SignPose: Sign Language Animation Through 3D Pose Lifting

Shyam Krishna*, Vijay Vignesh P*, Dinesh Babu J
IIIT Bangalore
Bangalore
krishnshyam@gmail.com, vijayvigneshp02@gmail.com, jdinesh@iiitb.ac.in

Abstract

Sign Language Generation (SLG) is a challenging task in computer animation as it involves capturing intricate hand gestures accurately, for several thousand signs in each sign language. Traditional methods require expensive equipment and considerable human involvement. In this paper, we provide a method to automate this process using only plain RGB images to generate sign poses for an avatar - the first of its kind for SLG. Current state of the art models for human 3D pose estimation do not perform satisfactorily in SLG due to the large difference between tasks. The datasets they are trained on contain only tasks like walking and playing sports, which involve significantly different types of motion compared to signing. Synthetic, manually created 3D animations are available for diverse tasks including sign language performance. Modern 2D pose estimation models which work on real world images are also robust enough to work on these animations accurately. Inspired by this, we formulate a novel method of leveraging animation data, using an intermediate 2D pose representation, to train an SLG animation model that works on real world sign language performance videos. To create the dataset for training, we extend an available animated dataset of signs in the Indian Sign Language (ISL) by permuting different hand and body motions. A novel quaternion based architecture is created to perform the task of lifting the 2D keypoints to 3D. The architecture is simplified to match the requirements of our task as well as to work with our smaller dataset size. We train a model, SignPose, using this architecture on the constructed dataset and demonstrate that it matches or outperforms current models for human pose reconstruction for the Sign Language Generation task. We will release both the dataset as well the model to the public to encourage further research in this field.

1. Introduction

Sign languages are complete languages, having their own grammar and syntax. Unlike spoken languages, which have a standard simple textual representation for communication, sign languages have no such text form. Children born Hearing Impaired (HI) have difficulty in learning spoken languages and by extension their reading comprehension is also low [17]. Therefore the only way to communicate or create media in sign languages is through realistic performance, either by a human interpreter, or through computer generated output, mostly using animation [2].

Sign Language Generation (SLG) is a developing field involving automatic generation of sign language in a visual medium. The most common mode of output involves 3D animations of a computer agent performing the signs. This requires a database of animations of all the signs in the specific sign language, which number in the several thousands. This database has traditionally been generated using either sophisticated motion capture equipment [2, 13], or using parametrized representation of signs to generate animations [1, 7, 20]. The first method requires expensive equipment and the second method creates robotic and unnatural results, besides requiring complicated annotation by hand for each sign. Recently, much work has been done in estimating 3D pose from RGB images. Sign languages generally have a paucity of data, but several video dictionaries of signs are available [28, 32]. In this paper we look into leveraging the techniques of monocular 3D pose estimation to generate animations for Indian Sign language (ISL), for use with such a dictionary.

Current 3D pose estimation models are for general human activity involving full body motion captured from different camera angles, without a primary focus on the hand configuration [14, 18]. With sign language videos, there is both a decrease and increase in complexity in different aspects of the problem. On the one hand, the person is relatively stationary, with the camera fixed directly before the person, capturing only the upper body. On the other, high accuracy is necessary in estimating the handshape, the most important part of signing. Current 3D pose estimation mod-

*indicates equal contribution

els fail at one of these aspects, failing to either work with images containing only the torso and upwards, or to capture accurate hand pose. The output avatar must also have realistic human features for ease of comprehension, especially facial features like eyes which are necessary for conveying expression. Current models output a featureless humanoid, which will require creating textures and clothing to make it seem more human. Some also generate a different mesh for each person, and even each frame [24]. This makes generating a reliable texture for use across different sign language performers very difficult. There is therefore a need to create a specific solution addressing these issues for the problem of pose estimation in the context of sign language animation.

Pose estimation models require large and diverse 3D pose datasets to train. The datasets are created mostly by having actors perform in a controlled environment and using expensive motion capture equipment to obtain 3D parameters [12, 19, 31]. We propose a novel approach of using completely synthetic, manually created animations as the dataset to train our model. An animated dataset of ISL signs [16] is used in conjunction with a 2D pose estimation model to learn human poses in real world sign language videos. Since this dataset is small, we have used a method of generating "pseudo-signs" by combining different hand and body poses to significantly increase the size of the dataset. While animation has previously been used to enhance datasets to make them look realistic enough to train image based pose estimation, this approach of using 2D pose estimation along with plain animation has not been done previously, to the extent of our knowledge.

