$

The advantage of spherical coordinate is that the polar angle θ and azimuthal angle φ do not contribute to the scale. Thus, using the distance from head to origin as a scaling factor, it scales every diameter d to make the same signs have comparable magnitude.

Position normalization: To make trajectory points position invariant, the resampled and scaled trajectory points were further converted from spherical coordinates back to Cartesian coordinates. Then all points were shifted to the first one, that is, subtracting each point's coordinate by the first point, for each hand, respectively. This step makes the trajectory well aligned, apart from the interference caused by the position of the first point. The final normalized 3D trajectory vector is as given in Equation (5).

$$t = [x_1'^R, y_1'^R, z_1'^R, \dots, x'^R_{N_f}, y'^R_{N_f}, z'^R_{N_f}, x_1'^L, y_1'^L, z_1'^L, \dots, x'^L_{N_f}, y'^L_{N_f}, z'^L_{N_f}]^t \quad (5)$$

Hand shape

Hand shape describes more regional information which hand trajectory cannot distinguish, as hand trajectory represents a more global feature. In this paper, Histograms of Oriented Gradients (HOG) (Yang and Wang 2015, Sharma et al. 2020) is considered as a hand shape feature. For HOG can well describe the shape and appearance of an object and also adapt to illumination variation or complex background. The positions of two hands from the skeleton frame can be obtained; this provides prior information for hand segmentation in color image obtained from Kinect. Centered at hand position, a square area which is linear to depth is covered on color image to crop hand region. The equation was generated using MATLAB fitting based on linear regression (LG). LG is a powerful tool for investigating the relationships between multiple variables by relating one variable to a set of variables. It can identify the effect of hand depth (Z) while adjusting for square size (L) differences. Based on several experimental tests, an empirical linearity formula between square size L and hand depth Z is

$$L = -50Z + 144.45 \quad (6)$$

When a hand moves towards the Kinect and depth $\boldsymbol{Z}$ decreases, the cropping window size will increase. After getting the cropped hand area image, a resizing operation is applied, making it into a fixed size patch of 32 × 32. Then HOG descriptors were computed for the 32 × 32 hand patch in each sign frame, resulting in a set of vectors $\boldsymbol{S_1} \dots \boldsymbol{S_N}$, where $\boldsymbol{S_i}$, $i = 1 \dots N$, denotes the HOG descriptor of i^{th} frame. To reduce dimensionality, an "average hand" was used to represent hand's spatial-temporal information.

$$\bar{s} = \tfrac{1}{N}\textstyle\sum_{i=1}^{N} S_i \quad (7)$$

Operating on two hands, it can generate $\bar{\boldsymbol{s}}_{\boldsymbol{r}}$ and $\bar{\boldsymbol{s}}_{\boldsymbol{l}}$ for right hand and left hand, respectively. The hand shape descriptor $\boldsymbol{S}$ is obtained by concatenating $\bar{\boldsymbol{h}}_{\boldsymbol{r}}$ and $\bar{\boldsymbol{h}}_{\boldsymbol{l}}$:

$$\boldsymbol{S} = [\, \bar{\boldsymbol{s}}_{\boldsymbol{r}}, \bar{\boldsymbol{s}}_{\boldsymbol{l}} \,]^t. \quad (8)$$

Finally, the feature vector of a given sign

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{S} \end{bmatrix} \quad (9)$$

Sparse coding based recognition

In the sparse coding based recognition framework, a sparse dictionary needs to be firstly constructed from the extracted features of each sign. Unlike the method in (Pöhlmann et al. 2016), which directly uses the training samples as the dictionary, here it was built through a learning methodology. The advantage of learning a dictionary is that the resulting dictionary can be compact and small even when the training samples are large and numerous, at the same time, keeping the dictionary reconstructive. This is important when there are limited computational resources.

