

Real-time Human Pose Estimation using RGB-D images and Deep Learning[☆]

림 빈 보 니 카¹
Beanbonyka Rim

성 낙 준¹
Nak-Jun Sung

마 준¹
Jun Ma

최 유 주²
Yoo-Joo Choi

홍 민^{3*}
Min Hong

ABSTRACT

Human Pose Estimation (HPE) which localizes the human body joints becomes a high potential for high-level applications in the field of computer vision. The main challenges of HPE in real-time are occlusion, illumination change and diversity of pose appearance. The single RGB image is fed into HPE framework in order to reduce the computation cost by using depth-independent device such as a common camera, webcam, or phone cam. However, HPE based on the single RGB is not able to solve the above challenges due to inherent characteristics of color or texture. On the other hand, depth information which is fed into HPE framework and detects the human body parts in 3D coordinates can be usefully used to solve the above challenges. However, the depth information-based HPE requires the depth-dependent device which has space constraint and is cost consuming. Especially, the result of depth information-based HPE is less reliable due to the requirement of pose initialization and less stabilization of frame tracking. Therefore, this paper proposes a new method of HPE which is robust in estimating self-occlusion. There are many human parts which can be occluded by other body parts. However, this paper focuses only on head self-occlusion. The new method is a combination of the RGB image-based HPE framework and the depth information-based HPE framework. We evaluated the performance of the proposed method by COCO Object Keypoint Similarity library. By taking an advantage of RGB image-based HPE method and depth information-based HPE method, our HPE method based on RGB-D achieved the mAP of 0.903 and mAR of 0.938. It proved that our method outperforms the RGB-based HPE and the depth-based HPE.

☞ keyword : Human pose estimation, human skeleton tracking, keypoint localization, deep learning

1. Introduction

Human Pose Estimation (HPE) which localizes the human body joints becomes a high potential for high-level applications in the field of computer vision. The 2D HPE framework detects and retrieves the 2D coordinates of human

body joints in a single image or frame. The main body joints of human which HPE framework commonly tracked are head, shoulders, elbows, wrists, hips, knees, and ankles. The skeleton is formed by pairing those body part joints. Diversity of these skeleton motion tracking allows us to recognize the various kinds of human activities such as walking, running, sitting, and more. The human activity recognition is useful for high level applications such as dynamic projection mapping [1], AR application [2] and more.

The main challenges of HPE in real-time are occlusion, illumination change and diversity of pose appearances. The single RGB image is fed into HPE framework in order to reduce the computation cost by using depth-independent devices such as a common camera, webcam, or phone cam. However, the single RGB image-based HPE is not able to solve the above challenges due to inherent characteristics of color or texture [3-5]. On the other hand, depth information which is fed into HPE framework and detects the human body parts in 3D coordinates is advantageous to solve the

¹ Department of Computer Science, Soonchunhyang University, Asan, 31538, Korea

² Department of Newmedia, Seoul Media Institute of Technology, Seoul, Korea

³ Department of Computer Software Engineering, Soonchunhyang University, Asan, 31538, Korea

* Corresponding author (mhong@sch.ac.kr)

[Received 6 February 2020, Reviewed 26 February 2020(R2 13 April 2020), Accepted 16 April 2020]

☆ This work has supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1F1A1062752), and was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF-2017R1D1A1B03035718) funded by the Ministry of Education.

☆ A preliminary version of this paper was presented at APIC-IST 2019.

challenge problems. However, the depth information-based HPE requires the depth-dependent device which has space constraint and is cost consuming. Especially, the result of depth information-based HPE is less reliable due to the requirement of pose initialization and less stabilization of frame tracking [6-9].

