3D Reconstruction of 2D Sign Language Dictionaries

Roman Riazantsev[1] and Maksym Davydov[2]

[1]Ukrainian Catholic University, Lviv, Ukraine
riazantsev@ucu.edu.ua
[2]ADVA Soft, Lviv, Ukraine
maks.davydov@gmail.com

Abstract. In this paper, we review different approaches to hand pose estimation and 3D reconstruction from a single RGB camera for converting 2D sign language dictionaries into animated 3D models. Unlike many other works aimed at real-time or near real-time translation, we focus on the quality of conversion given large video dictionary as input. Several approaches to training and validation are considered: pose reconstruction through depth estimation, training and validation with synthetic data, training and validation with multiple views. Besides that, the work provides a review of various end-to-end algorithms for keypoint detection trained on labeled data. Based on the results of the studied models, the outline of a possible solution to the 3D reconstruction task is proposed.

Keywords: Hand pose, Convolutional Neural Network (CNN), Sign Language

1 Introduction

Today virtual and augmented reality technologies (AR/VR) are becoming more and more popular. Such trend creates high demand for 3D image data processing, which applies to many areas. We focus our research on the conversion of available 2D sign language content into 3D. Our goal is to improve the quality of 3D reconstruction for video lessons of sign language. Sign language video dictionaries are widely available and reliable method for their conversion into 3D would create demanded content for use in AR and VR applications. Often people who want to learn sign language see only the front view of hands provided in 2D dictionaries. However, views from all angles carry value, as they reflect the nuances between similar words.

We aim at reconstruction specifically poses from sign language videos for the task of creating educational content in the future. Almost all of the other methods aimed at solving problems in general, but we propose a solution for specific subtasks, namely - reconstruction of sign language videos for further usage in AR/VR applications.

The task of pose reconstruction from a video is nontrivial and is not fully solved at the moment. The computational problems are related to blurred frames, which exist due to high speed of movement, and complex hand poses with overlapping hand parts along the z-axis. Often 3D reconstruction is performed with the usage of depth sensors, but

In: Proceedings of the 1st Masters Symposium on Advances in Data Mining, Machine Learning, and Computer Vision (MS-AMLV 2019), Lviv, Ukraine, November 15-16, 2019, pp. 2–9

there is much more available 2D data, which can be potentially mapped into 3D. Besides that, RGB camera is a more popular sensor, which can be used to record new information.

The different datasets can be used for training and testing of hand pose reconstruction models. There are many datasets with depth-camera input and 3D key points [1-3, 14-18], somewhat less datasets with 3D points and single RGB camera [4, 10, 11, 13, 19, 20]. The lack of multi-view Ukrainian Sign Language data prompts us to create a new dataset. Existing methods of hand pose reconstruction are reviewed in section 2. Section 3 outlines the proposed approach.

2 Background and Related Work

2.1 Background Overview

The task of determining the position of an object in space is not new. Over the past 20 years, a large number of works have been aimed at solving this problem [5, 6]. A lot has changed with the advent of depth sensors and neural networks. These technologies introduce new approaches to comprehensive scene analysis. Depth cameras produce information about the distance to an object, which allows reconstructions of more accurate 3D models, and neural networks calculate complex correlations in image patterns. Since 2012, neural networks started to outperform most of the classical methods in segmentation and classification problems. A large number of methods use a combination of depth-camera output and neural network for 3D reconstruction of the body position [7, 8]. The abovementioned technologies also apply widely to the hands. Often, researchers use a combination of depth sensors and gloves, which record the 3D position of the hand. Several sensors are used for collection of fully labeled training samples for 3D reconstruction, which may include depth map, joint angles, and 3D positions [1-3].

2.2 Related Works

Most methods for 3D hand pose generation from a single RGB image can be generalized into four stages (see Fig. 1). The first stage is detection of hands in the input image and cropping localized area, the second is detection of hand key points in 2D the third is mapping of 2D locations into 3D, and the fourth is generation of 3D hand model.

