International Journal Of Electrical, Electronics And Data Communication, ISSN(p): 2320-2084, ISSN(e): 2321-2950
Volume-5, Issue-12, Dec.-2017, http://iraj.in

3-D HAND POSE RECOGNITION FROM A PAIR OF DEPTH AND GEODESIC IMAGES USING DEEP CONVOLUTIONAL NEURAL NETWORK

[1]J. M. PARK, [2]G. GI, [3]T. Y. KIM, [4]H. M. PARK, [5]T.-S.KIM

Kyung Hee University, Republic of Korea
E-mail: {jmpark, geon, kty, hmp9669, tskim}@khu.ac.kr

Abstract - Accurate 3-D hand-pose recognition is one of thenovel user interface technologies that can facilitate interactions between humans and smart devices. In this work, we propose a methodology to recognize 3-D hand pose from a pair of depth and geodesic distance images of a hand. First, we train a Convolutional Neural Network (CNN) regressorwith a database of depth and Geodesic images of a hand along with the ground-truth joint positions.Second, using the trained CNN regressor, we estimate the 3-D joint positions from input pairs of depth and Geodesic images. Finally, based on the estimated joint positions, we reconstruct 3-D hand poses. Our results show that making use of Geodesic distancealongwith depthinformationimproves 3-D hand pose recognition by enhancing the capacity of regression via CNN

Index Terms - Depth image, Geodesic image, 3-D hand pose recognition, Deep learning, Convolutional neural network

INTRODUCTION

The user interface environment is moving away from a device-oriented system to a human-centric smart environment. Research works on user interface technology are actively under way to develop and provide such smart environment. Especially, hand-based gesture interfaces are actively under developed, since they provide natural, intuitive, and convenient interactions without additional input devices. There are two main methodological categories for hand gesture recognition: one is sensor-based method and the other is image-based method. The sensor-based recognition method has an advantage in that it can measure precise movementsviabody attached sensors, but unfortunately, it is inconvenient to wearthe sensors. However, the image-based recognition methodology is relatively simple in settings and has no additional attachments to the user. Also it can be used relatively freely, but field-of-view becomes a limitation.

There is a good review of earlier hand pose estimation works bySupancicet al. based on images [4]. Image-basedhand pose estimation methodologies can be divided into two categories. The first involves model-based tracking methods via unsupervised techniques. For instance, Oikonomidis et al. [6] useda Particle Swarm Optimization (PSO) method on multi-camera images.Qian et al. [11] used a 3-D hand model and Iterated Closest Point (ICP) method with PSO.Sharp et al. [1] and Taylor et al. [5] proposed regression methodologies with Golden energy to minimize distance of each points. Although those attempts tried to solve the hand tracking problem, such model-based tracking methods cannot handle drastic changes of hand motion well. The second is based on the estimation of joint positions or angles from RGB or RGB-Depth images. Most of these approaches use supervised learning-based methods. Generally, machine learning techniques are utilized to predict hand joint information. For example, Keskin et al. [7] and Tang et al. [12] used Random Decision Forest to estimate joint position. Both studies performed pixel classification and assigned each pixel to a hand part (i.e. hand parts recognition). However, actual 3-D hand joint positions are not directly predicted, but only classified hand parts from which hand joint positions are identified.

Recently, deep learning algorithm are adopted to predict hand joint positions directly. For instance, Convolutional Neural network (CNN) has been employed for 3-D hand pose estimation. Zhou et al. [10] used CNN to predict joint angle parameters of a hand model. Tompson et al. [13] produced a heat map image from CNN to predict 2D joint positions. Oberweger et al. [8] used CNN with multi-scale and multi-stage to regress 3-D joint positions with pose priors. Oberweger et al. [16] also used a CNN model to generate depth image and then update pose. Ge et al. [14] proposed to project a single depth image onto 3 orthogonal planes, then used them as an input of a CNN model to predict 3-D joint positions. Guo et al. [17] presented two-stream CNN. Their model used a single depth image and the corresponding edge image generated by Random Forest to improve fingertip estimation. Although their approach estimated only fingertips, it improved performance by employing additional information from the preprocessed images.

In this work, we propose a CNN-based 3-D hand pose estimation methodology using a set of hand depth and additional Geodesic distance images. Although 3-D hand joint positions can be predicted using only depth images, it may not be enough to predict joint positions accurately. To improve prediction, we propose to use Geodesic distance information along with depth as an input to CNN. Our results show that the additional Geodesic information along with depth improves the accuracy of 3-D hand pose recognition.