There are many human body parts which can be occluded by other parts. For example, most common RGB-based HPE is reliable to detect the head position at the front view, while it fails to detect at the back view. However, the depth-based HPE is reliable to detect head position in both views, but less reliable to detect other part joints. Therefore, this paper proposes a new HPE method which is robustly in estimating self-occlusion. However, this paper focuses only on head self-occlusion. Our new method combines a RGB-based HPE framework and the depth-based HPE framework. The method adapted the state-of-the-art RGB-based HPE framework called "Openpose [5]" to detects all human body parts in all views except head part. We also adapted the well-known depth-based HPE framework called "Kinect v2 [9] skeleton tracking system" to detect only head part in all views.

Our main contributions are (1) to study a possibility on the combination of RGB-based HPE framework and depth-based HPE framework; and (2) to develop a robust HPE framework based on RGB-D to solve the problem of head self-occlusion in all views.

2. Related Works

Recently, Human Pose Estimation is conducted in two approaches, bottom-up and top-down approach. The bottom-up approach is a study on firstly identifying the location of joints in the image. Then, the post-processing of part association is conducted. This approach reaches state-of-the-art level by using deep learning method on single RGB image. Convolutional Pose Machine (CPM) [4] introduced a sequential prediction architecture using convolutional neural network (CNN) and stacked hourglass network. CPM [4] showed the state-of-the-art result on single human detection in MPII Human Pose dataset [10]. Nie et al. [11] introduced a RGB-based HPE by defining a pyramid stacking and dilation module to provide better heatmap prediction of body joints in multiple scales. Openpose [5]

which extended from CPM [4] introduced the first 2D single-RGB-based HPE for multiple people in real-time. Openpose [5] detects up to 19 people in a video with the speed of 8.8 fps. The bottom-up approach relies on the part association step in which every keypoint is linked as a skeleton. For pose estimating on multiple people, the failure of part association is commonly occurred by linking the wrong joint from one person to another person.

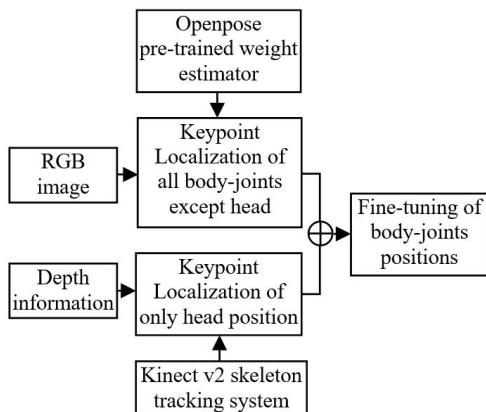
On other hand, the top-down approach is a study on firstly detecting the bounding box of human within the image. Then, the keypoints of human joints are localized within the bounding box. Park et al. [12] introduced 2D RGB-D-based HPE by using two CCTV cameras to detect the moving human and then combined with depth information to segment the moving human from the background. Finally, Park et al. [12] adapted CPM [4] HPE architecture to localize the keypoints. The Microsoft team introduced Kinect v2 [9] skeleton tracking system using an infra-red sensor to capture depth information. Based on top-down approach, Kinect v2 [9] system segments the moving human from the background. Then, it localizes the keypoint coordinates inside the moving human region. The top-down approach relies on the bounding box segmentation. If the bounding box segmentation fails to remove the human from the background, the HPE estimator could not localize any keypoints.

To solve the problem of self-occlusion, many approaches have proposed a 3D human pose estimation. Tang et al. [13] stated that a single color image yielded a less accuracy, high ambiguity and slow to estimate human pose in 3D coordinates. Therefore, Tang et al. [13] proposed a method using Resnet50 as a feature map and Faster-RCNN as an object detector to estimate on a color image to receive a 2D human pose and map the 2D joint coordinates to the corresponding depth image to receive 3D joint coordinates. The color and depth image were captured by Kinect SDK. Hong et al. [14] proposed a dynamic pose estimation using two RGB-D Kinect devices to estimate head, hands and feet. The two Kinect devices yielded more reliable on mapping coordinates between color and depth image. The process of these two devices are: mapping data into a single coordinate system, pose estimation using voting scheme and body orientation using Principal Component Analysis. This method

received an average accuracy of 87%. Chun et al. [15] proposed 3D human pose estimation from RGB-D images of two CCTV cameras. The two CCTV cameras generated the depth information which is used to segment the occluded objects. Then, the 2D joint information was estimated by Convolutional Pose Machine (CPM) which was trained on MPII Human Pose dataset. The 3D joint information was trained by the 2D joint information using Deep Neural Network on Human3.6M dataset.