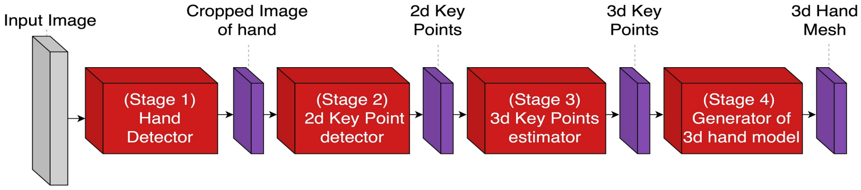
Input Image
Cropped Image of hand
2d Key Points
3d Key Points
3d Hand Mesh
(Stage 1) Hand Detector
(Stage 2) 2d Key Point detector
(Stage 3) 3d Key Points estimator
(Stage 4) Generator of 3d hand model

Fig. 1. A generalized schema of 3D hand pose estimation

Paper [9] introduces a three-stage algorithm that localizes the hands and determines the key points in 2D at the first two stages, and calculates 3D reconstruction at the third is studied in the paper. The first step is the YOLO (you only look once) neural network, which identifies the position of the hands, after which it cuts off this part of the image and passes cropped sub-images to the OpenPose detector. These two neural networks localize 21 2D key points in the video, which are then used as a target in the inverse kinematic optimization problem. A distinct drawback of this method is the limitation caused by the error of the OpenPose detector. This error causes the algorithm to optimize 3D locations using wrong 2D key points. Nevertheless, the addition of a hand position from a different view makes it possible to improve the optimization problem, and hence the accuracy. The runtime of the method on Nvidia GTX 1070 GPU is close to 53 ms.

Publication [21] describes one of the few methods, which fully reconstructs the 3D shape of the hand. It introduces graph convolutional neural network (CNN) for generating 3D mesh [21]. This work uses centered images of hands as input, thus hand detection was not necessary. Therefore, the first part of the approach is 2D key point detection, which is based on Stacked Hourglass Networks. The second part is the encoding of 2D features, and the third is 3D reconstruction using graph CNN network. The network outperforms the State-of-the-Art methods on RHD [12] and STB [13] datasets. The runtime of the method on Nvidia GTX 1080 GPU is on average 19.9ms. The pretrained model is available, but the training dataset is not provided.

2.3 Datasets Review

We examined several datasets and selected the most suitable for our task. Large portion of datasets for 3D reconstruction contain depth maps, key points, but not RGB image: NYU, ICVL, MSRA15, BigHand2.2M, SynHand5M, FHAD, MSRC (FingerPaint), HandNet, Hands in Action, MSRA14 [1-4, 14-18]. For the problem of reconstruction from single image. the most appropriate datasets are those featuring both RGB records and key points: FreiHAND, GANerated Hands, EgoDexter, SynthHands, STB, Dexter+Object, UCI-EGO, MHP [4, 10, 11, 13, 19, 20]. The possible complication of combining different datasets is that the number of key points, record types, and camera parameters may not match. From the available variety of datasets, we have selected only those with a central position of a hand and 21 labeled key points.

FreiHAND Dataset is a hand pose dataset for hand pose estimation from a single image. The dataset contains shots with 4 different backgrounds annotated with 21 key points for 2D and 3D spaces. There are 130240 of training samples, so 32560 images per one background [4].

GANerated Hands Dataset contains 330,000 examples annotated with 21 key points for 2D and 3D spaces. The downside of this dataset is that images are synthetically generated and have distorted edges of hands. All of these are recorded from one viewpoint [10].

SynthHands is a synthetic dataset, which provides information about 63,530 frames recorded from 5 views. Learning examples contain both RGB and depth records and represent records with and without object interaction. Data annotated for 21 points in

3D space. Hands were generated using Unity3D engine but animated using data captured from real motion [11].

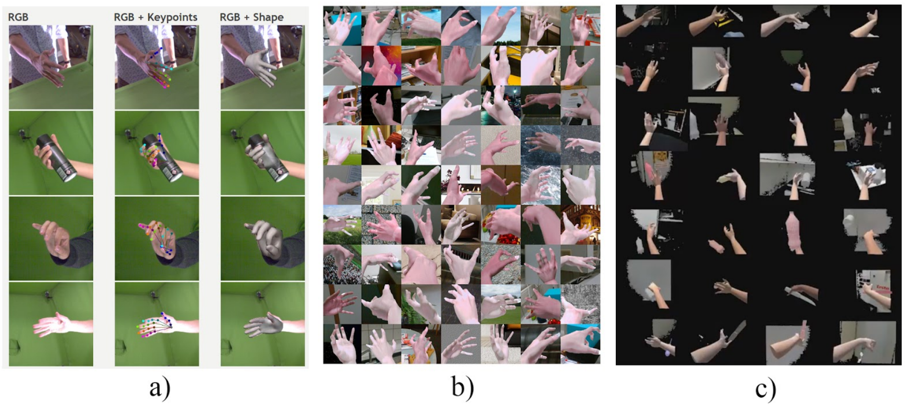

Fig. 2. Example of pictures in datasets: a) FreiHAND, b) GANerated Hands, c) SynthHands