This work makes the following contributions:

- We present a novel approach of using animations of a realistic looking human model to acquire 2D to 3D human pose data, and implement it in the context of sign language generation.
- We generate using this approach a dataset of 3D sign language poses. We utilize and extend an available database of 1300 signs performed by a fully textured model, creating a database of $> 100,000$ frames.
- We train a simplified, modular architecture based on previous 2D to 3D lifting models on this dataset. The simplification is done to exploit the constraints in our task as well as to work with our smaller dataset. We also use a novel method of using quaternions to parametrize the output. We compare this trained model with the state of the art in the field of 3D pose estimation and demonstrate that our model overcomes the issues they have with sign language generation.

The code, models and datasets associated with SignPose are opensourced.

2. Related work

2.1. 3D human pose estimation

With regards to our task, current 3D pose estimation models can be divided into models extracting pose directly from image [15, 18, 29], and those that convert a 2D pose to 3D, termed "lifting" [4, 27, 34].

The first method has the advantage of utilizing all information available in an image, but this also puts constraints on the training data. The visual difference between training data generated in a controlled lab environment versus the actual use-case of in-the-wild images needs to be accounted for. The training dataset must have a wide variation in visual features such as different people, clothing and backgrounds. This is sometimes overcome by modifying the dataset to mimic in-the-wild images, usually by adding different realistic textures and real world backgrounds [38]. Often, an intermediate or joint 2D representation of keypoints is also used to aid the process [25, 36]. These models may also estimate body shape parameters, which determine the body mesh of the person. These models largely work with the body mesh provided by [18] and [24]. However these meshes do not come with a realistic texture with clothes, hair and skin tone. Creating a uniform texture for these models is a challenging task, especially given that the mesh changes across different body types and there are variations across different frames of a video as well.

The second method utilizes 2D pose, estimated from the image using a separate model such as OpenPose [3] or MediaPipe [38]. These 2D pose estimation models are trained on a wide range of realistic images and work in most conditions. This method is therefore more robust to the visual variations associated with images, working purely with the geometric pose which abstracts these features away. This means that these models work well with in-the-wild images even if trained on datasets that are generated in controlled lab environments. However, this also means that the accuracy of these models depends on the the accuracy of the 2D pose estimation model. Our model builds from this research: animation training data visually differ from real world data, but 2D pose estimation models are robust enough to perform well on them.

The method of representing the 3D pose output also varies across models. Models either directly estimate the depth coordinate [21, 27] which gives the output in terms of 3D Cartesian points, or they estimate joint rotations [6, 18], which is in line with our model. In the case of the first method, additional constraints of bone length need to be taken care of [27], and Inverse Kinematics needs to be performed to animate the 3D model. Using rotations, on the other hand, gets rid of these length constraints, and animation is achieved through much simpler forward kinematics.

2.2. 3D human pose datasets

There are several datasets of 3D human pose data, generally stored in the form of motion captured data performed in a controlled environment. There are many large scale datasets for full body human poses [8, 12, 19], and a few datasets of only hand poses [23, 39]. Models estimating both body and hand poses usually combine data from both these kinds of datasets [24, 29]. This requires explicit alignment of the two components and leads to greater error in the common joint between the body and the hand, usually the wrist. As the datasets are recorded in controlled lab environments with motion capture devices, they need modifications to work in real world cases. Some approaches use synthetic 3D models with real world background images to generate realistic images for training robust models [12], which resembles our approach of employing synthetic training data for real world use. These datasets are also prohibitively expensive to create, due to the equipment cost. On the other hand, we have used an entirely synthetic and manually animated dataset, created without any specialized equipment, to generate our 2D to 3D pose data.

2.3. Sign language animation

When animating sign languages, accuracy in reproducing meaningful components such as handshape and place of articulation takes prime importance. The intricacy of finger poses and the relatively small space they take up visually adds to the complexity of the problem. This is further compounded by the fact that the process needs to work for a significantly large number of unique signs, numbering in the thousands in each sign language.

Older work on sign language animation generally focused on motion capture methods which, besides having a high cost, require significant human oversight of the process [2]. Alternatively, this process was automated to an extent by generating signs at run-time using a parametrization of signs [7]. The methods of parametrization were either based on general sign language transcription systems such as HamNoSys [20] or SigML [10], or used newly constructed parameters tailored to the individual sign language [1]. These parametrizations capture sign language features such as the places of articulation, handshapes and expressions to varying degrees. However, the parametrization of each sign needs to be annotated manually, which is a lengthy process requiring specialized knowledge [11]. This method also ends up losing the subtle differences in articulation that occur when the same features are present in different signs, making the animated output seem robotic. More recently, there have been attempts to completely automate the animation process, using 3D human pose estimation methods. Zelinka *et al*. [37] come very close to our work lifting 3D pose from 2D, although they simply generate a 3D stick skeleton, and not a human avatar animation.