The well-known K-SVD algorithm (Kemeng et al. 2015) is efficient for learning the dictionary D by solving

$$\{D, A\} = argmin\|Y - DA\|_2^2 \; s.t. \forall i \|a_i\|_0 \leq T \,, \qquad (10)$$

where $Y = [y_1 \dots y_N] \in \Re^{nXN}$ is a set of n-dimensional N input feature signals, and $D = [d_1 \dots d_K] \in \Re^{nXK}$ is the learned dictionary, $A = [\alpha_1 \dots \alpha_N] \in \Re^{KXN}$ is the N sparse codes of feature vectors Y, and T is a sparsity factor.

Then the sparse code α for an unknown input y is computed as

$$\alpha = arg\min_\alpha \|y - D\alpha\|_2^2 \; s.t. \forall i \|\alpha\|_0 \leq T, \qquad (11)$$

which can be solved by the Orthogonal Matching Pursuit (OMP) algorithm (Zhang et al. 2016). The dictionary learned in this way is reconstructive but makes the computed sparse code α not very suitable for recognition task. A modified version of K-SVD which makes the learned dictionary reconstructive and discriminative, LC-KSVD (Jiang et al. 2011), is applied in our system. It will first be briefly introduced, and then followed by a proposed classification approach.

Dictionary learning using label consistent K-SVD

An objective function is defined by adding a label consistency regularization term and a joint classification error into Equation (10).

$$\{D, W, L, A\} = arg \min_{D,W,L,A} \|Y - DA\|_2^2 + \eta \, \|Q - LA\|_2^2 + \lambda \|H - WA\|_2^2 \; s.t. \; \forall i, \|\alpha_i\|_0 \leq T \qquad (12)$$

where the second term $\|Q - LA\|_2^2$ represents the discriminative sparse code error, $Q = [q_1 \dots q_N] \in \Re^{KxN}$ are the discriminative sparse codes of Y. $q_i \in \Re^K$ is a discriminative sparse code corresponding to input y_i if the nonzero values(typically 1) of q_i occur at those indices where y_i and the dictionary atom d_k share the same label. L is a linear transformation matrix which transforms the original sparse code α to be most discriminative in sparse feature space $\Re^K$. The third term $\|H - WA\|_2^2$ represents the classification error, W denotes the classifier parameters. $H = [h_1 \dots h_N] \in \Re^{mxN}$ are the class labels of inputs Y, $h_i \in \Re^m$ is a label vector corresponding to y_i, where the nonzero position indicates the sign class of y_i. η and λ are the parameters controlling the relative contribution of the corresponding terms. The second term forces atoms of the same class in dictionary D having very similar sparse codes and those of different classes having discriminative sparse codes. The third term forces each atom in D having its corresponding class label.

The K-SVD algorithm can solve the optimization problem of Equation (12). Rewrite Equation (12) as

$$\{D, W, L, A\} = arg\ \min_{D,W,L,A} \left\| \begin{pmatrix} Y \\ \sqrt{\eta Q} \\ \sqrt{\lambda H} \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\eta L} \\ \sqrt{\lambda W} \end{pmatrix} A \right\|_2^2 s.t.\ \forall i, \|\alpha_i\|_0 \le T \qquad (13)$$

Let $Y_{new} = (Y^t, \sqrt{\eta Q^t}, \sqrt{\lambda H^t})^t$, $D_{new} = (D^t, \sqrt{\eta L^t}, \sqrt{\lambda W^t})^t$. So Equation (13) is equivalent by solving

$$\{D_{new}, A\} = argmin \|Y_{new} - D_{new} A\|_2^2\ s.t. \forall i \|a_i\|_0 \le T \qquad (14)$$

The classification approach

Original method: After employing the LC-KSVD algorithm, $\boldsymbol{D_{new}}$ is obtained. Splitting the joint dictionary $\boldsymbol{D_{new}}$ in $\boldsymbol{D} = [\boldsymbol{d_1} \dots \boldsymbol{d_K}]$, $L = [l_1 \dots l_K]$, and $\boldsymbol{W} = [\boldsymbol{w_1} \dots \boldsymbol{w_K}]$, then a re-computation is done to get