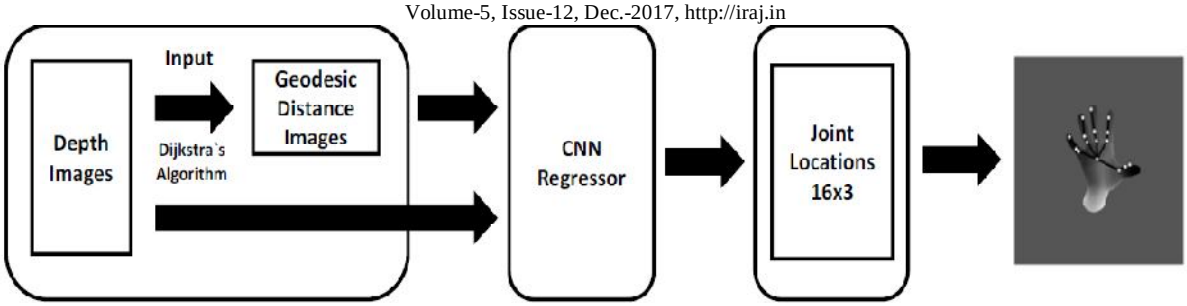

Figure 1. Overview of our proposed 3-D hand pose recognition system. A Geodesic distance image is calculated by Dijkstra's algorithm from a depth image. A trained CNN regressor produces hand joint positions (i.e., 16joints and their 3-D positions.) Finally, a 3-D hand pose gets reconstructed by connecting these joints.

II. METHODS

Fig. 1 shows the entire processes of our proposed system. First, the depth andgeodesic distance imagesare used as input to train and test our CNN regressor. The output of the CNN regressoris a matrix ofjoints positions for sixteen joints and their pixel coordinates positions(i.e, 16x3). The final 3-D pose of a hand gets reconstructed from these predicted joints positions.

A.Depth Hand Images

A Region of Interest(ROI) is extracted from a depth image based on the centroid of the hand and resized to a size of 128x128. This extracted ROI is normalized to a range of -1 to 1. The depth values reflect the distance between the camera and the hand. The normalization of the depth image ensures that hand is not affected by the distance between the hand and the camera during learning.

In this work, we utilize the NYU hand dataset[13] for training and validation. The NYU dataset is composed of 72k depth images for training and 8k depth images for testing. We used 40k images of the NYU hand dataset as a set of training data and randomly picked 6k images as a set of testing data.We used sixteen joints as the ground truth annotations without the wrist joint.

B. Geodesic Distance Images

The Geodesic distance is the shortest distance from the centroid of a hand to specific point that lies on the surface of ahand. The Geodesic distance is calculated by Dijkstra's algorithm which is a kind of greedy algorithm for finding the shortest path between nodes in a graph.

From the extracted ROI of depth image, we construct a graph $G_t = (Z_t, E_t)$, with 3-D points Z_t as vertices and edges E_t. Edges are created between a vertex and its 8-neighbors in a 2D depth image and each weight of edge is Euclidean distance between two points. Then we measure geodesic distance between different hand pointsusingG_t.

$$d_G(x, y) = \sum_{e \in SP(x,y)} w(e) \qquad (1)$$

There are all edges along the shortest path between x and y where $d_G(x, y)$ is the geodesic distance between two points x and y [3]. We calculate the shortest path to all other points from the centroid of the hand. This algorithm has a runtime complexity of $O(|Z_t| \cdot \log|Z_t|)$. Fig. 2 shows an example of Geodesic image and a Geodesic distance is indicated by an arrow.

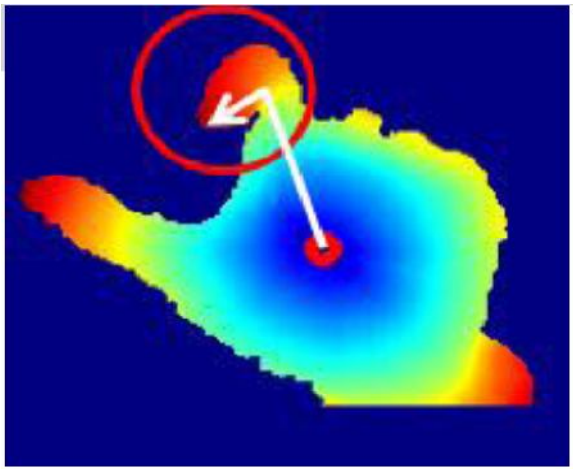

Figure 2. A Geodesic image is color-coded based on distance information from the centroid. Geodesic distance reflects the shortest distance between two points following the surface of the hand.