(a) (b)

Figure 1: Sample pose from the dataset (a), and another pose with an OpenPose overlay (b).

Parametrized SLG animation as described above is animated and designed by hand. Manual animation can be taken further to generate entire signs, creating a database of hand-animated signs to be used for SLG. Krishna *et al*. [16] utilize this method to generate ISL output, and they have also made available the dataset of these manually animated signs. This dataset serves as the base dataset for training our model.

3. Dataset generation and augmentation

The primary dataset used in this paper is obtained using the dataset presented in [16]. This is a dataset of 1300 animations of signs in ISL. There is a wide variety of hand and body poses, and more importantly, hand and body poses come integrated, instead of being separate. This dataset also comes with a realistic looking model that is used to perform the animations, allowing for a finished output of sign animation (Fig. 1a). The model is realistic enough that existing 2D pose estimation models such as OpenPose [3] and MediaPipe [38] work very accurately on it (Fig. 1b). Using these 2D pose estimates solves the problem of the model needing to be robust across visual changes, by avoiding using the image directly, and delegating this issue to more robust 2D pose estimation models.

Various forms of 3D pose parametrizations are obtainable from the animations, including 3D joint locations, Euler angle representations, as well as rotation quaternions. Our dataset consists of 2D pose estimates as generated by OpenPose on videos of the animations, along with the corresponding 3D pose represented by rotation quaternions. The videos have been generated using a fixed camera before the model at approximately chest height.

The animation dataset is fairly small in size, and we augment it by generating "pseudo-signs". This process involves generating permutations of the hand portion of one animation with the body portion of a different animation. This process ensures that, in the new dataset, both the handshapes as well as the body motion are typical of ISL. This data is filtered, keeping only data where OpenPose recognized all 54 joints of our model input. The final number of frames in the dataset is 102,582, which is split into 96,419 train and 6,163 test frames.

4. Methodology

Our project, SignPose converts an image of sign language performance to 3D pose for a human avatar. It starts with running a 2D pose estimation model to obtain the 2D keypoints, which serve as input to our model. Our model then converts the 2D pose into 3D. The model is composed of two separate models for the hands and the body, and the output from these two is combined to get the full body 3D pose. This 3D pose is compatible with the avatar we use, and is used to animate it, completing the pipeline.

4.1. Model input and output

The function of our model is converting a 2D representation of the pose, $\mathbf{p} \in \mathbb{R}^{j\times 2}$ into a 4D representation of the 3D human pose, $\mathbf{P} \in \mathbb{R}^{J\times 4}$. Here j is the number of joints in 2D pose, and J in the 3D pose, as these can be different. $\mathbf{p}$ is the list of the j 2D coordinates of the joints, and $\mathbf{P}$ is the list of the J quaternion parameters, each joint having 4 parameters. In both the 2D and 3D representations, only the joints above the pelvis are used, since our use-case does not require estimation of the lower body, which remains stationary. Among these, the body model receives all joints except those of the hand as input. The set of joints that go into the hand model consists of the joints of the arm along with those of the hand. This is to better estimate the orientation of the whole hand with respect to the rest of the body. This is done separately for each hand, which means the hand model runs twice per image, once for each hand.

Random noise is added to this input, in the form of rotation of the bones ($\leq 10°$), scaling ($\leq 40\%$), translation of the entire 2D pose ($\leq 20\%$ for both axes). We find this makes the model robust to data in-the-wild.

4.2. 3D pose parametrization

Previous work on 2D to 3D lifting that we encountered infer either 3D joint coordinates, or joint rotations. Rotations were represented either using Euler angles, or in the axis-angle notation form.

Using 3D joint coordinates comes with the issues of having to constrain bone lengths, significantly complicating model calculations. Furthermore, given the coordinates, we need to perform Inverse Kinematics in order to generate the pose. Using joint rotations bypasses both these issues: bone lengths are independent of rotations, and generating the pose is done through simple Forward Kinematics. There is, however, an added variation in 3D coordinates of the joints if the same rotation parameters are applied to different models having bones of different lengths. Since we use the same model for training and output, this issue does not appear in our case.