$$\widehat{\boldsymbol{D}} = \left[\frac{\boldsymbol{d_1}}{\|\boldsymbol{d_1}\|_2} \dots \frac{\boldsymbol{d_1}}{\|\boldsymbol{d_K}\|_2}\right], \widehat{\boldsymbol{L}} = \left[\frac{l_1}{\|\boldsymbol{d_1}\|_2} \dots \frac{l_1}{\|\boldsymbol{d_K}\|_2}\right], \widehat{\boldsymbol{W}} = \left[\frac{w_1}{\|d_1\|_2} \dots \frac{d_1}{\|d_K\|_2}\right] \qquad (15)$$

since each column in $\boldsymbol{D_{new}}$ is L_2-normalized. A simply linear predictive classifier $f(\widehat{\boldsymbol{a}}; \widehat{\boldsymbol{W}}) = \widehat{\boldsymbol{a}}\widehat{\boldsymbol{W}}$ is used in (Liang and Ouhyoung 1998), $\hat{\alpha}$ is the sparse code of a test input y with dictionary $\widehat{\boldsymbol{D}}$ by OMP algorithm. The index of largest element in $f(\widehat{\boldsymbol{a}}; \widehat{\boldsymbol{W}})$ is selected as the class of y.

Proposed method: When learning dictionary$\boldsymbol{D}$, it is by solving the object function of Equation (12). That is, optimizing $\boldsymbol{D}$ to make the sum of three terms$\|\boldsymbol{Y} - \boldsymbol{DA}\|_2^2, \eta\ \|\boldsymbol{Q} - \boldsymbol{LA}\|_2^2$, and $\lambda\|H - WA\|_2^2$ minimum. However, in the original method it is only by minimizing $\|\boldsymbol{y} - \boldsymbol{D\alpha}\|_2^2$ when compute sparse code $\widehat{\boldsymbol{\alpha}}$, which only corresponds to the first term of Equation (12). The information contained in L and W is lost in the process of computing $\widehat{\boldsymbol{\alpha}}$**.** In some cases, the term $\|\boldsymbol{Y} - \boldsymbol{DA}\|_2^2$, does not reach a minimum value (the summation value of three terms does) when learning $\boldsymbol{D}$, so the computation of $\widehat{\boldsymbol{\alpha}}$ by minimizing $\|\boldsymbol{y} - \boldsymbol{D\alpha}\|_2^2$ is not optimal. For the classification method, the entire $\boldsymbol{D_{new}}$ was used as the learned dictionary in the process of computing sparse codes, aiming to complement the losing information. To achieve this, one needs to re-construct $\widetilde{\boldsymbol{Y}}_{\boldsymbol{new}}$. Given an input sign feature y first makes m duplications of it yielding $\widetilde{\boldsymbol{Y}} = \underbrace{[y \dots y]}_{m}$, assuming the $j^{th}(1 \le j \le m)$ column of $\widetilde{\boldsymbol{Y}}$ belongs to the j^{th}sign class. Based on this assumption, then build the corresponding discriminative matrix $\widetilde{\boldsymbol{Q}} = [\widetilde{\boldsymbol{q}}_{\boldsymbol{1}} \dots \widetilde{\boldsymbol{q}}_{\boldsymbol{m}}] \in \Re^{Kxm}$, which makes each column of $\widetilde{\boldsymbol{Y}}$ have discriminative sparse code, and label matrix $\widetilde{\boldsymbol{H}} = [\widetilde{\boldsymbol{h}}_{\boldsymbol{1}} \dots \widetilde{\boldsymbol{h}}_{\boldsymbol{m}}] \in \Re^{mxm}$ (here is an identity matrix with size m), which guarantees that j^{th}column of $\widetilde{\boldsymbol{Y}}$ belongs to j^{th}sign class. The way of building $\widetilde{\boldsymbol{Q}}$ and $\widetilde{\boldsymbol{H}}$ is similar in the previous sub-section. For example, assuming the sign classes are $m = 3$ and $\boldsymbol{D} = [\boldsymbol{d_1} \dots \boldsymbol{d_6}]$ and $\widetilde{\boldsymbol{Y}} = [\boldsymbol{y}(1) \dots \boldsymbol{y}(3)]$, where $\boldsymbol{y}(1)$, $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ belong to class 1, $y(2)$, $\boldsymbol{d_3}$ and $\boldsymbol{d_4}$ belong to class 2, and $y(3)$, $\boldsymbol{d_5}$ and $\boldsymbol{d_6}$ belong to class 3, $\widetilde{\boldsymbol{Q}}$ can be defined as $\widetilde{\boldsymbol{Q}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$, $\widetilde{\boldsymbol{H}}$ can be defined as $\widetilde{\boldsymbol{H}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