3. Methodology

By our empirical study, we observed that Kinect v2 [9] skeleton tracking system estimates the head position more robustly in all views, comparing to Openpose [5]. On the other hand, Openpose estimates all body-joints positions, except head position, more robustly than Kinect v2. Through this study, we propose a method of combining Kinect v2 and Openpose.

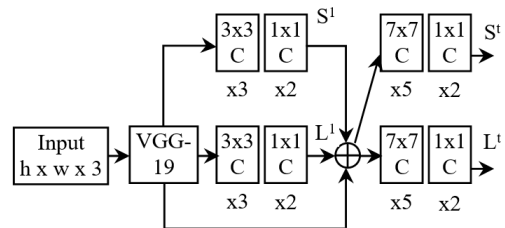


(Figure 1) An overall workflow of our method

(Figure 1) illustrates an overall workflow of our method. Firstly, we adapt Openpose HPE framework to obtain the confident map (all body joints positions except head position) and part association heatmap scores (all joint-pairs except a pair of head to neck position) using a pre-trained weight estimator in RGB image. Simultaneously, we also adapt Kinect v2 skeleton tracking system to obtain only the head position using Kinect v2 sensor in depth information. Finally,

we do a fine-tuning of body-joints positions by retrieving the confident map and part association heatmap scores of all body-joints except the head position from Openpose and the head position in 2D coordinates from Kinect v2. Then, all of them are combined to form a whole human body skeleton.

3.1 All body-joints localization except head position



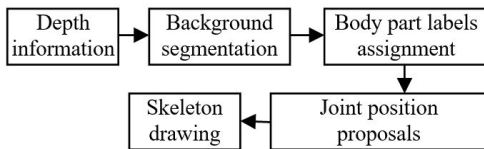
(Figure 2) An architecture of Openpose [5]

(Figure 2) illustrates an architecture of Openpose [5] which estimates the body-joints from an RGB image. The model works as the following: firstly, it extracts the feature map from the input image using 10 layers of VGG-19 model. Then, in the first stage, the feature map is fed into the top layers of the model in which outputs an array of scores of confident maps ($S1$) of all body-joints. The feature map is also fed into the bottom layers of the model in which outputs an array of score of part association heatmaps ($L1$) for each joint-pair. In the second step and on, the feature map, the confident maps score ($S1$) and part association heatmap score ($L1$) are cascaded and fed into next-step to achieve the final confident maps score (St) and part association heatmap score (Lt). The confident map scores are the body-joint position such as shoulders, elbows, ankles and more. The part association heatmaps scores are the joint-pair containing the probability score of each joint-pair such as a pairing between right shoulder and right elbow, and so on.

In our paper, we exploited a pre-trained weight of Openpose [5] which was trained on COCO 2014 keypoints dataset [16] and achieved the AP of 60.5% on COCO 2016 keypoint test challenge, to estimate the body-joints positions except head position and draw a skeleton. Firstly, the input image is captured by a camera of Kinect sensor with a resolution of 1920x1080x3 (height x width x channel). The

input image consists of human and background. However, we do not segment for the region of interest (ROI). The whole image was been used and scaled into 368x368, then fed into the pose estimator. The pose estimator generates 17 body-joints in 2D coordinates and 16 joint-pairs association. The 17 joints are: nose, eyes, ears, shoulders, elbows, wrists, hips, knees and ankles.