3 Thesis Statement

There is a problem of the lack of a method for accurate 3D reconstruction of 2D sign language video content. We aim to solve it by introducing the neural network to the 3D reconstruction pipeline trained on multi-view dataset.

Statement: usage of neural network trained to make projections onto several planes improves the quality of sign language 3D reconstruction from video sequence.

4 Tentative Outline of the Thesis

4.1 Methods Overview

We plan to use the schema of 3D hand pose estimation specified in Fig 2. To improve the performance of sign language 3D reconstruction, we are going to test various methods for calculating intermediate results such as 2D and 3D points. We also plan to capture new dataset to improve the accuracy of sign language 3D reconstruction.

We are considering several approaches to address the problem. We propose two ways of how to redesign the second and third stages of the computational pipeline (Fig. 2). The first solution is to introduce the pair of networks, which will estimate the key points in 2D and 3D. As the second method, we propose to calculate not points but transformations of points in space with a pair of CNNs. Both methods use concatenated information about previous and current frames as input, namely the location of 2D points of last frame and RGB data for two frames. Therefore, the depth of the input is 27 (21+3+3).

The first method. We are going to take as a basis the first method [9] described in section 2.2. The pre-trained neural network YOLO will be used to calculate hand localization. We will introduce two networks A and B to compute locations of 2D points and estimate hand marks in three-dimensional space respectively. The usage of the two connected networks makes it possible to use skip connection and increase the complexity of extracted patterns in the third stage by creating connections between hidden layers of the two networks.

Let *kc* denote the convolutional layer with *k* filters and stride 1, *kd* - the convolutional layer with *k* filters and stride 2, *kr* - the residual block with *k* filters, *ku* - the transposed convolutional layer with *k* filters and stride 2, *kfc* - fully connected layer with *k* neurons. Relu is an activation function on all layers, except the last one with sigmoid activation. All layers have kernel size of 3 and padding of zeros with size 1. Then an architecture of the network A is: *64c, 128d, 256d, 256c, 256r, 256r, 256r, 256r, 256c, 256u, 128u, 64c, 21c*; and architecture of the network B is: *64c, 128d, 256d, 256c, 256r, 256r, 256r, 256d, 128c, 64c, 32c, 256fc, 256fc, 21*3fc*. The outputs from fourth, fifth, and sixth layers of network A concatenated with the correspondent outputs of network B. Skip connections allow the second network to use encoded information about RGB image in the process of 3D points estimation. The Adam optimizer will be used to minimize the difference between the labeled and predicted 2D key points for network A, as well as to minimize the difference between projection of predicted locations of 3D points and known locations of their projections into different views for the network B (see Fig. 3).

The second method. The second method is a modified version of the first one. We are changing the architecture of networks A and B to approximate the transformation matrices of points between frames and not the entire 3D model. The architecture described below calculates 21 transformation matrices. For most frames, fever matrices can be used to describe hand motion. We plan to train another CNN to handle these cases.

Let *kc* denote the convolutional layer with *k* filters and stride *1, kd* - the convolutional layer with *k* filters and stride *2, kr* - the residual block with k filters, ku the transposed convolutional layer with *k* filters and stride 2. Relu is an activation function on all layers, except the last one with sigmoid activation and set of transposed convolutions between networks with leaky relu activation. All layers have kernel size of 3 and padding of zeros with size 1. Then an architecture of the network *A* is: *64c, 128d, 256d, 256c, 256r, 256r, 256r, 256r, 256d, 256d, 128d, 64d, 21c; and architecture of the network B* is: *64c, 128d, 256d, 256c, 256r, 256r, 256r, 256d, 128d, 64d, 32d, 21c.* The outputs from fourth, fifth, and sixth layers of network A concatenated with the correspondent outputs of network B. To concatenate input and output of the network A we are going to use 6 transposed convolutions with stride *2*, leaky relu activations and following number of channels: *32, 32, 16, 8, 4, 2* (see Fig. 4).