By jointing $\widetilde{\boldsymbol{Y}}$, $\widetilde{\boldsymbol{Q}}$ and $\widetilde{\boldsymbol{H}}$, the re-constructed $\widetilde{\boldsymbol{Y}}_{\boldsymbol{new}} = (\widetilde{\boldsymbol{Y}}^t, \sqrt{\eta\widetilde{\boldsymbol{Q}}^t}, \sqrt{\lambda\widetilde{\boldsymbol{H}}^t})^t$ is obtained. Note that each column term in $\widetilde{\boldsymbol{Y}}_{\boldsymbol{new}}$ has its corresponding class label. Through OMP algorithm the sparse codes can be computed using $\widetilde{\boldsymbol{A}} = [\widetilde{\boldsymbol{\alpha}}(1) - \widetilde{\boldsymbol{\alpha}}(m)]$ for $\widetilde{\boldsymbol{Y}}_{\boldsymbol{new}}$ with joint dictionary $\boldsymbol{D_{new}}$. Then the predictive class labels $\widehat{\boldsymbol{H}} = [\widehat{\boldsymbol{h}}_{\boldsymbol{1}} \dots \widehat{\boldsymbol{h}}_{\boldsymbol{m}}]$ are estimated as

$$\widehat{\boldsymbol{H}} = \boldsymbol{W}\widetilde{\boldsymbol{A}}, \qquad (16)$$

where $\boldsymbol{W}$ is classification matrix split from $\boldsymbol{D_{new}}$. After applying Equation (16), each term in $\widehat{\boldsymbol{H}}$ is similar to the corresponding term in $\widetilde{\boldsymbol{H}}$ for label matrix $\widetilde{\boldsymbol{H}}$ and discriminative matrix $\widetilde{\boldsymbol{Q}}$ have forced each column in $\widetilde{\boldsymbol{Q}}$ to get its class label. One can readily find out that the real class $\boldsymbol{c}$ which $\boldsymbol{y}$ belongs to lead the predictive class label $\widehat{\boldsymbol{h}}_{\boldsymbol{c}}$ has the most similarity with label

vector $\widetilde{\boldsymbol{h}}_{\boldsymbol{c}}$ than other "fake" classes. So, the class c of sign feature $\boldsymbol{y}$ is represented by

$$c = arg\min_i \left\|\widehat{\boldsymbol{h}}_{\boldsymbol{i}} - \widetilde{\boldsymbol{h}}_{\boldsymbol{i}}\right\|_2^2, 1 \le i \le m \quad (17)$$

Results and Discussions

To evaluate the proposed system, a database consisting of 21 sign words was created. Table 1 shows the details of all the signs. Database was performed by eight different individuals; each sign had 57 samples, approximately 7 samples of one sign per person. The total sign samples were 1197. A threshold based on the spine position was defined to indicate a sign's beginning and ending. When both hands move above the threshold, the system begins to record a sign, and when both hands move below the threshold, the system stops the recording.

Table 1: Signs in the database

Classification of signs	One-handed signs	Two-handed signs
Sign name	Sorry, Sky, Power, Need, Or, Responsible, Hope, Very, Airplane, Recognize, Everyone, Good-bye	Snow, Entrust, Welcome, Help, Warm, Fly, Staunch, Surrender, Happy

A sign's trajectory feature was the result of a 600 dimension vector (for both hands), HOG feature is of 504 dimension. Then the dimension of feature vector y for sign s was 1104. The dictionary consisted of 210 atoms, that is, 10 atoms per sign. The controlling parameters η and λ were chosen by a cross validation strategy and select the one which gave best performance. $\sqrt{\eta}$ and $\sqrt{\lambda}$ was set to a range from 0.1 to 1. The sparsity factor T was set to 20 in all experiments.