Representations for 3D rotation such as Euler angles and axis-angle representation exist in $\mathbb{R}^3$, and it has been shown that this ends up causing discontinuities or singularities in the representation space. A discontinuity in a rotation representation space refers to a subspace within which there is no change in the rotation of the object, alternatively, within this subspace no rotation is possible [9]. This manifests as issues like gimbal lock and the infinite number of representations due to periodicity. Quaternions, however, exist in $\mathbb{R}^4$ and do not have these issues. They are also more compact and computations involving them are more efficient. Furthermore, we obtain them directly from the animation, through Blender. This is in contrast to [26], another work that uses quaternions, where they had to be calculated and an additional constraint of quaternion length had to be added.

A quaternion $\mathbf{q}$ is represented as either a 4-tuple or a 4-dimensional vector $[w, x, y, z]$. Here w is the real term and x, y & z are the complex terms.

$$\mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

where $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are hypercomplex numbers satisfying the equation:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

Quaternions represent rotation around a vector $\hat{e}$ by an angle θ, by the equations $w = \cos(\theta/2)$ and $[x, y, z] = \hat{e}\sin(\theta/2)$.

4.3. Model architecture

We have based our model network on previous work which involved 2D to 3D lifting, mainly [5] and [21]. Fig. 2 gives an overview of our architecture. The network consists of a fully connected layer which encodes the 2D pose to a 4096-dimensional vector, which is then fed into a single residual block. This block consists of a fully connected layer, 1D batch normalization, ReLU activation, and dropout. The fully connected layer has the same dimension of 4096, and dropout probability is set to 0.5. Finally a last fully connected layer converts the output of the residual block into 4D quaternion parameters.

The primary difference in our model is that we use a single residual block instead of two blocks as suggested by previous work. Using two blocks on our smaller dataset

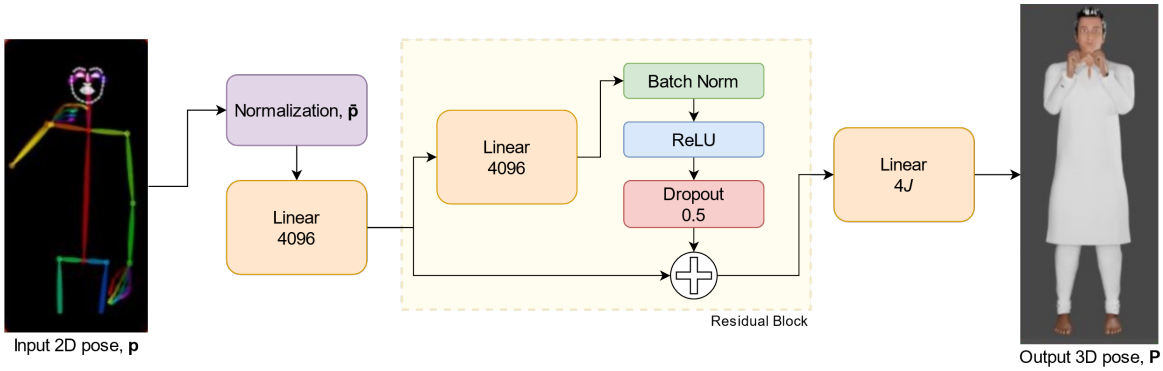

Figure 2: Architecture of the model network.

ended up causing issues with overfitting and we found using a single block was optimal for our task. We also based this simplification on the fact that only one camera view is utilized, as well as the fact that fewer joints are being estimated in the body.

The loss function we have used is the squared L2 norm between the predicted quaternions and ground truth quaternions i.e., the Mean Squared Error:

$$L_{quaternion} = \|\mathbf{P} - \mathbf{P}^*\|_2^2$$

$$= \sum_{i=1}^{J} \frac{(w_i - w_i^*)^2 + (x_i - x_i^*)^2 + (y_i - y_i^*)^2 + (z_i - z_i^*)^2}{4J},$$

where the asterisks represent predicted values.

4.4. Evaluation

As is standard in human 3D pose estimation, we have used the mean per joint position error (MPJPE). Since our dataset is completely synthetic, absolute metrics in mm needs to be estimated. To convert values to mm, we compared human femur length (taking the average value of 452.7mm) to the thigh bone length of our model, and calculated the rest of the values with regards to this. Another metric tested for is the Percentage Correct Keypoints (PCK), which measures the percentage of keypoints that are detected within a certain accuracy. We measured PCK@150mm, which is the standard.

There is a significant difference in the tasks present in datasets used in evaluation of existing models, versus our dataset. Furthermore, the animation avatar model that we have used differs from existing models in joint configuration and comparing evaluation results becomes difficult. Hence we only evaluated our model against our dataset while comparing results. While these results do not fit a specific competitive context, they provide a good benchmark to compare our model performance. Also given we are releasing our dataset to the public, we make our model open for future competitive evaluation.

