This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

Motionlets Matching with Adaptive Kernels for 3D Indian Sign Language Recognition

P.V.V. Kishore, *Senior Member, IEEE,* D.Anil Kumar, *Student, IEEE,* A.S.C.S.Sastry, *Member, IEEE,* and E.Kiran Kumar, *Student, IEEE*

***Abstract*—Recognizing human gestures in sign language is a complex and challenging task. Human sign language gestures are a combination of independent hand and finger articulations, which are sometimes performed in coordination with the head, face, and body. 3D motion capture of sign language involves recording 3D sign videos that are often affected by interobject or self occlusions, lighting, and background. This paper proposes characterization of sign language gestures articulated at different body parts as 3D motionlets, which describe the signs with a subset of joint motions. A two-phase fast algorithm identifies 3D query signs from an adaptively ranked database of 3D sign language. Phase-I process clusters all human joints into motion joints (MJ) and nonmotion joints (NMJ). The relation between MJ and NMJ is analyzed to categorically segment the database into four motionlet classes. Phase-II process investigates the relation within the motion joints to represent shape information of a sign as 3D motionlets. The 4-class sign database features 3 adaptive motionlet kernels. A simple kernel matching algorithm is used to rank the database according to the highest-ranked query sign. The proposed method is sign invariant to temporal misalignment and can characterize sign language based on a 3D spatiotemporal framework. In this study, five 500-word Indian sign language datasets were used to evaluate the proposed model. The experimental results reveal that the method proposed here improved recognition compared with state-of-the-art 3D action recognition methods.**

***Index Terms*—3D motion capture, 3D sign language, adaptive kernel matching, Motionlets matching, pattern classification.**

I. Introduction

Sign language recognition (SLR) is a subset of human action recognition, and has limitations related to hand tracking, finger shape modelling, torso tracking, head movements, and pattern recognition. The complexity of the problem lies in building a real-time computer application to mimic a human translator. There are two major problems in SLR machine translation: one is related to input dimension and the other is developing a recognizer that is immune to human motion dynamics.

Sign language is a visually articulated form of communication developed for hearing-impaired people. Recognizing these articulated motions, which are highly correlated in motion space, is a very difficult task. Researchers have used 1D/2D/3D sensors to capture and process sign language data. However, 1D and 2D data models are popular due to the availability of cheap sensors. 1D data captures only the finger movements, which is converted into 1D time-varying signals. An RGB color video camera can be used for representing visually articulated sign language. However, this 2D data suffers from considerable noise, such as lighting variations, background changes, camera movements, and interhand occlusions. These are the major limitations of 2D-video-based SLR presented in the literature [1].

Recent introduction of 3D depth cameras has enabled effortless segmentation of human objects in a cluttered background. Moreover, these cameras are a cost-effective tool that captures 3D motion of human joint positions and skeleton. A 3D skeleton model has no face and fingers, making SLR extremely difficult. Therefore, researchers have proposed a sensor data fusion model by using 3D depth and RGB data to capture sign language [2][3]. However, effectively combining the sensor data is a nontrivial problem in machined sign language translation. The major problem is mapping the depth information and color information on a similar scale. For example, identifying overlapping of hands in signs such as "together" and "with" is extremely difficult using this model. The occluded hand information is completely lost during capture. Moreover, sign language actions are dynamic and nonlinear temporal structures of specific length. For example, the action for "breakfast" comprises two different hand movements, such as moving hand from left to right or right to left across the face and putting food into mouth. Determining the temporal relation between these two action sequences is necessary to recognize a sign correctly. Finally, the double hand signs limit the recognition by creating data aliasing on overlapped RGB and depth map data. Moreover, no two human movements can be quantified into one movement based on the speed of the movements. Therefore, modeling these variations in sign language poses challenges to machine translation.

Studies on 3D human motion recognition have used a skeleton from either a 3D motion capture system or Kinect with 20–24 human joints. However, these are of little use for sign language recognition as most of the signs are based on finger movements. Modeling of human fingers is a complicated task due to their small size and limited motion compared with the entire body. One objective of this study is to design a motion capture signer model for a 3D motion capture system.

This paper proposes novel motionlets to represent sign language by using 3D-motion-captured sign data. First, we propose a 3D finger model for errorless capture of 3D sign language. Second, motion segmentation divides the human joint set into motion joints (MJ) and nonmotion joints (NMJ) in a sign video sequence. The recognition occurs in two phases.

This work was supported by the Department of Science and Technology, SEED Devision, New Delhi, India. (Grant No: SEED/TIDE/013/2014(G)).

The authors are with the Department of Electronics and Communication Engineering, Biomechanics and Vision Computing Research Center, Koneru Lakshmaiah Education Foundation (Deemed–to–be–University), Guntur 522502, India, e-mail: (pvvkishore@kluniversity.in).

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

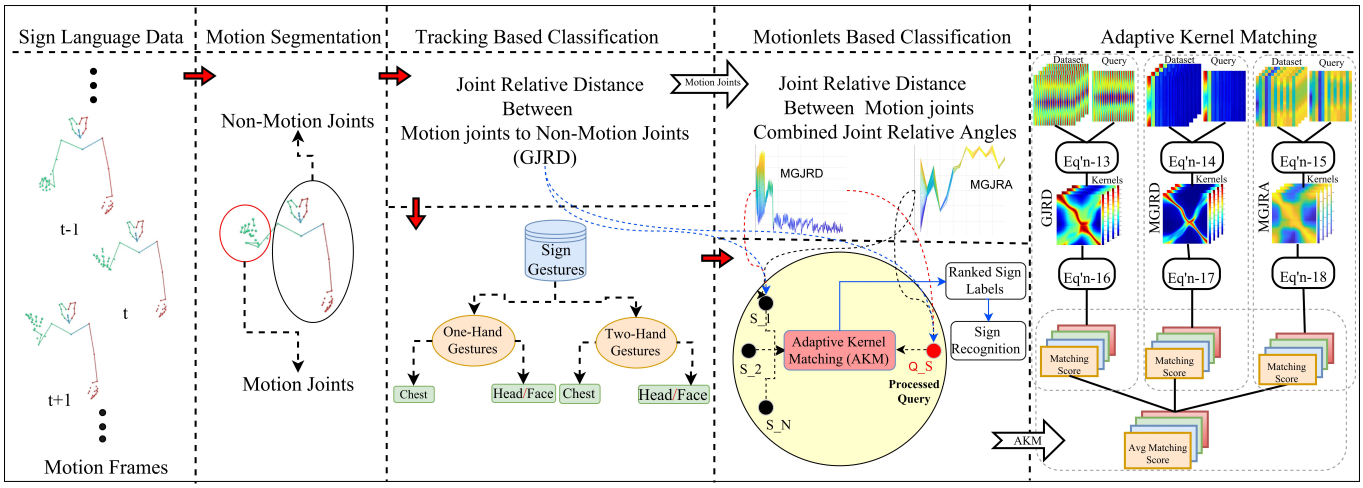

Fig. 1: Framework of the proposed 3D sign language recognizer.

Phase-I comprises motion segmentation and 4-class database creation. Motion segmentation separates motion joints from nonmotion joints. The relation between MJ and NMJ results in the first set of tracking features, which initialize phase-I classification. The 4-class database is labelled as Single-hand_Face-Head, Two-hand_Face-Head, Single-hand_Chest, and Two-hand_Chest. This allows faster searches through the database at the later stages of the classification. Phase-II extracts finger shapes from MJ by applying joint relative distance (JRD) and joint relative angles (JRA).

Fig. 1 illustrates the framework of the proposed 3D sign language classifier in two phases. Although an articulated sign language involves the entire upper body joints for meaningful communication, a specific sign may involve a small subset of joints. Three adaptive kernels are defined based on the tracking, shape, and orientation of motion joints or motionlets. Multiple adaptive kernel matching recognizes the query sign from the ranked database. This new model uses multiple 3D features to distinguish between scuttle variations in fingers for accurate sign recognition.