Firstly, three different features were tested: only trajectory feature, only HOG feature, and trajectory and HOG feature. The sign database was split into two parts (randomly select half samples of each sign), one part for training (588 samples) and the other for testing (609 samples). Table 2 shows the results. The results showed that the system can still acquire good performance when using only trajectory, but the accuracy drops evidently when using only HOG, which indicates hand trajectory contains more discriminative information than hand shape in the database. The confusion matrix when using both trajectory and HOG feature is shown in Figure 2.

Table 2: Recognition results for different sign features

Feature set	Accuracy
Only trajectory ($\sqrt{\boldsymbol{\eta}}$ -0.2, $\sqrt{\boldsymbol{\lambda}}$ -0.1)	95.40%
Only HOG ($\sqrt{\boldsymbol{\eta}}$ = 0.1, $\sqrt{\boldsymbol{\lambda}}$ = 0.1)	82.43%
Trajectory & HOG ($\sqrt{\boldsymbol{\eta}}$ = 0.2, $\sqrt{\boldsymbol{\lambda}}$ = 0.3)	98.24%

To measure the classification ability of the proposed classification method, comparisons were made with the two methods in Jiang et al. (2011) and Zheng et al. (2020). As shown in Table 3, several cross validations were conducted on the dataset: 2-fold, 3-fold, 4-fold, and 5-fold. Every cross validation was repeated for 10 times and the final recognition accuracy was the average of the sum. From the results, it was observed that the proposed classification method outperforms the classification method in Jiang et al. (2011). Since the accuracy of both methods were high and similar, then it was compared to the decrease of error rate (ER) for a proposed method versus the method in Jiang et al. (2011), which is shown in the fourth row. The proposed method leads the error rate decreasing nearly 30%-40%.

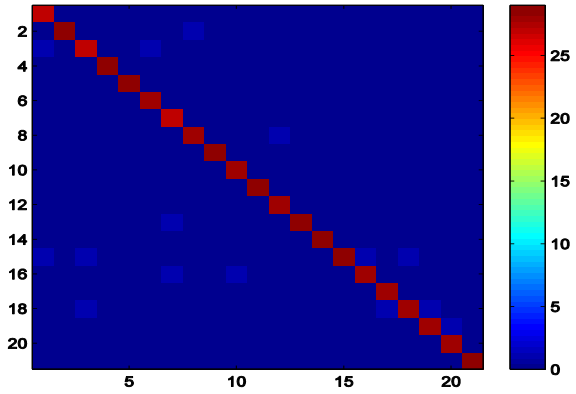

Figure 2: Confusion matrix using trajectory and HOG feature.

In addition, another experiment was performed for verifying whether the proposed system can handle signer independence problems. Eight repetitions were made. Each time the experiments were performed by treating one person's samples as testing data and the remaining as training data. The results are summarized in Table 4. Parameters η and λ were set as $\sqrt{\eta}$ = 0.2 and $\sqrt{\lambda}$ = 0.3. It was observed that an average recognition accuracy of 95.34% was obtained, which indicated the proposed system is suitable for signer independent recognition tasks.

Table 3: Recognition results for different classification methods

Classification methods	2-Fold	3-Fold	4-Fold	5-Fold
Method in Jiang et al. (2011) ($\sqrt{\eta}=0.2,\sqrt{\lambda}=0.5$)	95.63%	95.99%	96.29%	96.47%
Method in Zheng et al. (2020) ($\sqrt{\eta}=0.2,\sqrt{\lambda}=0.5$)	96.08%	96.89%	96.99%	97.21%
Proposed method($\sqrt{\eta}=0.2,\sqrt{\lambda}=0.3$)	**98.01**%	**98.10**%	**98.27**%	**98.60**%
ER decrease	31.58%	31.13%	30.72%	35.98%

Table 4: Recognition results on different signers

Testing on	Number of test signs	Accuracy	Average accuracy
Person 1	168	96.07%	95.34%
Person 2	126	97.62%	
Person 3	168	96.43%	
Person 4	168	96.64%	
Person 5	126	91.27%	
Person 6	147	96.56%	
Person 7	168	95.24%	
Person 8	126	92.86%	

In table 4, another experiment was performed for verifying whether the proposed system can handle signer independent problems. Eight repetitions were made. Each time it was treated for one person's samples as testing data and the remaining as training data. The results are summarized in Table 4. A real time recognition application using entire samples as training data was created. An example for the recognition of the sign "Airplane" is shown in Figure 3.