4.5. Implementation details

The videos of each sign are generated using Blender, keeping the camera fixed across all animations. OpenPose whole body pose estimation is run on these videos to obtain the 2D pose at each frame. The network is constructed using Pytorch. Adam optimizer is used to update weights with minibatch size of 80. The learning rate is set to 10^{-4} and after 30 epochs, it is decreased to 10^{-5}. Two models are independently trained: one for the body without the fingers, and one for both the hands. The input for the body model has $j = 54$ (including face keypoints) and the output, $J = 14$. For the hand model, these are $j = 49$ (including arm joints) and $J = 38$. The models are run for 60 epochs. The training is done on a single NVIDIA RTX 2080 Ti GPU with 12 GB RAM, and takes around half a day to run on our dataset.

5. Results

5.1. Quantitative results

Quantitatively, the values we get for the metrics MPJPE and PCK@150mm on our datasets are significantly better compared to other models on general human pose datasets. Table 1 shows the MPJPE and PCK@150mm results of existing models. The MPJPE values are on the Human3.6M [12] dataset, which we noted gave the best results for those models, and the PCK@150mm values are on the MPI-INF-3DHP [22] dataset, which is more complex. Table 2 lists the evaluation results of our hand model on our dataset as compared to existing models on the EHF [24] and FreiHAND [39] datasets. Our left hand and average prediction values were comparable to these models, but the right hand performs poorly. However, the values reported here for

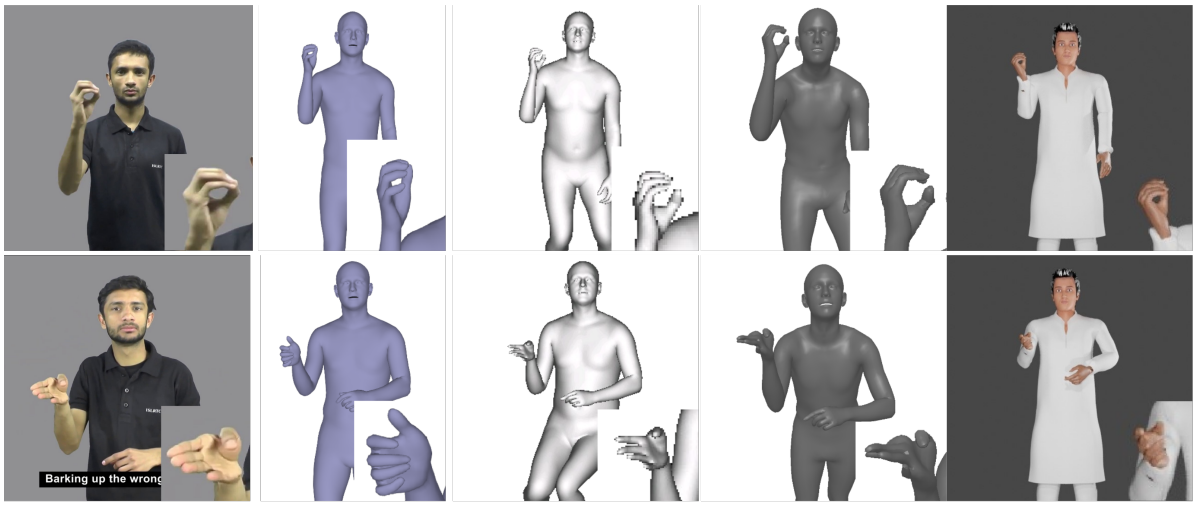

Figure 3: Results on two different signs from the ISLRTC dictionary [28]: (L-R) original, ExPose, FrankMocap, SMPL-X, SignPose (ours).

Table 1: Comparison with existing body models

Model	MPJPE	PCK@150mm
Pavlakos '18[25]	56.2	71.9
Yang '18[36]	58.6	69.0
Kolotouros '19[15]	**41.1**	76.4
Pavllo '19[27]	46.8	-
Chang '20[4]	52.5	83.9
Choi '20[5]	64.9	-
Sarandi '20[30]	49.3	**89.6**
SignPose (Ours)	**27.09**	**99.0**

Table 2: Comparison with existing hand models: the first set is on the EHF and the second set on the FreiHAND datasets

Model	Joint Error	
	L/R Hand	Both
SMPL-X [24]	**12.2**/13.5	**12.8**
MTC [35]	16.3/17.0	16.6
ExPose [6]	13.5/12.7	13.1
MANO [39]	N/A	10.9
Pose2Mesh [5]	N/A	**7.4**
ExPose [6]	N/A	12.2
SignPose (Ours)	**11.7**/17.1	14.4