The proposed method models finer details that can detect query sign accurately with 75% lesser search space than the visual methods. The proposed framework was tested on five sets of 500 sign datasets of Indian sign language. The five sets were captured using Vicon 9-camera motion capture technology with five different signers; one is an expert sign language interpreter and the other four are sign language learners. The learner set has numerous subtle variations in hand poses, hand speeds, and the place of articulation randomly introduced in signs to check the robustness of the proposed framework. This indicates that the proposed model is insensitive to signer movements and view invariants. The framework was then compared with 3D human motion retrieval models with sign language data [4][5] and other 3D sign language sensor-based approaches from the literature [6][7][8][9].

The rest of the paper is organized as follows. Section 2 presents the related works in brief; Section 3 presents the joint motion segmentation model; Sections 4 and 5 present the phase-I classification based on tracking and phase-II recognition based on motionlets, respectively. Section 6 presents an evaluation of the proposed model by using an Indian sign language dataset.

II. Related Literature

Sign language is a visual language designed with human motions to create a medium of communication for hearing-impaired people. Human motions are spatiotemporal patterns in 3D space. The frameworks for 3D SLR can be decoded into two parts: features representing spatiotemporal data and recognizing models for learning dynamic patterns in the data.

In machine translation in SLR, the features depend on the acquired data. In the last three decades, researchers have used 1D/2D/3D data of sign language to represent various characterizations of these datasets. 1D data is captured with cyberlove [10] in which transition-movement models are used as 1D signals to effectively handle transition between two signs. The average accuracy of these devices reported in the literature is 88%. These are commercially available, and researchers worldwide are exploring possibilities of building SLR systems [8][9][11]. Recently, these methods [12][13] have used data fusion models that combine leap motion data with Kinect depth or RGB video data to improve the recognition rates in SLR. However, line-of-sight and hand orientation problems with leap motion devices heavy restrict nontechnical signers.

Another low-cost depth sensor that has been popular with sign language researchers since 2011 is Kinect [14]. The associated SDK libraries enables researchers to access human skeletal joint data and RGB-depth data for recognition and classification. A large online dataset is available for 3D action recognition using Kinect data under different environments [15]. Action recognition tasks focused on human joint skeletons can be used as action recognition datasets for pose estimation, human–computer interaction, human tracking, human activity recognition, and human–object interaction applications [16][17][18]. Prior to Kinect and the leap motion sensor, the Time-of-Flight (ToF) [19] was a low-cost depth image sensor used for SLR. In comparison, ToF produced highly accurate 3D hand gestures of robust sizes, orientations, and

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

backgrounds for recognition. ToF cameras were used for recognizing gestures in sign language with considerable accuracy [19], [20]. However, ToF cameras were slow in capturing 3D data with a relatively small capture area compared to the object size. Therefore, it difficult to capture the full body of the signer in one attempt using leap-motion or ToF cameras. Kinect depth features should be effectively combined with RGB and skeletal data for creating an optimized feature, representing human motions [21]. These sensors offer improved 3D motion datasets with efficient algorithms for recognition. Moreover, all these sensors have their advantages in terms of low cost and drawbacks related to motion data.

The most accurate computerized 3D action generator is the 3D motion capture system (3D mocap). Moreover, the capture process in this system is rather complex and tedious with post-processing requirements [22]. This is a multicamera system that projects infrared light to precisely map the 3D position of each joint in 3D space. We used a 9-camera model with 8 motion-capture cameras and one video camera for capturing 3D sign motions. Fig. 2 shows the camera setup used to capture the 3D Indian sign language action dataset.

Fig. 2: 9-Camera Mocap system for Indian sign language capture

The datasets are a combination of joint positions in 3D space and are robust to interhand occlusions, overlapping, lighting, background, and object motion blur. The joint angles of each joint can be computed from the captured joint positions. Fifteen different features for a motion can be extracted from the system. The position vectors in 3D space are analysed in different models. In one of the simplest methods [23], cosine-based similarity is calculated using the dot product between two position vectors of the same motion by using an Optitrack 3D motion capture system. The proposed model uses 12 joints based on which different human motions such as walking, running, jumping and sitting are analysed.

A probabilistic framework using local mixture of gaussians (LGMM) defines the local gaussian process on joint position data [24]. Because of local processing, the speed and accuracy of action recognition can be enhanced for large 3D action datasets. Similarly, the Hidden Markov Model (HMM) can be used to represent position variations in 3D state space for action recognition [25]. Estimation of 3D joints for SLR using probabilistic models results in noisy classifications due to very small variations in the actions of fingers. Determining the accurate state with a small set of features is a difficult task that results in false predictions and thus an inefficient SLR system.

The biggest drawback of this approach is establishing synchronization between the dataset action frames and query frame sequences. This problem can be solved using dynamic time warping (DWT) on 3D position data in 3D mocap video sequences [26]. However, DWT suffers from large spatiotemporal changes between query sign and dataset of different signers. The DWT model was extended to tensor DWT (TDWT), where the local joint-to-joint similarity is measured between 3D skelatal joints [27]. Additional mapping from tensor shape descriptor (TSD) to local subspace limits the computation speed of the algorithm.

Artificial intelligence models such as neural networks [28], deep neural networks, and convolutional neural networks (CNN) [29] have made 2D action recognition a real-time application. However, these models suffer from limited reliability in accepting time series data and large training sample size.

Graph matching has accurately transformed the 3D action recognition [30]. The position vectors of the 3D model are a natural choice to fit the graph model. Matching models can be constructed in the form of kernels on these graphs from 3D action data videos. Recent works show their superior performance in 3D action recognition on skeletal and mocap datasets [31]. The limitations are missing nodes due to inefficient data capture and inaccurate feature models to represent the nodes.

Thus, our study has the following advantages: (1) eliminating spatiotemporal misalignments between query and datasets by removing frames from the matching process, (2) discovery of small action variations using motionlet adaptive kernels, and (3) database structure for large vocabulary 3D sign language.

III. JOINT MOTION SEGMENTATION

Sign language involves human body movements that are either independent or dependent on some other part of the body. The sign actions may be single-handed, double-handed, or both with respect to static chest or head regions. Hence, sign representations have motion and nonmotion human parts, with a high degree of correlation between the two parts. The literature shows 20–25 marker templates or joints used for human motion representation and analysis. Fingers and face are not part of these models. For SLR, fingers and face skeletal representation is a major component. Fig. 3 shows the human joint skeletal designed for capturing 3D signs. A 57-marker template is designed with precision placed marker positions to capture full potential of Indian sign language. The sign template shows 16 face–head joints, 1 chest, 20 left hand and right hand finger joints. The joint motion segmentation (JMS) module is a part of phase-I classification, whose objective is to differentiate motion joints from nonmotion joints in a sign. In this study, motion joints are defined as hand and finger joints that move in the entire video of a sign against the nonmoving joints. To achieve JMS, we apply the gradient distance between similar joints in consecutive frames and extract the moving joint indexes in a sign using mean motion threshold (MMT).