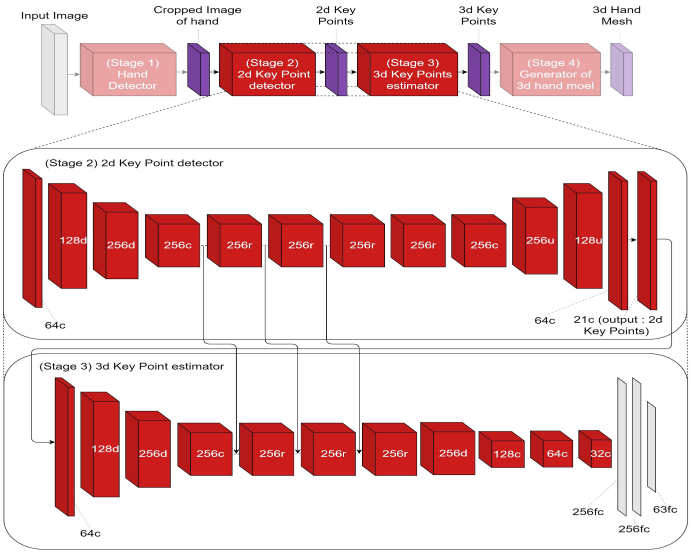

Fig. 3. First method scheme

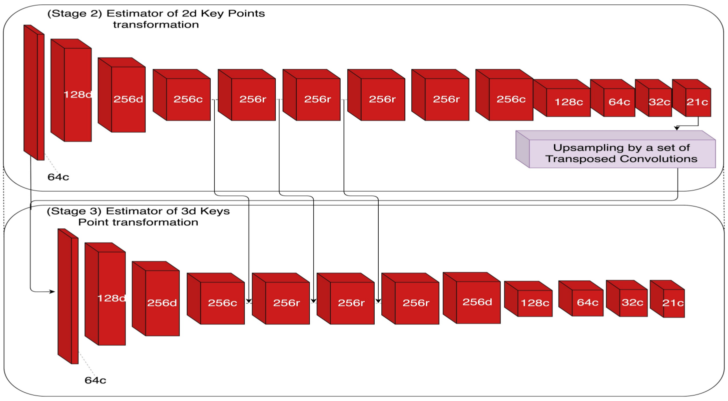
(Stage 2) Estimator of 2d Key Points transformation
128d
256d
256c
256r
256r
256r
256r
256c
128c
64c
32c
21c
Upsampling by a set of Transposed Convolutions
64c
(Stage 3) Estimator of 3d Keys Point transformation
128d
256d
256c
256r
256r
256r
256d
128c
64c
32c
21c
64c

Fig. 4. Second method scheme

4.2 Dataset Creation

We are going to record a video with hand movements similar to sign language gestures from at least three cameras. We expect to improve reconstruction accuracy by training the networks on this dataset.

4.3 Experiments and Evaluation

We will train the network on FreiHAND, GANerated Hands, and SynthHands datasets, and then fine-tune on the introduced sign language dictionary dataset. The proposed methods will be evaluated on the STB and RHD datasets. Since the accuracy of sign language dictionary reconstruction could not be completely evaluated with error metric, we plan to engage sign language experts to evaluate the result.

5 Timeline to Completion

October 2019 - Create sign language dataset. Implement and evaluate method one. Describe results.

November 2019 - Implement and evaluate method two. Describe results.

December 2019 - Compare results to the State-of-the-Art methods for 3D reconstruction, formulate conclusions.

January 2020 - Make final edits. Defend the thesis.

6 Conclusion

We propose the methods aimed at solving the task of 3D reconstruction from video sequences. We plan to compare the performance of multiple architectures and describe the data pre-processing pipeline. The work is not only aimed at investigation of sign language reconstruction problems but also at the preparation of the baseline algorithm for future VR and AR products.