C.CNN Regressor

Our CNN consists of twelve convolution layers, four pooling layers, and three fully-connected layers, as shown in Fig. 3.At first, a set of input goes through convolution with one stride and zero-padding on each convolutional layer. As it passes through the layer, the number of kernels increasesfrom 24, 32, 48, 96, to 128 to get various features from the input. In the intervals of the convolutional layers, the pooling layers carry out 2x2 max-pooling. After four pooling layer, the fully-connected layers perform regression based on the features. The final output is a vector of 16x3, reflecting sixteen joints and positions in 3-D.The Rectified Linear Unit (ReLU) [9] function is used in all of layers as an activation function. In this work, our CNN regressor is structured deep and wide to get best features through the convolutional layers and to predict the accurate joint positions of hand from the fully-connected layers.

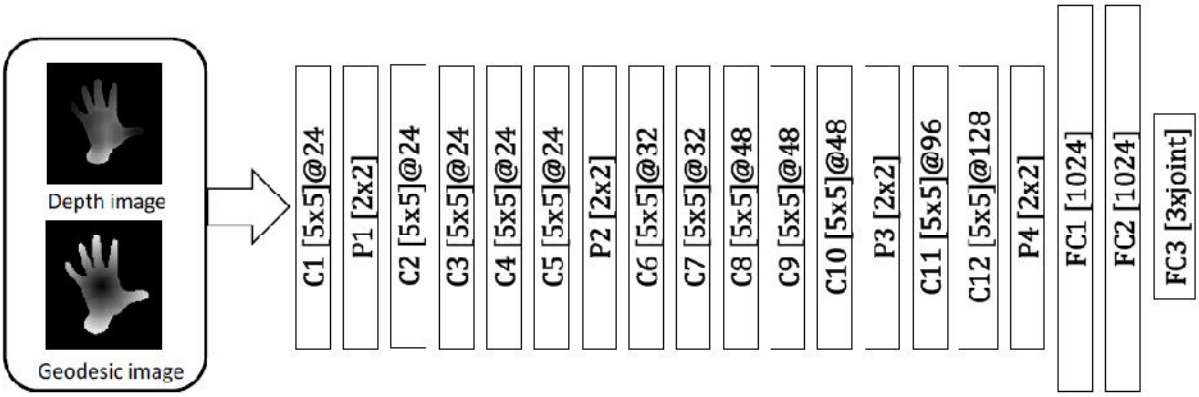

Figure 3.Our proposed architecture of CNN. The network receives the two channel input which is composed ofa pair depth and geodesic distance images. The network contains several pooling layers to reduce the size of extracted features. At the end of the P4 layers, the extracted features are used for regression in the fully connected layers.

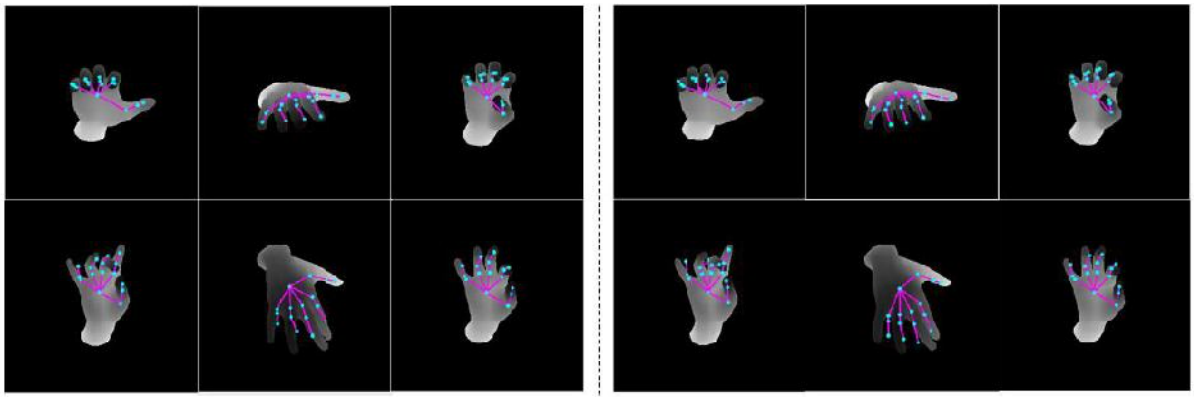

Figure 4. Qualitative results of 3-D hand pose reconstruction. The reconstructed 3-D hand poses are superimposed on their corresponding depth images. The left six images showthe results from depth only, and the right six show is the results from depth and Geodesic.

D.Training and Testing

To estimate thehand joints, we implemented our CNN using Tensorflow[15]on Python 2.7. We used a PC with Ubuntu 16.04 with i7-6850k, 16GB RAM and NVIDIA Geforce GTX 1080. We applied error back-propagation and drop-out with a probability of 0.5 to avoid overfitting. Adam optimizer [2] was used to optimize our problem of minimizing the squared Euclidean distance between the ground-truth joints and estimated joints. The learning rate was set as 1.0E-5. The batch size was 128.