3.2 Head position localization

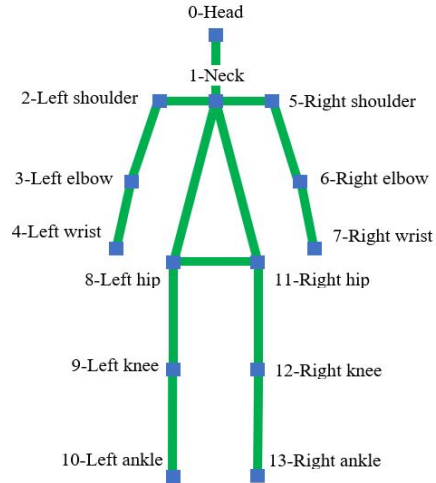


(Figure 3) An architecture of Kinect v2 [9] skeleton tracking system

(Figure 3) illustrates an architecture of Kinect v2 skeleton tracking system which estimates the body-joints positions from depth information. The Kinect skeleton tracking system works as the following: firstly, Kinect v2 system segments the moving-human from the background. Then, a single input depth image of human body with the size of 512x424 is clustered as a per-pixel classification task into a 31-body-part labeling using a deep randomized decision forest classifier. The 31-body parts are defined to be spatially localized near skeleton joints of interest. The classifier was trained on 3 trees, 20 deep, 300k images per tree, 2k pixels per image, 2k candidate features and 50 candidate thresholds per feature. Next, the positions of 3D skeleton joints are proposed by accumulating the global 3D centers of probability mass for each part. This probability is calculated using mean shift with a weighted Gaussian kernel. Finally, the skeleton is drawn using kinematic and temporal constraints.

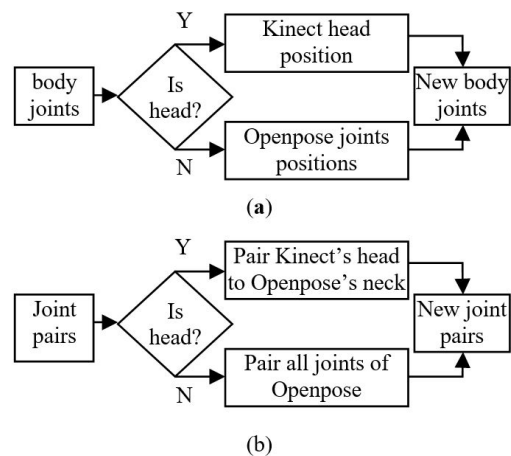
In our paper, we exploited a skeleton tracking system SDK which shipped with Kinect v2 device. Firstly, the input depth image with size of 512x424 is captured by an infra-red sensor of Kinect v2. The depth image is used to estimate the head position. Then, the head position is mapped into the resolution of 1920x1080. The mapping is conducted in order to fit with the RGB image which will be used to match with the skeleton estimated by Openpose.

3.3 Fine-tuning of all body-joints localization



(Figure 4) Our 14 body-joints proposal

The Openpose is able to detect 17 body-joints, while Kinect v2 can detect up to 25 body-joints. Therefore, we defined new 14 body-joints which are matched by Openpose’s and Kinect’s joints. The 14 joints are head, neck, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle and right ankle, as illustrated in (Figure 4).



(Figure 5) Fine-tuning architecture: (a) new body-joints generation and (b) new joints pairs association

To form a skeleton tracking, two information are acquired: joint location and joint pair association. (Figure 5) illustrates our method of fine-tuning in order to form a whole human skeleton composing of 14 body-joints. The joint position localization is acquired by conditional decision, as shown in (Figure 5a). For every body-joint, if the joint type is a head, the head position is estimated by Kinect skeleton tracking system. Kinect tracking system which estimates the head position was described in section 3.2. Otherwise, the other 13-joints positions are predicted by Openpose estimator. The Openpose estimator predicts the position of the joints using confident map scores, as described in section 3.1. The joint pair association is also acquired by a conditional decision, as shown in (Figure 5b). For every joint pair, if the pair association is between a head to neck, the joint pair between Kinect's head to Openpose's neck is associated. Otherwise, all joint pairs are accumulated by Openpose part association heatmaps scores, as described in section 3.1.