Table 3: Comparison of execution times

Model	Time (s)
SMPL-X	130
FrankMocap	0.5
ExPose	0.3
SignPose (Ours)	1.11 (1.1[a] +0.01)

[a] OpenPose running time

other models in PA-MPJPE, which uses ground truth values in performing Procrustes Analysis for alignment. For the body, we have noticed a reduction of more than 30% of the value of MPJPE (using data collated in [6]), which puts our hand MPJPE values well within the range of the others. As noted earlier there is incompatibility in direct comparison of these models and datasets versus ours. These results, however, demonstrate that the performance of our model on our datasets is comparable to the performance of state-of-the-art pose estimation models on general pose datasets. Since we are also releasing our dataset to the public, we expect more competitive comparisons from future work using our data.

Another important aspect of our model is the simplified architecture, which leads to short execution times. The model, given OpenPose input, processes 1 frame in 0.01s. However, OpenPose ends up taking 1.1s per frame, slowing the process significantly. Even with that included, our model is much faster than SMPL-X, but ExPose and FrankMocap perform quicker when this added OpenPose time is taken into consideration. Our model, therefore is

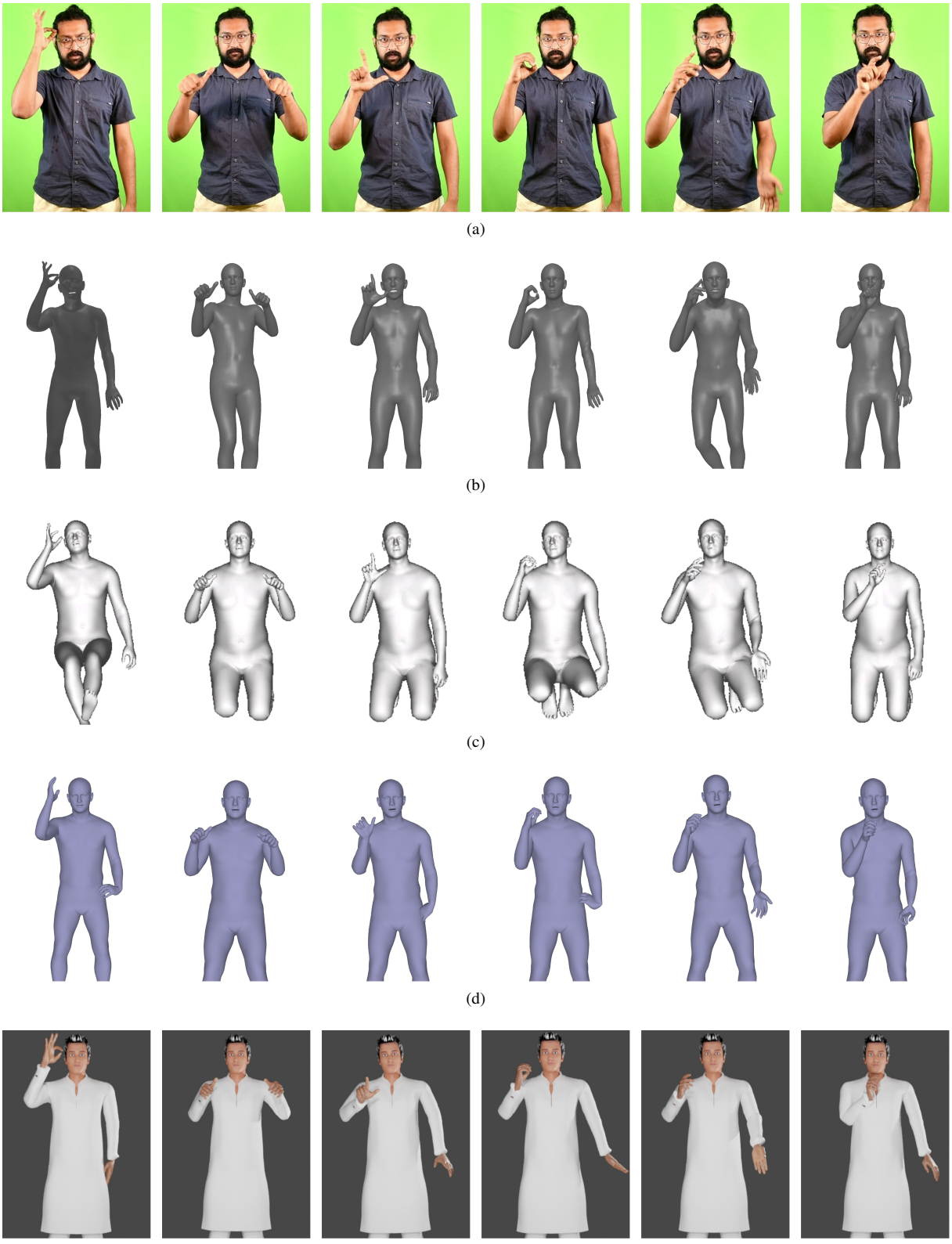

Figure 4: Qualitative comparison of outputs for various models: (a) original input , (b) SMPL-X [24] output , (c) FrankMocap [29] output, (d) Expose [6] output and (e) SignPose (our) output.