References

1. Tompson, J., Stein, M., Lecun, Y.. Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans on Graphics 33(5), Article No. 169 (2014)
2. Tang, D., Jin Chang, H., Tejani, A., Kim, T. K.: Latent regression forest: structured estimation of 3d articulated hand posture. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3786–3793. IEEE Press, New York (2014)
3. Sun, X., Wei, Y., Liang, S., Tang, X. and Sun, J.: Cascaded hand pose regression. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition, pp. 824–832. IEEE Press, New York (2015)
4. Malik, J., Elhayek, A., Nunnari, F., Varanasi, K., Tamaddon, K., Heloir, A., Stricker, D.: Deephps: end-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In: 2018 IEEE International Conference on 3D Vision, pp. 110119. IEEE Press, New York (2018)
5. Athitsos, V., Sclaroff, S.: Estimating 3D hand pose from a cluttered image. In: 2003 IEEE Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. Vol. 2, pp. II-432. IEEE Press, New York (2003)
6. Chen, X., Wang, G., Guo, H., Zhang, C.: Pose guided structured region ensemble network for cascaded hand pose estimation. Neurocomputing (2019). doi: 10.1016/j.neucom.2018.06.097
7. Marín-Jiménez, M.J., Romero-Ramirez, F.J., Muñoz-Salinas, R., Medina-Carnicer, R.: 3D human pose estimation from depth maps using a deep combination of poses. Journal of Visual Communication and Image Representation **55**, 627–639 (2018). doi: 10.1016/j.jvcir.2018.07.010

8. Ye, M., Wang, X., Yang, R., Ren, L., Pollefeys, M.: Accurate 3d pose estimation from a single depth image. In: 2011 IEEE International Conference on Computer Vision, pp. 731–738. IEEE Press, New York (2011)
9. Panteleris, P., Oikonomidis, I. Argyros, A.: Using a single RGB frame for real time 3D hand pose estimation in the wild. In: 2018 IEEE Winter Conference on Applications of Computer Vision, pp. 436–445. IEEE Press, New York (2018)
10. Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D. Theobalt, C.: GANerated hands for real-time 3d hand tracking from monocular RGB. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, pp. 49–59. IEEE Press, New York (2018)
11. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D. Theobalt, C.: Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: 2017 IEEE International Conference on Computer Vision, pp. 1284–1293. IEEE Press, New York (2017)
12. Zimmermann, C., Brox, T.: Learning to estimate 3D hand pose from single RGB images. In Proceedings of the IEEE International Conference on Computer Vision, pp. 4903-4911. IEEE Press, New York (2017)
13. Zhang, J., Jiao, J., Chen, M., Qu, L., Xu, X., Yang, Q.: 3D hand pose tracking and estimation using stereo matching. arXiv preprint arXiv:1610.07214 (2016)
14. Yuan, S., Ye, Q., Stenger, B., Jain, S., Kim, T.K.: BigHand2.2m benchmark: hand pose dataset and state of the art analysis. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2605-2613. IEEE Press, New York (2017)
15. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, pp. 409419. IEEE Press, New York (2018)
16. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D.: Accurate, robust, and flexible real-time hand tracking. In: 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 3633–3642). ACM (2015)
17. Wetzler, A., Slossberg, R., Kimmel, R.: Rule of thumb: deep derotation for improved fingertip detection. arXiv preprint arXiv:1507.05726 (2015)
18. Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., Gall, J.: Capturing hands in action using discriminative salient points and physics simulation. International Journal of Computer Vision **118**(2), 172–193 (2016). doi: 10.1007/s11263-016-0895-4
19. Rogez, G., Khademi, M., Supančič III, J.S., Montiel, J.M.M., Ramanan, D.: 3D hand pose detection in egocentric RGB-D images. In: Agapito, L., Bronstein, M., Rother, C. (eds.) ECCV 2014 Workshops. LNCS, vol. 8925, pp. 356–371. Springer, Cham (2014)
20. Gomez-Donoso, F., Orts-Escolano, S., Cazorla, M.: Large-scale multiview 3D hand pose dataset. arXiv preprint arXiv:1707.03742 (2017)
21. Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., Yuan, J.: 3D hand shape and pose estimation from a single RGB image. In: 2019 IEEE Conference on Computer Vision and Pattern Recognition, pp. 10833–10842. IEEE Press, New York (2019)