Each joint J is represented with a 3D coordinate joint location $l_J(t) = [x_J(t), y_J(t), z_J(t)] \in \mathrm{R}$ on each frame

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

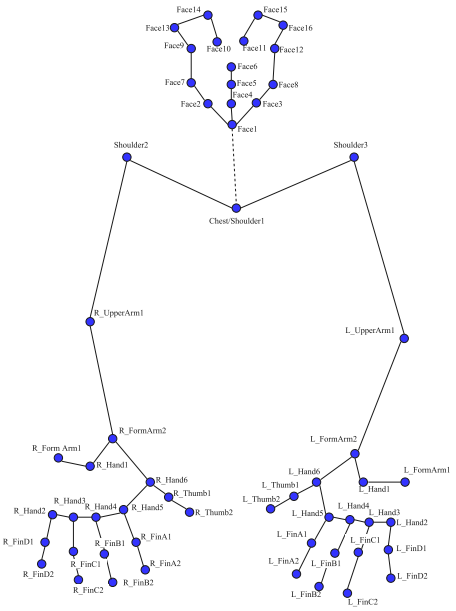

Fig. 3: Signer representation in 3D motion capture

t. In most of the previous works, the joint positions are normalized with respect to the mean positions or variance. However, the present study uses kernel learning modules in the recognition phase, which is built based on the difference in joint locations rather than the joint positions themselves. The distance gradient is computed between two consecutive 3D frames from a joint location $l\left(J_i^t\right)$ to $l\left(J_i^{t+1}\right)$ with Euclidian distance function, where i is the joint index. Euclidean distance function J_D^t of i^{th} joint in consecutive frames t and $t+1$ is defined as

$$J_D^t = \left\| l\left(J_i^t\right) - l\left(J_i^{t+1}\right) \right\|_2^2 \ \forall\ i = 1\ to\ N_J,\ t = 1\ to\ N_t \quad (1)$$

where N_J and N_t are the number of joints and frames in a 3D video, respectively. $J_D^t \in \mathrm{R}^{N_J \times (N_t - 1)}$ is a real matrix with values representing positional difference between joints in all frames of a 3D video. To separate the motion joints from nonmotion joints, we threshold the matrix in (1) using MMT, computed for i^{th} joint in two consecutive frames:

$$J_D^{tr} = \frac{1}{2} \sum_{t=1}^{2} J_D^t \ \ \forall\ i^{th}\ \text{joint} \quad (2)$$

The joint motion segmentation yields joints that have a relatively high motion content in a 3D video sequence. The action joint set is denoted by B_t for t^{th} frame as

$$B_t = \left\{ J_D^t \geq J_D^{tr} \right\} \quad (3)$$

where B_t is a binary vector indicating the separation between motion joints and nonmotion joints for a sign. The motion joint set M_j^t and nonmotion joint $\hat{M}_k^t$ in the entire video sequence with N_t frames can expressed as

$$\begin{aligned} \left\{ M_j^t \right\} &= \underset{j}{ind} \left\{ B_t == 1 \right\} \\ \left\{ \hat{M}_k^t \right\} &= \underset{k}{ind} \left\{ B_t == 0 \right\} \end{aligned} \quad (4)$$

The number of joints in a MJ set $\left\{ M_j^t \right\}$ can change according to the hand movements in the sign. For example, an NMJ can transform into a MJ from the 50^{th} frame and vice versa. Therefore, it is indispensable to validate the MJ and NMJ sets for $N_t - 1$ frames in the video sequence. The necessary condition for creating disjoint MJ and NMJ sets can be given as $\underset{\forall\ N_t}{\bigcap} M_j^s\, \hat{M}_k^s = \{\phi\}$.

The MJ set for an entire video sequence is computed on a frame-by-frame basis as

$$\left\{ M_j^s \right\} = \begin{cases} \underset{j}{ind} \left\{ M_j \right\} & if \bigcap_{i=1}^{N_t - 1} \left\{ M_j^i \right\} = \{\phi\} \\ \underset{j}{ind} \left\{ \bigcup_{i \in N_t} M_j^i \right\} & if \bigcap_{i=1}^{N_t - 1} \left\{ M_j^i \right\} \neq \{\phi\} \end{cases} \quad (5)$$

Finally, the NMJ set can be expressed as

$$\left\{ \widehat{M_k^s} \right\} = \underset{k}{ind} \left\{ \widehat{M_k} \right\} \qquad if \ \underset{i=1}{\overset{N_t - 1}{\bigcap}} \left\{ \widehat{M_k^i} \right\} = \{\phi\} \quad (6)$$

The MJ and NMJ set for as entire video sequence with N_t frames is $M^{N_t} = [M^1, M^2, ..., M^{N_t}]$ and $\hat{M}^{N_t} = [\hat{M}^1, \hat{M}^2, ..., \hat{M}^{N_t}]$. At this stage, it is difficult to classify the signs based on the joint distance data. Identifying a sign involves tracking hands with respect to other parts of the body, tracing hand trajectories, and finger shapes. The simplest and most effective method for calculating these components from 3D motion capture data is by using JRD [30]. The other problem is managing the database efficiently for faster searches by using the large sign language database. We first handle the database model and then extract components to model signs.

IV. TRACKING-BASED CLASSIFICATION—DATABASE SETUP:PHASE-I

The sign language is captivated by single and/or double hand movements with respect to the head or chest. Majority of the signs in a sign language bundle the head–face–chest joints as an NMJ set. The first step is to preclassify the 3D sign database based on the MJ set. The MJ set can have either single hand joints or both hand joints depending on the class of the sign. For a single hand sign, the joints on the right hand belong to the set $\left\{ M_j^s \right\}$ while the remaining joints on the left hand, face, head and chest belong to the set $\left\{ \hat{M}_k^s \right\}$. Similarly, with two hand moving signs, the MJ set will have left and right hand joints, and the NMJ set comprises the remaining joints. Hence, classification of the dataset is accomplished in two phases. In the first phase, the database is classified by tracking hands and tracking the finger shapes and orientations in the second phase.

To calculate the hand motion with respect to the face–head or chest, we apply JRD between each joint in the MJ set $\left\{ M_j^s \right\}$ and all joints in the NMJ set $\left\{ \hat{M}_k^s \right\}$. JRD yields a matrix of distance values of size $L \times K$, where L and K are joints in the MJ and NMJ sets. This JRD is designated as gross JRD (GJRD). To calculate GJRD, $\left\{ M_j^t \right\}$ and $\left\{ \hat{M}_k^t \right\}$ are featured

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

with the corresponding joint position vectors $l^t_{M_j}$ and $l^t_{\widehat{M_k}}$ in each frame t. GJRD is calculated as

$$GJRD^t_S = \left\| l^t_{M_j}(x,y,z) - l^t_{\widehat{M_k}}(x,y,z) \right\|^2_2 \quad \forall\ j = 1\ to\ \mathrm{L}, k = 1\ \mathrm{to\ K},\ t = 1\ to\ N_t \tag{7}$$

The matrix $GJRD^t_S \in \mathrm{R}^{L\times K}$ gives the movement of each MJ with respect to NMJ. This is important in sign language classification to determine whether the moving hand is pointing towards the head or the chest or to a specific location on the face such as nose or lips or eyes. For example, the signs "MAN" and "WOMAN" are derived with respect to face attributes such as moustache and nose, which are otherwise not detectable as the hand shape is constant throughout the sign. Similarly, signs such as "HAPPY" and "MY" are based on the chest . Hence, the database labels are pre-classified into four classes labelled as *Single-hand_Face-Head, Single-hand_Chest, Two-hand_Face-Head*, *and Two-hand_Chest* in Phase-I classification. The inclusive advantage of this framework is the faster sign detection than similar models [30] [31]. Classification into the four labels requires a decision on the GJRD matrices of individual signs calculated as

$$L\left(Single - hand\_Face - Head\right) = \arg\min_{sh\_fh(J)}\left(GJRD_S\right) \tag{8}$$

$$L\left(Single - hand\_Chest\right) = \arg\min_{sh\_c(J)}\left(GJRD_S\right) \tag{9}$$

$$L\left(Two - hand\_Face - Head\right) = \arg\min_{th\_fh(J)}\left(GJRD_S\right) \tag{10}$$

$$L\left(Two - hand\_Chest\right) = \arg\min_{th\_c(J)}\left(GJRD_S\right) \tag{11}$$

The $\min_{sh\_fh(J)}$ is an argument to determine the minimum distance between the single hand and head_face regions; this is extended for the other three labels, and is a form of minimum distance classifier. The entire dataset is classified and attached to the corresponding labels. The GJRD values provide an exact location on the face where the hand is pointing. Each dataset label contains a set of signs with GJRD values related to a sign. However, identifying a sign is categorically impossible without knowing the finger shapes. This requires a local distribution of finger joints with respect to the adjacent fingers for representing the hand shape as motionlets. Motionlets are small intrafinger motion variations and orientations per frame that are analyzed collectively for shape identification and correct classification. Two regular problems in video matching or classification are (1) different number of frames in the query and database, and (2) frame location in both query and database video for matching. By using kernel matching framework based on graph theory techniques, the SLR model can be made independent of the frame count and frame location in the sign video sequence. Phase-II of the SLR matching provides a theoretical solution to 3D SLR recognition.