After training, we tested our system. In order to evaluate the performance of the proposed method, a set of test data was randomly extracted from the NYU hand pose dataset.We evaluated our system twice: one with depth images only and the other,the depth and Geodesic images together. We compared the performance by comparing the estimated joint positions to the ground-truth joint positions in terms of Euclidean distance.

III. RESULTS

Table 1 summarizes the quantitative results. When depth images were used only in training, there was an average distance error of 56.51mm with a standard deviation of 42.07 between the true and estimated joint positions. However, with the depth and Geodesic images together in training, an average error of 46.78mm was obtained with a standard deviation of 33.59. We consider this is because the information of hand shape and hand end point can be learned not only from the depth image but also from the Geodesic distance image. Fig. 4 shows a set of exemplar results.

Methods	Error(mm)	
	Mean	STD
Depth only	56.51	42.07
Depth and Geodesic	46.78	33.59

Table 1. Distance errors of 3-D hand pose reconstruction from depth only vs. depth and Geodesic together. Error reflects the average 3-D Euclidean distance of all hand joints between the ground truths and estimated ones.

The quantitative evaluation of two cases : depth only and depth with Geodesic distance, is shown in Fig. 5. We compare the two cases along with several distance thresholds. Each line represents the fraction of test data of which Euclidean distance between the estimated joints and the ground-truth is under the threshold in a set of test data. The case of depth with

geodesic distance as an input shows superior result against the case of depth only.

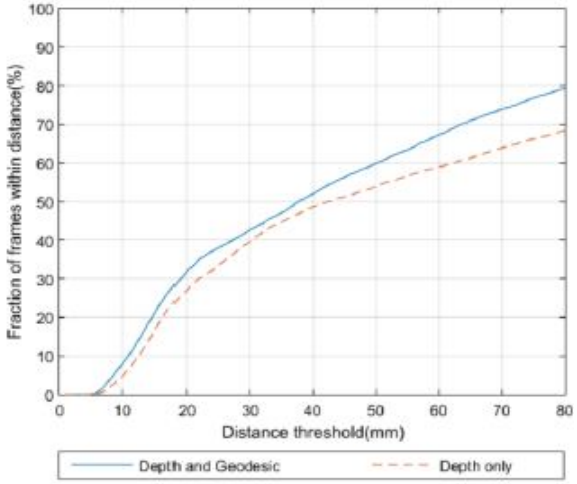

Figure 5.Quantitative evaluation via fraction of frames. Depth only is shown in orange (--), whereas depth and Geodesic is shown in blue (-)

CONCLUSION

In this work, we present a method for 3-D hand pose recognition from a pair of depth and Geodesic distance images. Our results show that complimenting depth information with Geodesic distance information helps 3-D hand pose recognition by increasing the accuracy of regression via CNN. The presented methodology could be used in future applications of smart human computer and machine interfaces.

ACKNOWLEDGEMENT

This work was supported by International Collaborative Research and Development Program (funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea) (N0002252).

REFERENCES

[1] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, and Y. Wei, "Accurate, robust, and flexible real-time hand tracking," Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 3633-3642, 2015.

[2] D. Kingma, and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[3] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," Image and Vision Computing, vol. 30, no. 3, pp. 217-226, 2012.

[4] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: data, methods, and challenges," Proceedings of IEEE International Conference on Computer Vision, pp. 1868-1876, 2015.

[5] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, and B. Luff, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," ACM Transactions on Graphics (TOG), vol. 35, no. 4, pp. 143, 2016.

[6] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints," Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 2088-2095, 2011.

[7] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," European Conference on Computer Vision, pp. 852-863, 2012.

[8] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," arXiv preprint arXiv:1502.06807, 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.

[10] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," arXiv preprint arXiv:1606.06854, 2016.

[11] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,pp. 1106-1113, 2014.

[12] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," Proceedings of the IEEE International Conference on Computer Vision, pp. 3224-3231, 2013.

[13] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," ACM Transactions on Graphics (ToG), vol. 33, no. 5, pp. 169, 2014.

[14] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D hand pose estimation in single depth images: from single-view CNN to multi-view CNNs," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3593-3601, 2016.

[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, and M. Devin, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," arXiv preprint arXiv:1603.04467, 2016.

[16] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," Proceedings of the IEEE International Conference on Computer Vision, pp. 3316-3324, 2015.

[17] H. Guo, G. Wang, and X. Chen, "Two-stream convolutional neural network for accurate RGB-D fingertip detection using depth and edge information," IEEE International Conference on Image Processing, pp. 2608-2612, 2016

★★★