4. Experimental Result

4.1 Experimental environment

Our experiment has been conducted on a desktop of Windows 10, 64-bit OS, Intel® Core™i7-7700 CPU @3.60GHZ, RAM 16.0GB and Kinect v2 sensor. The Kinect v2 sensor is used for both capturing RGB image streaming of 1920x1080 resolution and depth information streaming of 512 x 424 resolution. The user performs any activities between 1.5m to 4.5m from Kinect sensor.

The experiment for a single-person HPE was conducted within indoor laboratory environment in various activities such as walking, running, sitting, dancing, leaving the sensor and going back, and more.

4.2 Evaluation method

We evaluate our method performance using COCO Object Keypoint Similarity (OKS) [16] library. The OKS formula is as the shown in Eq. (1).

$$OKS = \sum_i [\exp(-d_i/2s^2\kappa_i^2)\delta(v_i>0)] / \sum_i [\delta(v_i>0)] \quad (1)$$

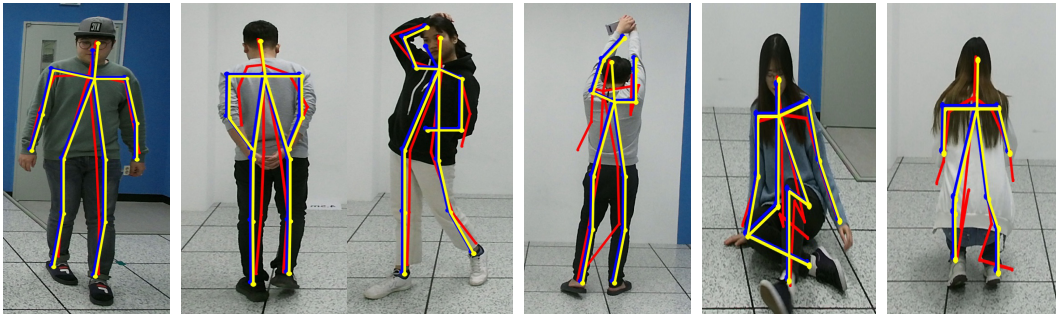
where d_i is the Euclidean distance between each detected keypoint to the corresponding ground truth keypoint. s means an object size. If the area of object is between 32x32 and 96x96, it is considered as a medium size. If the area of object is larger than 96x96, it is considered as a large size. κ_i is the keypoint constants defining the distance ratio of all keypoints. It means that the distance between detected right wrist location and the ground truth right wrist location should be smaller than the distance between detected right hip location and the ground truth right hip location. The COCO OKS defined κ_i value to all part joints as following: $\kappa_i=2\sigma_i$, where $\sigma = [0.026, 0.025, 0.035, 0.079, 0.072, 0.062, 0.107, 0.087, 0.089]$ which are corresponding to nose, eyes, ears, shoulders, elbows, wrists, hips, knees, and ankles, respectively. Due to our method keypoints contains only 14 joints, we modified the σ value as the following: $\sigma = [0.062, 0.079, 0.079, 0.072, 0.062, 0.107, 0.087, 0.089]$ which are corresponding to head, neck, shoulders, elbows, wrists, hips, knees and ankles, respectively. Due to the portion of head distance to be well considered is similar to the portion of wrists, we give the same σ value of wrists to head. Similarly, we give σ value of shoulders to neck. v_i is the visibility flag of each keypoint where $v_i = 0$ when keypoint does not present in the image, $v_i = 1$ when keypoint presents in the image but it is occluded, and $v_i = 2$ when keypoint presents in the image and it is well visible.

4.3 Result and discussion

Our method was tested by 10 users. They conduct activities differently for around 2 minutes. We trimmed 200 frames per user to evaluate the performance. We manually generated the ground truth of those 200 frames following the format of COCO ground truth [16]. Totally, for 10 users, we annotated the ground truth of 2,000 frames.