V. MOTIONLETS-BASED CLASSIFICATION—PHASE-II

We define motionlets as intrafinger variations, which are relatively tiny motions compared with hand movements. A finger shape is a combination of small variations between adjacent finger joints in each frame formed with individual finger joint motionlets. The trajectories of each finger are mapped in the previously calculated GJRD between static NMJ and moving MJ. We measure the motionlet movements in each 3D frame with MGJRD computed on the MJ in each frame. However, the exact finger shape is difficult to model based on only the distance between MJ due to interfinger overlaps in some signs. To compensate for this, we add the Gross Joint Relative Angle (MGJRA) between fingers. MGJRA gives the angles between three pairs of joints at which the angle is measured as the projection vector. The MGJRDs between MJs $\left\{M^s_j\right\}$ measured using (7) gives a $\left(\frac{L(L-1)}{2}\right) \times N_t$ matrix of real distances, where L is the number of MJs on N_t sign frames. MGJRAs between MJs $\left\{M^s_j\right\}$ are calculated by choosing three joints with two projection vectors, where one joint is common to both the vectors. For a 3-joint marker set $(J_1, J_2, J_3) \subset J_i,\ \forall\ \ i = 1\ to\ L$, where $J_j = l\left(x_j, y_j, z_j\right)$ is a 3D location, which forms two projection vectors $\overrightarrow{P_{12}} = d\left(J_1, J_2\right) \in \mathrm{R}^3$ and $\overrightarrow{P_{23}} = d\left(J_2 J_3\right) \in \mathrm{R}^3$ measuring an angle at joint J_2 as

$$\theta_{J_2} = \cos^{-1}\frac{\overrightarrow{P_{12}}\overrightarrow{P_{23}}}{\sqrt{\overrightarrow{P_{12}}}\sqrt{\overrightarrow{P_{23}}}} \tag{12}$$

MGJRA is a matrix of angular values of each MJ in the sign. The size of the MGJRA matrix is $L \times N_t$. The distances on each motionlet are represented with MGJRD, and the angles with MGJRA. However, a combination of Phase-I and Phase-II procedures is necessary for accurately identifying a queried sign in the database. The procedure adopted in this study for recognition is based on adaptive kernels. We used three features to identify a sign: GJRD from Phase-I, and MGJRD and MGJRA from Phase-II.

Adaptive kernels match a query 3D sign with the database signs to translate input 3D video into ranked text labels. Rank in the output shows other possible near matches to the query video. Adaptive kernels are computed for the three attributes that represent a 3D sign. The database of 3D signs is preclassified into a 4-structured label matrix with 2 cells per structure per sign. One cell occupies data from Phase-I and the other from Phase-II. We calculate three kernels per sign in each structure to represent a sign in the sign language database. This yields a query sign $Q(g_1, g_2, a_2)$and the prelabelled dataset sign $D^c_i(g_1^{'}, g_2^{'}, a_2^{'})$, where c represents class index and i represents the sign index; the objective of the adaptive kernel classifier is to obtain a matched label for the query; g_1, g_2 and a_2 are matrices of GJRD, MGJRD, and MGJRA for each sign. We define $k_1\left(g_1, g_1^{'}\right)$, $k_2\left(g_2, g_2^{'}\right)$, and $k_a\left(a_1, a_2^{'}\right)$ as the adaptive kernels representing GJRD, MGJRD and MGJRA data for two motion signs from the query and database, respectively; here, g_1 and $g_1^{'}$ characterize GJRD values from Phase-I in the query and database sign. Similarly, $\left(g_2, g_2^{'}\right)$

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

and $\left(a_2, a_2^{'}\right)$ characterize MGJRD and MGJRA motionlet distances and angles, respectively. $k_1\left(g_1, g_1^{'}\right)$, $k_2\left(g_2, g_2^{'}\right)$, and $k_a\left(a_1, a_2^{'}\right)$ measure the similarity between attributes defining a 3D query and database sign. Using the 3D joint descriptor in Phase-I as the trajectory descriptor attribute of a sign, the trajectory kernel can be defined as

$$k_1\left(g_1, g_1^{'}\right) = \begin{cases} \exp\left(-\frac{\left\|GJRD_S^D - GJRD_S^{'}\right\|_2^2}{2\sigma_1^2}\right) & if \left\|GJRD_S^D - GJRD_S^{'}\right\|_2 < \tau_1 \\ 0 & otherwise \end{cases} \tag{13}$$

where $GJRD_S^D$ and $GJRD_S^{'}$ are the sign joint trajectory descriptors for database sign and query sign representing the motion between MJ and NMJ, respectively. $\sigma_1 > 0$ is the gaussian function scale parameter and τ_1 is the positive threshold. The Phase-II kernels are motionlet shape kernels of MJ only, defined using MGJRD and MGJRA matrices as

$$k_2\left(g_2, g_2^{'}\right) = \begin{cases} \exp\left(-\frac{\left\|MGJRD_S^D - MGJRD_S^{'}\right\|_2^2}{2\sigma_2^2}\right) & if \left\|MGJRD_S^D - MGJRD_S^{'}\right\|_2 < \tau_2 \\ 0 & otherwise \end{cases} \tag{14}$$

where $MGJRD_S^D$ and $MGJRD_S^{'}$ are joint relative distances of MJ in query sign and dataset sign video under inspection, and are finger-shaped attributes. $\sigma_2 > 0$ is a gaussian scale parameter and $\tau_2 > 0$ is the threshold. The orientation kernel is given by

$$k_a\left(a_1, a_1^{'}\right) = \begin{cases} \exp\left(-\frac{\left\|MGJRA_S^D - MGJRA_S^{'}\right\|_2^2}{2\sigma_3^2}\right) & if \left\|MGJRA_S^D - MGJRA_S^{'}\right\|_2 < \theta_t \\ 0 & otherwise \end{cases} \tag{15}$$

where $MGJRA_S^D$ and $MGJRA_S^{'}$ are joint relative angles of MJ in the query sign and the dataset sign video under inspection, which are finger-shaped attributes. $\sigma_3 > 0$ is a gaussian scale parameter and $\theta_t > 0$ is the threshold.

The three kernels $k_1\left(g_1, g_1^{'}\right)$, $k_2\left(g_2, g_2^{'}\right)$, and $k_a\left(a_1, a_2^{'}\right)$ for a sign form a matrix of values having sizes $N_D \times N_Q$, where N_D and N_Q are number of frames in dataset sign and query sign. The proposed algorithm manifests the kernel as a matrix whose rows and columns are database frames and query frames, respectively. The intersection between query and database frames gives the matching score. The maximum is the score, the stronger the match. A maximum score point on the matrix indicates to a set of query and database frames that are exact matches to each other. This algorithm has two advantages: (i) it is independent of the number of frames and (ii) the sign location in the video sequence. These advantages result from the multiframe kernel matching between all frames of dataset to all frames in query. For example, the 20^{th} frame in query sign can match with the 15^{th} sign in the dataset, and only the maximum kernel value at this intersection is needed to calculate the decision boundary of the classifier. The decision boundary for the r^{th} row in the kernel matrix can be given by

$$p_r^{s\_trj} = \frac{1}{N_Q} \sum_{b \in N_Q} \arg\max_{r \in N_D} \left(k_1^r\left(g_1, g_1^{'}\right)\right) \tag{16}$$

$$p_r^{s\_shp} = \frac{1}{N_Q} \sum_{b \in N_Q} \arg\max_{r \in N_D} \left(k_2^r\left(g_2, g_2^{'}\right)\right) \tag{17}$$

$$p_r^{s\_ang} = \frac{1}{N_Q} \sum_{b \in N_Q} \arg\max_{r \in N_D} \left(k_a^r\left(a_1, a_1^{'}\right)\right) \tag{18}$$

where $p_r^{s\_trj}$ represents the trajectory patterns between signs. $p_r^{s\_shp}$ and $p_r^{s\_ang}$ are patterns matching scores for MJ distances and angles indicating the shape of the motion parts of the sign. $p_r^{s\_trj}$, $p_r^{s\_shp}$ and $p_r^{s\_ang}$ belong to the range [0,1]. The value of zero or near zero indicates no matching and one indicates a perfect match. Instead of the three values, an average of three parameters is considered as a measure of similarity between the query and database sign. The query sign kernels interact with all the database sign kernels and a similarity measure is returned. The similarity vector gives matching scores between query sign and database signs. This vector is ranked to separate out the maximum ranked sign from the database. The 2^{nd} and 3^{rd} ranked signs in the dataset are closely related to the query sign but not the sign itself. The database sign label of the 1^{st} ranked sign is displayed on the screen. The labels are English text corresponding to the detected sign.