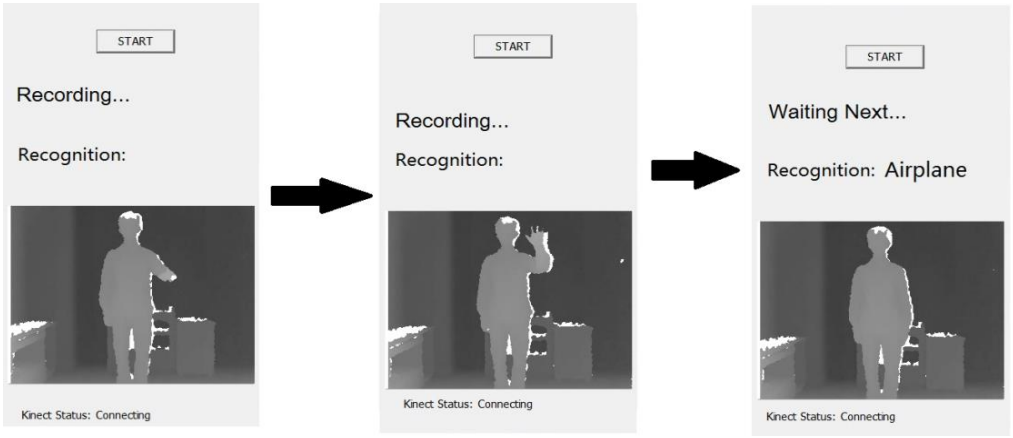

Figure 3: Recognition example of sign "Airplane".

Conclusion

The objective of this study was to develop a sign language recognition system based on color stream and skeleton points. The sparse coding technique was applied to the sign language recognition system based on techniques that use hand trajectories and hand shapes in combating sign recognition challenges. The Kinect sensor was used to acquire data: color stream and skeleton stream, through Kinect SDK. Then the sign feature vector composed by normalized hand trajectories and HOG feature was extracted, which represented a hand shape. A sparse dictionary learning algorithm, LC-KSVD, was applied to obtain a discriminative dictionary. Based on that, it was further used to develop a new classification approach which was optimal for computing the sparse code of an unknown input sign and came up with better results. Several experiments were conducted to evaluate the proposed system. A database consisting of 21 sign words was collected. Experimental results showed the proposed system gets high recognition accuracy, and is suitable for signer independent recognition task. The following suggestions are open research directions discussed by this work. First, the proposed technique does not rely on large number of sign words. It would naturally be desirable to focus on increasing the number of sign words in the database. Furthermore, the large database of the sign words can further be investigated and improved, or even extended to continuous sign language recognition.

Acknowledgment

I am thankful to my employer; University of Dar es Salaam for all the support and encouragement. The author also thanks the Chief Editor (Prof. John Mahugija) for his valuable comments and suggestions which helped to improve the quality of the article.

Conflict of Interest: I declare that there is no conflict of interest regarding this work.

References

Angelopoulou A, Garcia-Rodriguez J, Orts-Escolano S, Kapetanios E, Liang X, Woll B and Psarrou A 2019 Evaluation of different chrominance models in the detection and reconstruction of faces and hands using the growing neural gas network. *Pattern Anal. Appl.* 22(4): 1667-1685.

Azis NA, Choi HJ and Iraqi Y 2015 Substitutive skeleton fusion for human action recognition. *Int. Conf. Big Data Smart Comput*. (p. 170-177). IEEE.

Camgoz NC, Hadfield S, Koller O, Ney H and Bowden R 2018 Neural sign language translation. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 7784-7793, Salt Lake City.

Chai X, Li G, Lin Y, Xu Z, Tang Y, Chen X and Zhou M 2013 Sign language recognition and translation with kinect. *Proc. IEEE Conf. Automatic Face and Gesture Recognit.* 655: 4.