(Figure 6) visualizes the experimental results. The results of Kinect v2 skeleton tracking system are presented in red color. The results of Openpose and the proposed method are presented blue and yellow, respectively.

The COCO OKS generates the performance of detected keypoints in 10 matrices but we are focus on only 2 main criterion matrices: (1) is average precision (AP) of OKS threshold between 0.50 to 0.95 and (2) is average recall



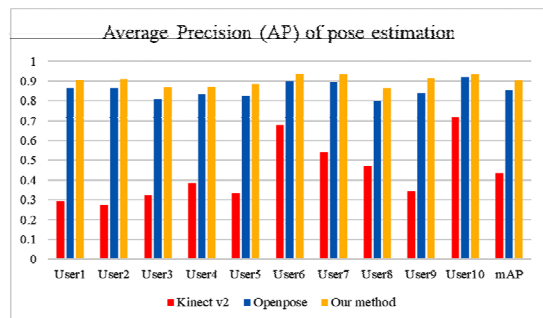
(Figure 6) Experimental results of Kinect system (red), Openpose (blue) and our method (yellow)

(AR) of OKS threshold between 0.50 to 0.95 for evaluation and comparison with Kinect v2 system and Openpose.

(Table 1) and (Figure 7) depict the results of pose estimation for 10 users in average precision (AP). The Kinect v2 system performed with low accuracy, with AP values ranging from 0.274 to 0.716. The Kinect v2 system failed to detect when user does a rare pose appearance or moves fastly. The Kinect v2 system estimated the pose reliably when user performs a common activity or moves slowly. However, Kinect v2 system detected occlusion robustly in head detection even from the back view. On other hand, Openpose was reliably performed with AP values ranging from 0.801 and 0.920. We notice that Openpose reliably performs all keypoints except the head position when there is the occlusion, especially the head detection at the back view. Our method was performed with the AP values ranging from 0.863 and 0.938. If Kinect tracking system is successfully localized, our method detects robustly head self-occlusion even though Openpose fails to detect the head.

(Table 1) Average Precision (AP) of pose estimation

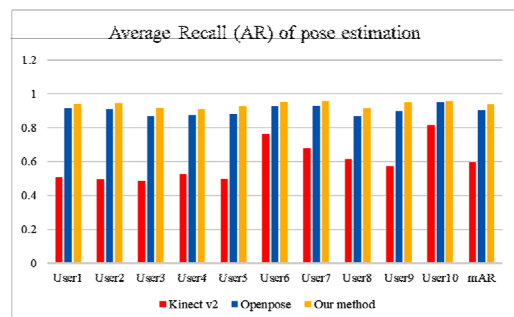
User	Kinect v2 [9]	Openpose [5]	Our method
User1	0.293	0.864	0.906
User2	0.274	0.863	0.911
User3	0.320	0.807	0.868
User4	0.384	0.832	0.871
User5	0.333	0.827	0.885
User6	0.677	0.899	0.938
User7	0.539	0.897	0.936
User8	0.473	0.801	0.863
User9	0.344	0.839	0.914
User10	0.716	0.920	0.935
mAP	0.435	0.855	0.903



(Figure 7) Average Precision (AP) of pose estimation

(Table 2) Average Recall (AR) of pose estimation

User	Kinect v2 [9]	Openpose [5]	Our method
User1	0.504	0.913	0.940
User2	0.493	0.908	0.944
User3	0.487	0.867	0.917
User4	0.525	0.876	0.910
User5	0.499	0.881	0.927
User6	0.765	0.926	0.955
User7	0.679	0.930	0.958
User8	0.615	0.866	0.915
User9	0.571	0.898	0.951
User10	0.814	0.948	0.958
mAR	0.595	0.901	0.938



(Figure 8) Average Recall (AR) of pose estimation

Average Recall (AR) of each method was measured as shown in (Table 2) and (Figure 8). Average Recall (AR) is a criteria to measure on how good our true positive prediction over all true positive of ground truth. We notice that our method was performed with higher AR than Kinect v2 system and Openpose. In both mean AP(mAP) and mean AR(mAR), our method outperforms Kinect V2 system and Openpose with mAP of 0.903 and mAR of 0.938.