Experiments are performed using five sets of 500 signs from 5 different signers. One signer is a native sign language translator providing perfect shapes and movements while the other 4 are experienced sign language users. The five signers have different body sizes and hand movements with matching articulation of signs. This helps to test the robustness of the proposed algorithm with 3D sign language mocap data. Regular performance indicators such as recognition, precession and recall are used. Analyzing the individual categories of signs can help explore the advantages and drawbacks of our method against different models used exclusively for SLR. Two popular sensors used for this purpose are Kinect and leap motion sensors. In the following sections, we compare and describe the advantages of using 3D motion capture for sign language modeling compared with Kinect and leap motion sensors.

VI. RESULTS AND DISCUSSION

A. 3D sign capture setting

3D motion capture setup from vicon feeds the algorithm with 3D Indian sign language models. Mocap is a setup with 8 IR cams and a video camera. The signer must wear reflective markers on the body, which are captured by the system. The experimental setup is shown in Fig. 2. The cameras are placed at various heights, locations, and focus to capture full motion in every sign. The positions of the reflective markers are mapped to define signs accurately. Fig. 3 shows the template designed for 3D capture of Indian sign language. The 500 dataset signs are categorically selected to reflect daily usage of

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

the sign language. The categories are sports, directions, dishes and spices, behavioural nouns, art, entertainment, agriculture, household articles, insects, measures, places, equipment, trees, flowers and health, or medical. Each category does not have a fixed set of signs. The 500 signs from these categories are single-handed or two-handed signs articulated around the chest and face regions. The Mocap software is used to extract 3D position vectors of each marker in 3D space in each video frame. The marker positions are joint positions in the human body. Each joint is represented as a 3D coordinate with $l\,(x, y, z, t)$ in the entire sign. The 3D sign data is arranged as $N_J \times N_t \times 3$ matrix, where N_J is the number of joints, N_t is the number of frames, and 3 is the (x, y, z) position space. For 500 signs, the raw dataset of position values is formed into a multidimensional sign space of size $N_J \times N_t \times 3 \times 500$. Five such datasets are captured, one with a native signer and the other four with experienced signers having knowledge of sign language and also little practice of it. The proposed model is tested for its ability to recognize 3D sign data effectively and accurately. Further testing of various state-of-the-art action recognition methods is performed for 3D sign language mocap data for data validation. Finally, the proposed SLR model is compared with the state-of-the-art SLR models using 3D data.

B. Testing the proposed method using 3D Sign Language Mocap data

1) Phase-I: Structured database creation: The raw 3D positional data obtained from the mocap system is translated into a structured database with four classes. JRDs are computed with 3D joint locations in each sign frame. A motion threshold J_D^{tr} on JRD clusters the joints into MJ and NMJ between consecutive frames. The MJ and NMJ set forms a basis for the creation of the structured 4-class dataset in Phase-I. In Phase-I, the motion segmented joints in a sign act as pointers to determine the movements of MJ with respect to NMJ.

Tracking is an important attribute in sign language understanding. Fig.4 shows the sign frames in each class of the database. The division starts by computing Gross JRD between every MJ and every NMJ. GJRD forms a confusion matrix showing which MJ is moving close or away from NMJ. Most of the signs are articulated either near to the face–head combo or chest with one hand or two hands. Moreover, there are signs that use more than one track during the signing process. These signs are classified based on the last label appearing in the sign. For example, if the sign starts with one hand near the face and is converted into a two-hand sign, then this sign is classified into two hand face–head combo labels. If a single hand sign near the face–head combo moves to the chest position, then it is labelled as the single hand face–head database sign.

The 500 datasets are classified into 4 classes framing a structured database. The Phase-I algorithm classifies the 500 signs into 159 $L\,(Single - hand\_Face - Head)$, 132 $L\,(Single - hand\_Chest)$, 82 $L\,(Two - hand\_Face - Head)$, and 127 $L\,(Two - hand\_Chest)$. The minimum distance classifier classifies the dataset based on the minimum distance between MJ and NMJ in a sign. This is the advantage of this work compared to the Kinect and leap motion sensor sign language data. In Kinect and leap motion sensors data, these signs are difficult to identify because of occlusions in Kinect and no body reference points in the leap motion sensor.

2) Phase-II: shape and orientation motionlets: However, exact sign identification at this stage is a difficult task due to missing hand shapes and hand orientations in the space during movement. The shape and orientation attributes of a sign are measured from MJs leaving out the NMJs. The distance measures between MJs termed as MGJRD is a matrix of local distances among MJs of size $\left(\frac{L(L-1)}{2}\right) \times N_t$. Each row of the MGJRD is a motionlet for finger motions. Instead of combining the motionlets with "and" and "or" functions, we concatenate all motionlets in all frames of a sign. This model presents small variations among fingers for identifying the correct sign. The angles at each joint are measured with position vectors at that MJ and are represented as MGJRA matrix of size $L \times N_t$, where L is the number of MJs and N_t is the number of frames per sign. Similar to the distance among motionlets, the MGJRA is a combination of angles between motionlets.

3) Adaptive Kernel Matching: Adaptive kernels are constructed from the trajectory matrix in Phase-I along with MJ shape and orientation matrices of Phase-II for matching.

Adaptive multi kernel matching is proposed to identify the query sign in the four database categories obtained in Phase-I. The adaptive kernels describe 3D signs based on their trajectories, hand shapes, and orientations. All 3 kernels are formed between the query sign and database signs in Phase-I and Phase-II. The kernel matrices are independent of MJ and NMJs. For example, the trajectories kernel matrix for a 3D sign "I" is formed from GJRDs between 19 MJ and 38 NMJs for $N_t = 109$ frames. The query "I" from a different signer has the same number of MJ and NMJ with 132 frames. The trajectories kernel is formed between query and dataset signs by using (13). The resulting trajectories kernel matrix has size 109×132, independent of joints in motion. Furthermore, this model results in matching between every frame in the query and the database sign. Row maximum values are extracted and averaged on columns to determine the decision boundary p_r^{s-trj}. This value decides the similarity between signs. Similarly, hand shape and orientation kernels match the query sign to database signs. Database signs are iteratively ranked after each query input is classified to the new ranking. This accelerates the recognition as the signs are closely related to the previous query sign.