Cheok MJ, Omar Z and Jaward MH 2019 A review of hand gestures and sign language recognition techniques. *Int. J. Mach. Learn. Cybern.* 10(1): 131-153.

Donahue J, Hendricks LA, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K and Darrell T 2015 Long-term recurrent convolutional networks for visual recognition and description. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* p. 2625-2634, Boston.

Fatmi R, Rashad S, Integlia R and Hutchison G 2017 American Sign Language recognition using hidden Markov models and wearable motion sensors. *Trans. MLDM* 10: 41-55.

Gao W, Ma J, Wu J and Wang C 2000 Sign language recognition based on HMM/ANN/DP. *Int. Pattern Recognit. Artif. Intell.* 14(5): 587-602.

Guha T and Ward RK 2012 Learning sparse representations for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (8): 1576-1588.

Huang J, Zhou W, Zhang Q, Li H and Li W 2018 Video-based sign language recognition without temporal segmentation. In *Proc. AAAI Conf. Artif. Intell.* 32(1).

Jiang Z, Lin Z and Davis LS 2011 Learning a discriminative dictionary for sparse coding via label consistent K-SVD. *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR).* p. 1697-1704.

Jing L, Vahdani E, Huenerfauth M and Tian Y 2019 Recognizing American sign language manual signs from RGB-D videos. *Comput. Vis. Pattern Recognit.* 1-16.

Kemeng L, Shao F, Jiang G and Yu M 2015 Joint structure–texture sparse coding for quality prediction of stereoscopic images. *Electron. Lett.* 51(24): 1994-1995.

Liang RH and Ouhyoung M 1998 A real-time continuous gesture recognition system for sign language. *Proc. 3rd IEEE Int. Conf. Autom. Face Gesture Recognit.* p. 558-567, Nara.

Liu T, Zhou W and Li H 2016 Sign language recognition with long short-term memory. *Proc. IEEE Int. Conf. Image Process.* p. 2871-2875.

Mahmoud M, Baltrušaitis T and Robinson P 2016 Automatic analysis of naturalistic hand-over-face gestures. *ACM Trans. Interact. Intell. Syst. (TiiS)* 6(2): 1-18.

Raheja JL, Mishra A and Chaudhary A 2016 Indian sign language recognition using SVM. *J. Pattern Recognit. Image Anal.* 2(26): 434-441.

Reyes M, Dominguez G and Escalera S 2011 Feature weighting in dynamic time warping for gesture recognition in depth data. *IEEE Int. Conf. Comput. Vision Workshops, ICCV 2011 Workshops*. p. 1182-1188.

Patsadu O, Nukoolkit C and Watanapa B 2012 Human gesture recognition using kinect camera. *Proc. 9th Int. Conf. Comput. Sci. Softw. Eng.*, p. 28-32, IEEE, Bangkok.

Pöhlmann STL, Harkness EF, Taylor CJ and Astley SM 2016 Evaluation of kinect 3D sensor for healthcare imaging. *J. Med. Biol. Eng.* 36: 857-870.

Sharma A, Mittal A, Singh S and Awatramani V 2020 Hand gesture recognition using image processng and feature extraction techniques. *Procedia Comput. Sci.* 173: 181-190.

Yang HC and Wang XA 2015 Cascade face detection based on histograms of oriented gradients and support vector machine. *10th Int. Conf. P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)* (pp. 766-770). IEEE.

Zhang Q and Li B 2010 Discriminative K-SVD for dictionary learning in face recognition. *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* (p. 2691-2698), IEEE.

Zhang D, Zhang Y, Hu X, Zheng G, Tang J and Feng C 2016 Fast OMP algorithm for 3D parameters super-resolution estimation in bistatic MIMO radar. *Electron. Lett.* 52(13): 1164-1166.

Zheng S, Zhang Y, Liu W, Zou Y and Zhang X 2020 A dictionary learning algorithm based on dictionary reconstruction and its application in face recognition. *Math. Probl. Eng.* 2020, Article ID 8964321,10 pages.