5. Conclusion

In this paper, we proposed new HPE method to robustly localize the human pose and track the human skeleton in real-time. We proved that the proposed method estimates the pose with high accuracy even though the head self-occlusion has happened. By taking an advantage of RGB-based HPE method and depth-based HPE method, our RGB-D-based HPE method achieved the mAP of 0.903 and mAR of 0.938. Our method outperforms the RGB-based HPE and the depth-based HPE in both AP and AR. If Kinect tracking system is successfully localized, our method detects robustly head self-occlusion even though Openpose fails to detect the head.

As future work, we should improve the execution speed of the proposed method. Since our RGB-D-based HPE framework combines two HPE frameworks and we use Kinect v2 sensor for two functions, RGB image streaming and depth information capturing, we have achieved the speed of localizing and tracking less than 10fps for a video. Additionally, we still have challenges on multiple-person pose estimation and other part joints self-occlusion. Those are our interesting future works.

Reference

- [1] S. Kim and Y. Choi, "Design of Authoring Tool for Dynamic Projection Mapping onto Multiple Bodies", *Proceeding of International Conference on Internet, ICONI*, pp. 197-199, 2018.
<https://design486.tistory.com/486>
- [2] S. K. Kim, S. Kang, Y. Choi, M. Choi and M. Hong, "Augmented-Reality Survey: from Concept to Application," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 2, pp. 982-1004, 2017.
<https://doi.org/10.3837/tiis.2017.02.019>.
- [3] V. Ramakrishna, D. Munoz, M. Hebert, J. Bagnell and Y. Sheikh, "Pose machines: Articulated pose estimation via inference machines." *Proceeding of European Conference on Computer Vision*, pp. 33-47, Springer, Cham, 2014.
https://doi.org/10.1007/978-3-319-10605-2_3
- [4] S. Wei, V. Ramakrishna, T. Kanade and Y. Sheikh, "Convolutional Pose Machines", *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724-4732, 2016.
https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Wei_Convolutional_Pose_Machines_CVPR_2016_paper.html
- [5] Z. Cao, T. Simon, S. Wei and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291-7299, 2017.
http://openaccess.thecvf.com/content_cvpr_2017/html/Cao_Realtime_Multi-Person_2D_CVPR_2017_paper.html
- [6] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman and A. Blake, "Efficient Human Pose Estimation from Single Depth Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 12, pp.2821-2240, 2013.
<https://doi.org/10.1109/TPAMI.2012.241>
- [7] Y. Zhu, B. Dariush and K. Fujimura, "Controlled human pose estimation from depth image streams", *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1-8, Anchorage, AK, 2008.
<https://doi.org/10.1109/CVPRW.2008.4563163>
- [8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, "Real-Time Human Pose Recognition in Parts from Single Depth Images", *Proceeding of Computer Vision and Pattern Recognition Conference*, pp. 1297-1304, Providence, RI, 2011.
<https://doi.org/10.1109/CVPR.2011.5995316>
- [9] P. Kohli and J. Shotton, "Key Developments in Human