effectively limited by the speed of the 2D pose estimation model. MediaPipe, another 2D pose estimation model, performs much faster than OpenPose, running at 0.17s per frame and using it would bring down our processing time significantly below 1s and comparable to the other two models.

5.2. Qualitative results

The whole body outputs of our model versus current pose estimation models images are presented for five sample sign images in Fig. 4. While the upper body pose accuracy is comparable to our model, FrankMocap and SMPL-X perform poorly on lower body stability and accuracy on images where the entire body is not present. The error in the lower body output is very drastic in FrankMocap (Fig. 4c), where the legs end up various very unnatural positions. SMPL-X performs better, but there can be significant full body rotation which is not present in the original image (Fig. 4b). Our experiments with trying to use the upper body results only for these models gave very unsatisfactory results due to changes in the entire body orientation that occurs. Expose is better at full body estimation, but hand estimation is not very accurate (Fig. 4d, many extended fingers not estimated). ExPose also has some very unnatural left hand poses when the hand is out of frame (Fig. 4d, third image). Our model on the other hand, works very well with upper body only images, as well as maintaining hand pose accuracy.

Hand outputs comparison of our model against other current models is presented in Fig. 3 for signs in the ISLRTC dictionary. Here we can see how Expose still struggles with fingers in extension. SMPL-X performs better, but allows for a slightly unnatural looking hand shape in the second image, and body hunching is also evident. FrankMocap gives the best looking results for hands, with our model also performing comparably.

From the above examples, we can see that our model combines good accuracy in estimating full body pose, while also predicting the hand pose well. Furthermore, as can be seen, our model has human textures, making the output more fit for use in an end product for generating sign language. One drawback we noticed was that our model has some difficulty with the left hand when it is out of frame, though this is not as drastic as in ExPose.

5.3. Drawbacks and future work

One aspect of sign language that our work lacks is that it does not capture facial expressions. Facial expressions are an important feature in Sign Language communication. As can be seen in Fig. 4, both ExPose and SMPL-X are expressive, and they capture facial expressions. The dataset we use has a subset of signs which have expressions built into them, and we want to pursue it in the future, taking inspiration from works like FACSvatar [33]. Our model also fails to simulate larger motion of the head, and overall torso rotation and bending. This is due to the lack of such data in the dataset. Encouraged by our experiments with the extension of the dataset, we wish to experiment with automatically adding this variation to the animations to capture this behaviour. In the context of generating videos, our model ends up causing considerable jitter in motion as there is no dependency across different frames of the videos. Since our model keeps the lower body stationary, this jitter is lower than those of existing models, but is still a significant hurdle to realistic video generation. As our dataset is one of animation videos, sequence training is possible. In the future, we intend to incorporate sequence modeling into the pose estimation procedure to produce smooth video output.

6. Conclusions

We present a novel approach of utilizing manually created animations in the task of 3D human pose generation, specifically for Sign Language Generation. Using this approach on a previously existing animation dataset, we generate a dataset for 2D to 3D pose lifting for Sign Language animation. We simplify the current architecture for 3D lifting to better fit both, the input parameters specific to the task, as well as the smaller size of our dataset as compared to the ones the previous models have been trained on. A new way to represent the output pose using quaternions is implemented as a better representation of the output. We train a model, SignPose, using this architecture on the dataset created and demonstrate it performs competitively, while having a low computational footprint. Our work also produces outputs compatible with a realistic human avatar making it suitable for end-user consumption. We make both dataset and model available to the public to encourage further research in the field.

7. Acknowledgements

This work received funding from the Mphasis CSR grant as well as the MINRO grant to IIIT-B.