The proposed algorithm is tested with one databased model and multiple instances of query signs. Multiple instances, involving different orientations of the signer with varying sign speeds are recorded. Three hand speeds and six orientations with an angle of 10 degrees variation in horizontal and vertical directions are recorded. The three hand speeds determine that the query sign has almost equal sign frames, less than the database sign frames and more than database sign frames. We select three orientations in the horizontal direction and three in the vertical direction. Nine variations of the query sign are recorded with a single signer for testing. Therefore, for a single

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

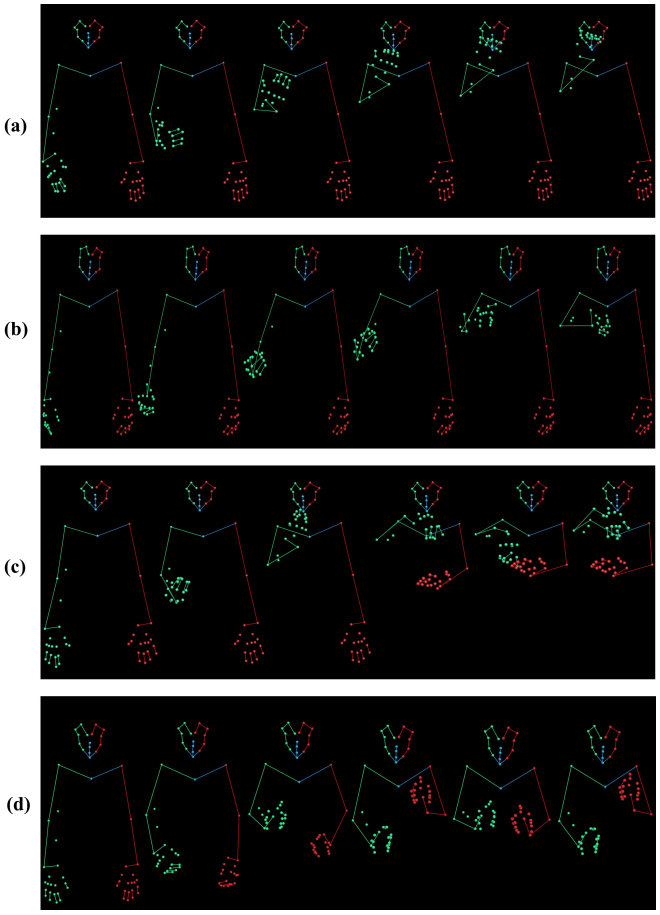

Fig. 4: 3D sign database classes (a) single hand head–face combo sign "Biscuit", (b) single hand–chest sign "My", (c) double hand face–head combo sign "Dish," and (d) double hand chest sign "Sports."

signer, we have 9 different variations of a sign. Similarly, recording is performed with 4 different experiment signers in 9 different orientations per sign. For testing a sign, we use 36 instances of a sign. For 500 signs, we record $36\text{x}500 + 500$ signs. The database has 500 signs while the testing set has 18000 signs with 36 variations per sign. Precision, recall, and recognition parameters are calculated for each of the 500 signs in the database. Table 1 shows some of the sign categories from the database.

The recognition rate determines the relativity of the query sign to that of a sign in the database. Precision and recall report the ability of the algorithm in delivering the outcome relevancy and de facto relevance outcomes, respectively. All the performance measures range within [0,1] with a value close to one, indicating maximum performance of the classifier algorithm. Table 1 shows the performance measures for 4 sign categories with values > 0.9 by using the proposed algorithm. Multiple testing on the proposed datasets resulted in same performance values. To understand the effects of small number of misclassifications, we analysed each sign in 4 database classes. The analysis reveals that the two-hand signs were difficult to recognize compared with the single-hand signs. Fig. 5 and 6 show the proof for the analysis presented using

TABLE I: Performance parameters of the proposed method for a set of 3D signs

Categories	Signs	Precision	Recall	Recognition
Sports	Sports	0.9925	1	0.9891
	Basketball	0.9808	0.9527	0.9935
	Ball	0.994	0.9867	0.9805
	Goal	0.9734	0.9927	0.9893
	Trophy	0.9551	0.9906	0.9968
Dishes and spices	Bread	0.9587	1	0.9739
	Biscuit	1	0.9927	1
	Salt	0.9867	0.9857	0.9802
	Curry	0.9867	0.9678	0.9789
	Curd	0.9837	1	1
Health or medical	Alive	0.9627	1	0.9896
	Cold	0.9195	0. 9167	0.9191
	Cure	0.9125	0.8937	0.9278
	Health	0.9824	0.9862	0.9567
	Wound	1	0.9586	0.9987
Household articles	Balcony	0.9573	0.9487	0.9875
	Kitchen	1	0.9679	0.9678
	Bed	0.9678	0.9876	0.9768
	Fan	0.9975	1	0.9758
	Mat	0.9687	0.9339	0.9759

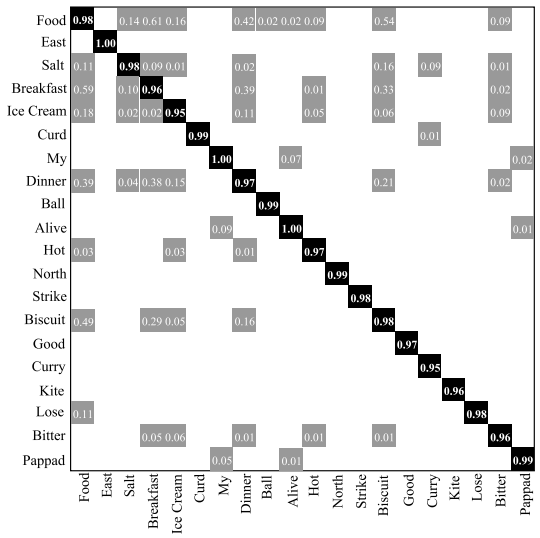

Fig. 5: Confusion matrix for single hand signs in various categories

confusion matrices. Fig. 5 shows the confusion matrix for single-hand signs and Fig. 6 shows that for two-hand signs.

The confusion matrix in Fig. 5 shows the uniqueness of the adaptive kernel matching for three mixed features, although they are closely related. For example, the sign breakfast distantly relates to signs "dinner," "biscuit," "food," "salt," and "ice cream." The reason is quite simple. They have common motionlet signatures as they belong to same category of dishes. Moreover, all signs use hand and some portions on the face, that is, mouth to construct a sign. These signs are prone to misclassification among themselves by using the sensor gloves, RGB video sensor, Kinect, and leap motion sensors. However, 3D mocap data with the proposed framework uniquely recognizes these signs.

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

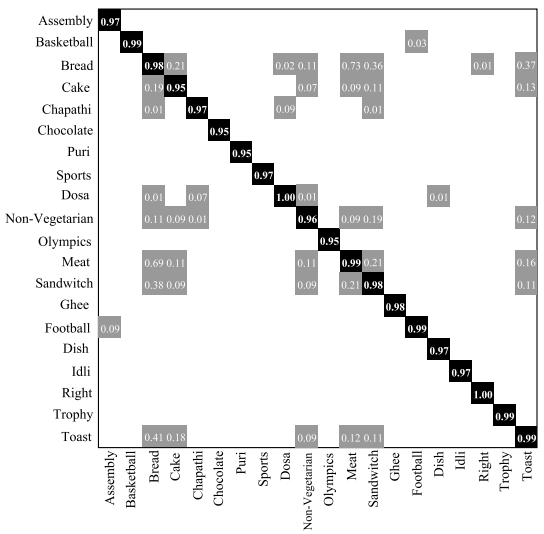

Fig. 6: Confusion matrix for two hand signs

C. Validating the proposed method

The proposed method was validated against the state-of-the-art action recognition algorithms such as DWT [4], recurrent neural network (RNN) [5], histogram of motion segments [32], weighted graph matching [33], and KNN + support vector machine (SVM) [34]. All the algorithms were presented with the GJRD. MGJRD and MGJRA were used as a 3-layer matrix per sign for training and testing. For RNN and KNN + SVM, large training sets comprising 2700 sign samples and 900 signs were tested. For RNN, the 3D data is converted to RGB images by encoding each plane with GJRD, MGJRD, and MGJRA. The proposed algorithm has a precision of 0.983, recall of 0.978, and recognition of 0.989 compared with algorithms from the literature [4][5][32][33][34]. Most of the algorithms are dependent on number of frames and location of sign in a sign video for matching. Our proposed algorithm is immune to the number of frames and location of signed frame and also the number of joints in the frame. DTW works well in situations where the number of frames between query and database is less than 10. RNN requires huge training times with multiple example sets for training. Histogram is a probability based model, where missing information is unpredictable. Weighted graph matching has too many imposed constraints such as walk length, node matching, and frame-by-frame matching, making it slower and unreliable during missing joint data. The reliability of K-NN and SVM reliability is questionable if the sign similarity is maximum in some cases. Only 100 samples were used to calculate the performance measures in table–II. The table values are averaged for 36 testings of a same sign.