- Pose Estimation for Kinect.” Consumer Depth Cameras for Computer Vision, pp.63-70, Springer, London, 2013. https://doi.org/10.1007/978-1-4471-4640-7_4
- [10] M. Andriluka, L. Pishchulin, P. Gehler and B. Schiele, “2D Human Pose Estimation: New Benchmark and State of the Art Analysis”, Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3686-3693, Columbus, OH, 2014. <https://doi.org/10.1109/CVPR.2014.471>
- [11] Y. Nie, J. Lee, S. Yoon and D. S. Park, “A Multi-Stage Convolution Machine with Scaling and Dilation for Human Pose Estimation,” KSII Transactions on Internet and Information Systems, vol. 13, no. 6, pp. 3182-3198, 2019. <https://doi.org/10.3837/tiis.2019.06.023>.
- [12] S. Park, M. Ji and J. Chun, “2D Human Pose Estimation based on Object Detection using RGB-D information,” KSII Transactions on Internet and Information Systems, vol. 12, no. 2, pp. 800-816, 2018. <https://doi.org/10.3837/tiis.2018.02.015>.
- [13] H. Tang, Q. Wang and H. Chen, “Research on 3D Human Pose Estimation Using RGBD Camera,” 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 538-541, Beijing, China, 2019. <https://doi.org/10.1109/ICEIEC.2019.8784591>
- [14] S. Hong and Y. Kim, “Dynamic Pose Estimation Using Multiple RGB-D Cameras”, Sensors, vol. 18(11), no. 3865, 2018. <https://doi.org/10.3390/s18113865>
- [15] J. Chun, S. Park and M. Ji, “3D Human Pose Estimation from RGB-D Images Using Deep Learning Method”, SSIP 2018: Proceedings of the 2018 International Conference on Sensors, Signal and Image, Association of Computing Machinery, pp. 51-55, New York, USA, 2018. <https://doi.org/10.1145/3290589.3290591>
- [16] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar and C. Zitnick, “Microsoft COCO: Common Objects in Context.” Proceeding of European conference on computer vision, pp. 740-755. Springer, Cham, 2014. https://doi.org/10.1007/978-3-319-10602-1_48

● 저 자 소 개 ●



림 빈 보 니 카 (Beanbonyka Rim)

2007: Royal University of Phnom Penh (Cambodia), Computer Science (Bachelor degree)

2010: GroupT - KU Leuven University (Belgium), E-Media (Master degree)

2018 ~ Present: Soonchunhyang University (Korea), Computer Science (PhD degree)

Research Interests : Deep Learning, Image Processing, Computer Vision, Computer Graphics, Augmented Reality

E-mail : rim.beanbonyka@sch.ac.kr



성 낙 준 (Nak-jun Sung)

2016: Soonchunhyang University (Korea), Computer Software Engineering (Bachelor degree)

2018: Soonchunhyang University (Korea), Computer Science (Master degree)

2018 ~ Present: Soonchunhyang University (Korea), Computer Science (PhD degree)

Research Interests: Dynamic Simulation, Physical-based Simulation, Deep Learning, Augmented Reality, Virtual Reality

E-mail: njsung@sch.ac.kr

● 저 자 소 개 ●



마 준 (Jun Ma)

2018: Chungnam National University (Korea), Computer Science (Bachelor degree)
2018 ~ Present: Soonchunhyang University (Korea), Computer Science (Master degree)
Research Interests: Augmented Reality, Virtual Reality, Convergence Rehabilitation
E-mail: majun@sch.ac.kr



최 유 주 (Yoo-Joo Choi)

1991: Ewha Womans University (Korea), Computer Science (Master degree)
2005: Ewha Womans University (Korea), Computer Science (PhD degree)
2006 ~ Present: Professor at Department of Newmedia, Seoul Media Institute of Technology (SMIT), Korea
Research Interests: Image Processing, Computer Vision, Computer Graphics, Augmented Reality, Human-computer Interaction
E-mail: yjchoi@smit.ac.kr



홍 민 (Min Hong)

1995: Soonchunhyang University (Korea), Computer Science (Bachelor degree)
2001: University of Colorado at Boulder (USA), Computer Science (Master degree)
2005: University of Colorado at HSC (USA), Bioinformatics (PhD degree)
2006 ~ Present: Professor at Department of Computer Software Engineering, Soonchunhyang University, Korea
Research Interests: Computer Graphics, Dynamic Simulation, Bioinformatics, Image Processing
E-mail: mhong@sch.ac.kr