References

[1] Inês Almeida, Luísa Coheur, and Sara Candeias. From european portuguese to portuguese sign language. In *SLPAT@Interspeech*, 2015. 1, 3

[2] Andrew Bangham, Stephen Cox, John Glauert, I. Marshall, S. Rankov, and Mariah Wells. Virtual signing: capture, animation, storage and transmission-an overview of the visicast project. In *IEE Seminar on Speech and Language Processing for Disabled and Elderly People*, 2000. 1, 3

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2, 3

[4] Ju Yong Chang, Gyeongsik Moon, and Kyoung Mu Lee. Poselifter: Absolute 3d human pose lifting network from a single noisy 2d human pose. *arXiv e-prints*, 2020. 2, 6

[5] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision (ECCV)*, 2020. 4, 6

[6] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Monocular expressive body regression through body-driven attention. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 6, 7

[7] A. Conway and T. Veale. A linguistic approach to sign language synthesis. In *BCS HCI*, 1994. 1, 3

[8] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F. Troje. Movi: A large multipurpose motion and video dataset. *arXiv e-prints*, 2020. 3

[9] F. Sebastin Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3(3):29–48, Mar. 1998. 4

[10] Angus B. Grieve-Smith. Signsynth: A sign language synthesis application using web3d and perl. In Ipke Wachsmuth and Timo Sowa, editors, *Gesture and Sign Language in Human-Computer Interaction*, pages 134–145, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. 3

[11] Thomas Hanke. Sign language transcription with syncwriter. *Sign language & linguistics*, 4(1-2):275–283, 2001. 3

[12] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 2, 3, 5

[13] Richard Kennaway, John Glauert, and Inge Zwitserlood. Providing signed content on the internet by synthesized animation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14:15, 09 2007. 1

[14] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5252–5262, 2020. 1

[15] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 2, 6

[16] Shyam Krishna, Ankit Rajiv Jindal, Mahesh R, Rahul K, and Dinesh Jayagopi. Virtual indian sign language interpreter. In *Proceedings of the 2020 4th International Conference on Vision, Image and Signal Processing*, 2020. 2, 3

[17] Fiona Kyle and Kate Cain. A comparison of deaf and hearing children's reading comprehension profiles. *Topics in Language Disorders*, 35:144–156, 04 2015. 1

[18] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 1, 2

[19] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019. 2, 3

[20] Ian Marshall and Éva Sáfár. A prototype text to British Sign Language (BSL) translation system. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 113–116, Sapporo, Japan, July 2003. Association for Computational Linguistics. 1, 3

[21] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017. 2, 4

[22] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. 5

[23] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision*, 2020. 3

[24] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 5, 6, 7

[25] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 6

[26] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. Modeling human motion with quaternion-based neural networks. *International Journal of Computer Vision*, 128(4):855–872, Oct 2019. 4

[27] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6

[28] Indian Sign Language Research and Training Center. Islrtc new delhi. http://www.islrtc.nic.in/. 1, 6

[29] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. FrankMocap: Fast Monocular 3D Hand and Body Motion Capture by Regression and Integration. *arXiv e-prints*, page arXiv:2008.08324, 2020. 2, 3, 7

[30] Istvan Sarandi, Timm Linder, Kai O. Arras, and Bastian Leibe. Metric-scale truncation-robust heatmaps for 3d human pose estimation. *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, Nov 2020. 6

[31] L. Sigal, A. Balan, and M. J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1):4–27, Mar. 2010. 2

[32] Thomas Troelsgård and Jette Hedegaard Kristoffersen. An electronic dictionary of danish sign language. In *Theoretical Issues in Sign Language Research Conference, Florianopolis, Brazil*, 2008. 1

[33] Stef van der Struijk, Hung-Hsuan Huang, Maryam Sadat Mirzaei, and Toyoaki Nishida. Facsvatar: An open source modular framework for real-time facs based facial animation. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, IVA '18, page 159–164, New York, NY, USA, 2018. Association for Computing Machinery. 8

[34] Zeye Wu and Wujun Che. 3d human pose lifting: From joint position to joint rotation. In Yongtian Wang, Qingmin Huang, and Yuxin Peng, editors, *Image and Graphics Technologies and Applications*, pages 228–237. Springer Singapore, 2019. 2

[35] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6

[36] Wei Yang, Wanli Ouyang, X. Wang, J. Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. *Computer Vision and Pattern Recognition*, 2018. 2, 6

[37] Jan Zelinka and Jakub Kanis. Neural sign language synthesis: Words are our glosses. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 3

[38] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv e-prints*, page arXiv:2006.10214, June 2020. 2, 3

[39] C. Zimmermann, D. Ceylan, J. Yang, B. Russel, M. Argus, and T. Brox. FreiHAND: A dataset for markerless capture of hand pose and shape from single rgb images. In *International Conference on Computer Vision (ICCV)*, 2019. 3, 5, 6