The proposed recognition method based on kernel matching has the distinct advantages of using a linear combination of multiple kernels for generating a single matching score. The training database does not require a large set of observations. The number of iterations required in the proposed method is limited due to its structured representation based on the Phase-I procedure.

TABLE II: Performance compared with state-of-the art action recognition algorithms

Categories	Precision	Recall	Recognition
Dynamic Time Warping [4]	0.8175	0.7934	0.7997
RNN [5]	0.944	0.927	0.942
Histogram [32]	0.812	0.803	0.807
Weighted Graph Matching [33]	0.911	0.918	0.927
K-NN + SVM [34]	0.897	0.874	0.904
Our Proposed	0.983	0.978	0.989

D. Validating the sensors used to capture 3D SLR

To project the practicality of the data capturing model for sign language recognition, we juxtaposed our 3D mocap SL data with popular 3D capturing sensors, such as Microsoft Kinect and leap motion sensor.

TABLE III: Sensor based comparison of SLR models with the proposed mocap system

Technology	Algorithm	Accuracy
Microsoft Kinect	Cao Dong etal. [6]	90
	Pradeep Kumar etal. [7]	83.77
	Hee-Deok Yang etal. [35]	90.4
	Santiago-Omar etal. [36]	96.2
	Chana Chansri etal. [37]	84.05
	Cicero Ferreira etal. [38]	96.31
Leap motion	Pradeep Kumar etal. [39]	97.85
	Basma Hisham etal. [40]	95
	LuisQuesada etal. [41]	96
	Mohandes M etal. [8]	99.1
	Elons A S etal. [42]	88
	Ching-Hua Chuan etal. [9]	79.83
Motion Capture	Our Proposed model	98.9

This is the first study reporting recognition of 3D Indian sign language based on 3D motion capture. The previous works used either Microsoft Kinect or leap motion sensors for 3D sign language classification. The works on Microsoft Kinect are based on depth data featured along with RGB video with wide range of classifiers from minimum distance to deep CNNs. The recognition accuracy is close to 0.92 for most of the classifier models. However, most of the signs reported are simple in movement and single-handed. Interobject occlusions are difficult to handle, and the rebuilding feature matrix of the occluded part is a complex phenomenon, resulting in misclassifications. The leap motion sensor is a 3D model based approach for signs that are single handed or two handed. Moreover, signs are not finger-based entities, and include a complex movement of hands in 3D space with respect to the head, face as well as hand and torso. The sign must be performed above the sensor at a distance on the sensor. This sensor reported higher accuracies compare to 2D video approaches and Microsoft Kinect. Mocap system uses 360-degree view of the human signer, capturing all joints on the human object. Reconstruction is near perfect in every frame in the sign video. Joint losses during capture are almost nil compared to Kinect or leap motion sensor. The results in table–III reflect the superiority of the data used in capturing the sign language for processing. In this study, a recognition accuracy of 0.9894 is reported based on mocap data. This comparison is intended to shed light on the usage of markerless and marker-

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

based sensors for human action recognition or sign language recognition.

The idea of SLR is to machine translate visual information into text or voice and vice versa. The existing low-cost capture models such as 2D video or 3D depth capture using Microsoft Kinect or leap motion sensor does not fully comprehend the motions and shapes of a gesture in sign language. Microsoft Kinect uses dual cameras to depth map the captured object and relates it with a 3D skeleton model. The accuracy of the 3D skeletal model is questionable in many applications such as gait analysis [43]. However, researchers are exploring the possibility of using 3D depth maps and skeletal data to capture and process sign language gestures. This comparison will help SL researchers to use the 3D mocap models as a template with Microsoft Kinect for developing a low-cost SLR application.

VII. CONCLUSIONS

A model for recognizing gestures of Indian sign language 3D motion captured data is presented. The model builds a two phase algorithm which handles multiple attributes of 3D sign language motion data for machine translation. In phase–I, the unordered 3D sign database is restructured into a 4–class structured motionlet database from the measured trajectories of motion segmented 3D joints. Each action in a signed frame is motion segmented into motion joints and nonmotion joints. Phase–II extracts shape and orientation of 3D motionlets by applying joint relative distance and joint angle measurements respectively. Three feature kernels based on trajectories, finger shape and their orientations are constructed, which measure the similarity between the query signs and the database signs. It is observed that the motionlet based adaptive kernel matching algorithm on 500 class 3D sign language data gives better classification accuracies compared to state–of–the–art action recognition models. More importantly, it significantly optimizes the database search space by 75% over the existing kernel matching methods. Besides, a sensor based validation for sign language capture shows that 3D motion capture models have superior classification accuracies compared to Microsoft Kinect and leap motion sensor. Further, the 3D sign language model powers the augmented–reality–based sign language machine translator for building a real time mobile application.

ACKNOWLEDGEMENT

This work is supported under the sponsored research project scheme titled “Visual–Verbal Machine Interpreter Fostering Hearing Impaired and Elderly” by the “Technology Interventions for Disabled and Elderly (TIDE)” programme of the Department of Science and Technology, SEED division, Govt. of India, Ministry of Science and Technology, with file no. SEED/TIDE/013/2014(G).

REFERENCES

[1] S. Mitra and T. Acharya, “Gesture recognition: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[2] C. Sun, T. Zhang, B.-K. Bao, C. Xu, and T. Mei, “Discriminative exemplar coding for sign language recognition with kinect,” *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1418–1428, 2013.

[3] G. Plouffe and A.-M. Cretu, “Static and dynamic hand gesture recognition in depth data using dynamic time warping,” *IEEE transactions on instrumentation and measurement*, vol. 65, no. 2, pp. 305–316, 2016.

[4] D. Leightley, B. Li, J. S. McPhee, M. H. Yap, and J. Darby, “Exemplar-based human action recognition with template matching from a stream of motion capture,” in *International Conference Image Analysis and Recognition*. Springer, 2014, pp. 12–20.

[5] J. Martinez, M. J. Black, and J. Romero, “On human motion prediction using recurrent neural networks,” *arXiv preprint arXiv:1705.02445*, 2017.

[6] C. Dong, M. C. Leu, and Z. Yin, “American sign language alphabet recognition using microsoft kinect,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 44–52.

[7] P. Kumar, R. Saini, P. P. Roy, and D. P. Dogra, “A position and rotation invariant framework for sign language recognition (slr) using kinect,” *Multimedia Tools and Applications*, pp. 1–24, 2017.

[8] M. Mohandes, S. Aliyu, and M. Deriche, “Arabic sign language recognition using the leap motion controller,” in *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*. IEEE, 2014, pp. 960–965.

[9] C.-H. Chuan, E. Regina, and C. Guardino, “American sign language recognition using leap motion sensor,” in *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*. IEEE, 2014, pp. 541–544.

[10] G. Fang, W. Gao, and D. Zhao, “Large-vocabulary continuous sign language recognition based on transition-movement models,” *IEEE transactions on systems, man, and cybernetics-part a: systems and humans*, vol. 37, no. 1, pp. 1–9, 2007.

[11] M. Funasaka, Y. Ishikawa, M. Takata, and K. Joe, “Sign language recognition using leap motion controller,” in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015, p. 263.

[12] M. Mohandes, M. Deriche, and J. Liu, “Image-based and sensor-based approaches to arabic sign language recognition,” *IEEE transactions on human-machine systems*, vol. 44, no. 4, pp. 551–557, 2014.

[13] G. Marin, F. Dominio, and P. Zanuttigh, “Hand gesture recognition with jointly calibrated leap motion and depth sensor,” *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14 991–15 015, 2016.

[14] A. Kurakin, Z. Zhang, and Z. Liu, “A real time system for dynamic hand gesture recognition with a depth sensor,” in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 1975–1979.

[15] Z. Cai, J. Han, L. Liu, and L. Shao, “Rgb-d datasets using microsoft kinect or similar sensors: a survey,” *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4313–4355, 2017.

[16] D. Schroeder, F. Korsakov, C. M.-P. Knipe, L. Thorson, A. M. Ellingson, D. Nuckley, J. Carlis, and D. F. Keefe, “Trend-centric motion visualization: Designing and applying a new strategy for analyzing scientific motion collections,” *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2644–2653, 2014.

[17] A. Ahmadi, F. Destelle, L. Unzueta, D. S. Monaghan, M. T. Linaza, K. Moran, and N. E. OConnor, “3d human gait reconstruction and monitoring using body-worn inertial sensors and kinematic modeling,” *IEEE Sensors Journal*, vol. 16, no. 24, pp. 8823–8831, 2016.

[18] C. Chen, R. Jafari, and N. Kehtarnavaz, “A survey of depth and inertial sensor fusion for human action recognition,” *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4405–4425, 2017.

[19] J. Molina, J. A. Pajuelo, and J. M. Martínez, “Real-time motion-based hand gestures recognition from time-of-flight video,” *Journal of Signal Processing Systems*, vol. 86, no. 1, pp. 17–25, 2017.

[20] T. Kapuscinski, M. Oszust, and M. Wysocki, “Recognition of signed dynamic expressions observed by tof camera,” in *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2013*. IEEE, 2013, pp. 291–296.

[21] S. G. M. Almeida, F. G. Guimarães, and J. A. Ramírez, “Feature extraction in brazilian sign language recognition based on phonological structure and using rgb-d sensors,” *Expert Systems with Applications*, vol. 41, no. 16, pp. 7259–7271, 2014.

[22] J. C. Chan, H. Leung, J. K. Tang, and T. Komura, “A virtual reality dance training system using motion capture technology,” *IEEE Transactions on Learning Technologies*, vol. 4, no. 2, pp. 187–195, 2011.

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2810449, IEEE Sensors Journal

[23] E. Hegarini, A. B. Mutiara, A. Suhendra, M. Iqbal, and B. A. Wardijono, "Similarity analysis of motion based on motion capture technology," in *Informatics and Computing (ICIC), International Conference on*. IEEE, 2016, pp. 389–393.

[24] Z. Liu, L. Zhou, H. Leung, and H. P. Shum, "Kinect posture reconstruction based on a local mixture of gaussian process models," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 11, pp. 2437–2450, 2016.

[25] J. Gu, X. Ding, S. Wang, and Y. Wu, "Action and gait recognition from recovered 3-d human joints," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 4, pp. 1021–1033, 2010.

[26] H. Cheng, Z. Dai, Z. Liu, and Y. Zhao, "An image-to-class dynamic time warping approach for both 3d static and trajectory hand gesture recognition," *Pattern Recognition*, vol. 55, pp. 137–147, 2016.

[27] J. Li, X. Mao, X. Wu, and X. Liang, "Human action recognition based on tensor shape descriptor," *IET Computer Vision*, vol. 10, no. 8, pp. 905–911, 2016.

[28] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[29] X. Wang, L. Gao, J. Song, and H. Shen, "Beyond frame-level cnn: Saliency-aware 3-d cnn with lstm for video action recognition," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 510–514, 2017.

[30] M. Li and H. Leung, "Graph-based approach for 3d human skeletal action recognition," *Pattern Recognition Letters*, vol. 87, pp. 195–202, 2017.

[31] B. Wu, C. Yuan, and W. Hu, "Human action recognition based on context-dependent graph kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2609–2616.

[32] M. Barnachon, S. Bouakaz, B. Boufama, and E. Guillou, "Ongoing human action recognition with motion capture," *Pattern Recognition*, vol. 47, no. 1, pp. 238–247, 2014.

[33] Q. Xiao, Y. Wang, and H. Wang, "Motion retrieval using weighted graph matching," *Soft Computing*, vol. 19, no. 1, pp. 133–144, 2015.

[34] I. Kapsouras and N. Nikolaidis, "Action recognition on motion capture data using a dynemes and forward differences representation," *Journal of Visual Communication and Image Representation*, vol. 25, no. 6, pp. 1432–1445, 2014.

[35] H.-D. Yang, "Sign language recognition with the kinect sensor based on conditional random fields," *Sensors*, vol. 15, no. 1, pp. 135–147, 2014.

[36] S.-O. Caballero-Morales and F. Trujillo-Romero, "3d modeling of the mexican sign language for a speech-to-sign language system," *Computación y Sistemas*, vol. 17, no. 4, 2013.

[37] C. Chansri and J. Srinonchat, "Hand gesture recognition for thai sign language in complex background using fusion of depth and color video," *Procedia Computer Science*, vol. 86, pp. 257–260, 2016.

[38] C. F. F. Costa Filho, R. S. d. Souza, J. R. d. Santos, B. L. d. Santos, and M. G. F. Costa, "A fully automatic method for recognizing hand configurations of brazilian sign language," *Research on Biomedical Engineering*, vol. 33, no. 1, pp. 78–89, 2017.

[39] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "A multimodal framework for sensor based sign language recognition," *Neurocomputing*, 2017.

[40] B. Hisham and A. Hamouda, "Arabic static and dynamic gestures recognition using leap motion," 2017.

[41] L. Quesada, G. López, and L. Guerrero, "Automatic recognition of the american sign language fingerspelling alphabet to assist people living with speech or hearing impairments," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2017.

[42] A. Elons, M. Ahmed, H. Shedid, and M. Tolba, "Arabic sign language recognition using leap motion sensor," in *Computer Engineering & Systems (ICCES), 2014 9th International Conference on*. IEEE, 2014, pp. 368–373.

[43] E. Ceseracciu, Z. Sawacha, and C. Cobelli, "Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: proof of concept," *PloS one*, vol. 9, no. 3, p. e87640, 2014.

P.V.V.Kishore is a professor of Image & Video Processing with the department of Electronics and Communications Engineering, where he manages the Image, Speech and Signal processing Research Group. He went on to study M.Tech at Cochin University of science and technology and Ph.D. from Andhra University College of engineering in 2013. He is the chair of the Biomechics and vision computing research center.

His works focus on mechine learing, biomechanics, artificial intelligence, human motion analysis and sign language machine translation. His research explores how motion capture data models can effectively model low end video objects in real time for better recogntion and analysis. He is particularly intersted in developing new innovations in the areas of computer vision and mechine learing. He has authored several publications in these fields.

D. Anil Kumar received the B.Tech degree from the University of JNTUK, India in 2014, and the M.Tech degree from the university of Koneru Lakshmaiah Education Foundation, in 2016. He is currently pursuing the Ph.D. degree with the department of Electronics and Communications Engineering from the same university. His research interests are in the area of video processing, computer vision and sign language machine translation. His work is mainly focused on the development of 3-D processing algorithms for computer vision applications.

A.S. Chandra Sekhara Sastry received the M.Tech. and Ph.D. degrees in Electronics and Communications Engineering from JNTU college of engineering, Kakinada, India. Currently, he is a Professor with the Department of Electronics and Communication Engineering and he is the associate dean academics in the University of Koneru Lakshmaiah Education Foundation.

His research includes adaptive signal processing, biomedical signal processing, medical image processing and human-computer interaction. In these areas, he has authored more than 65 publications in journals and conferences.

E.Kiran Kumar received the B.Tech degree in Electronics and Communication Engineering from the JNT University, Kakinada, India, in 2009, M.Tech degree in Systems and Signal Processing from the same University, in 2013, specializing in evolving optimized object segmentation and recognition. Currently, he is working as Junior Research Fellow in the project "Visual–Verbal Machine Interpreter Fostering Hearing Impaired and Elderly" and pursuing the Ph.D. degree from University of Koneru Lakshmaiah Education Foundation, India. His research interests include the analysis of musculoskeletal movements of hand and movement strategies of the wrist and fingers in Indian sign language recognition.

1558-1748 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.