HAND GESTURE RECOGNITION USING A SKELETON-BASED FEATURE REPRESENTATION WITH A RANDOM REGRESSION FOREST

Shaun Canavan, Walter Keyes, Ryan Mccormick, Julie Kunnumpurath, Tanner Hoelzel, and Lijun Yin

Binghamton University

ABSTRACT

In this paper, we propose a method for automatic hand gesture recognition using a random regression forest with a novel set of feature descriptors created from skeletal data acquired from the Leap Motion Controller. The efficacy of our proposed approach is evaluated on the publicly available University of Padova Microsoft Kinect and Leap Motion dataset, as well as 24 letters of the English alphabet in American Sign Language. The letters that are dynamic (e.g. j and z) are not evaluated. Using a random regression forest to classify the features we achieve 100% accuracy on the University of Padova Microsoft Kinect and Leap Motion dataset. We also constructed an in-house dataset using the 24 static letters of the English alphabet in ASL. A classification rate of 98.36% was achieved on this dataset. We also show that our proposed method outperforms the current state of the art on the University of Padova Microsoft Kinect and Leap Motion dataset.

Index Terms— Gesture, Leap, ASL, recognition

1. INTRODUCTION

Automatic hand gesture recognition has a wide range of applications in fields such as human-computer interaction, computer gaming, automatic sign language recognition, and robotics. There has been some success with hand gesture recognition using wearable devices [11][14], however, vision based methods [10][4][8][3][9][7] are less invasive and allow for more natural interaction. With the release of affordable consumer grade cameras such as the Leap Motion Controller (Leap), vision based methods can be more readily explored. Recently, there has been some success with using 3D motion trajectory captured from the Leap to classify both letters and numbers [4]. Using the Leap Motion and a support vector machine Marin et al. [13] achieved 81.5% gesture classification accuracy, using Leap Motion features alone, on the University of Padova Microsoft Kinect and Leap Motion dataset [12][13].

The Leap makes use of two cameras and three infrared LEDs, which give an interactive area of 2.6 feet above the controller and 2 feet on each side. It applies mathematical algorithms to the raw sensor data acquired from the cameras. The Leap is designed specifically for hand tracking and because of this it is natural to use this camera for hand gesture recognition. The Leap skeletal tracking model gives us access to the following information that we use to create our feature descriptors; (1) palm center; (2) hand direction (3) fingertip positions; (4) total number of fingers (extended and non-extended); and (5) finger pointing directions. Using this information from the Leap we propose six new feature descriptors which include (1) extended finger binary representation; (2) max finger range; (3) total finger area; (4) finger length-width ratio; and (5-6) finger directions and distances. Each of these feature descriptors are detailed in section 2. Once we have each of the features descriptors they are concatenated into one feature vector which is then used as input to a random regression forest to classify the gesture. See figure 1 for an overview of our system. A summary of the main contributions of this work follows:

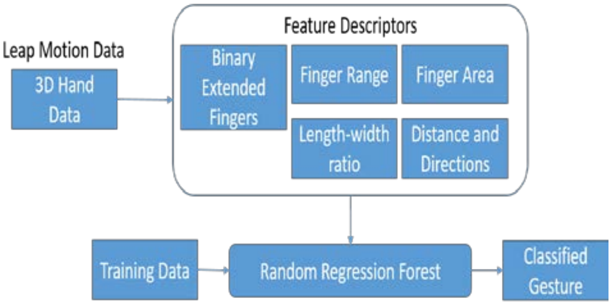

Figure 1. Proposed gesture recognition overview.

(1) We propose six novel feature descriptors, captured from the Leap; extended fingers binary representation, max finger range, total finger area, finger length-width ratio, and fingertip directions and distances.
(2) We propose the use of a random regression forest to classify hand gestures from skeleton-based feature representation constructed from 3D information capture from the Leap.
(3) We test our new proposed features and classification scheme on the publicly available dataset [12][13], as well as a new Leap Motion dataset that contains the 24 static letters of the ASL alphabet.
(4) We show the power of our proposed features and classification scheme by comparing to state of the art.

2. LEAP MOTION FEATURE DESCRIPTORS

Using two cameras and three infrared LEDs, the Leap can infer the 3D position of a subject's hand. As such it is a natural fit to use the Leap for gesture recognition, as relevant information about each gesture can be extracted from the controller. This information can be used to enhance hand gesture classification accuracy. Given this information we propose six new features to help with hand gesture recognition, which are detailed below.

2.1. Hand scale factor based on number of fingers

In order for our proposed features to remain scale invariant, those features need to be scaled. Our hand scaling factor is based on the current number of fingers, the fingertip positions of the extended fingers, and the palm center. We calculate our new scale, *s*, by first finding the average extended fingertip position as

$$A_L = \frac{\sum_{i=1}^{n}(f_{L_i})}{n} \quad (1)$$

where n is the number of extended fingers, f_{L_i} is the fingertip positions of the extended fingers. Given this average fingertip position we then find the distance from this new position to the palm center as

$$s=\sqrt{(A_{Lx}-P_{Lx})^2+(A_{Ly}-P_{Ly})^2+(A_{Lz}-P_{Lz})^2} \quad (2)$$

where (A_{Lx}, A_{Ly}, A_{Lz}) is the 3D average fingertip position and (P_{Lx}, P_{Ly}, P_{Lz}) is the 3D palm center.

2.2. Extended finger binary representation

Our extended finger binary representation is a feature descriptor that details exactly which fingers are extended for the current gesture. For this we create a 5-bit feature vector where each bit represents one of the fingers. The most significant bit (MSB) represents the thumb and the least significant bit (LSB) represents the pinky. We use a binary representation of the extended fingers to populate our feature vector as

$$e_{b_i} = \begin{cases} 1, & f_i \in E, \\ 0, & otherwise. \end{cases} \quad (3)$$

where e_{b_i} is the binary finger bit, f_i is the current finger, E is the set of extended fingers, and *i* = [1,5] for each finger.

2.3 Max finger range

Our max finger range feature allows us to classify gestures that vary in size based on the number of fingers. For example, a gesture that has one finger extended vs all five fingers will have a different max finger range. Using the palm center, and fingertip positions of the extended fingers we construct our max finger range feature descriptor. We take the max (x,y) values of the extended fingertip positions with respect to the palm center. As an example, a small max x value can indicate that the middle finger is extended, while other fingers further from the palm center, such as the pinky, are not. A large max x value can indicate the opposite that the middle finger is not extended and the pinky is.

2.4 Total finger area

Total finger area also gives us information about where the fingers are pointed. Using the extended fingers, we calculate the Euclidean distance between all combinations of extended fingers. We then look at which two extended fingers give us the largest calculated distance and create a triangle between these two fingertip positions and the palm center. Given this triangle, we calculate the area, which is divided by the number of extended fingers. This allows us to differentiate between two distinctly different gestures that have the same area. For example, all gestures that have the pinky and thumb extended could have the same finger area. We can overcome this by making this area in relation to the number of extended fingers. See figure 2(a) for an example.

2.5 Finger length-width ratio

Finger length-width ratio is complementary to our proposed total finger area, as it allows us to get the total ratio between all extended fingers. Our finger length ratio is an extension of the work detailed in [6], where Ding et al. use a length-width ratio to differentiate the difference between the number 0 and the alpha O. We expand upon this idea to create a finger length ratio for all gestures. Similar to the area, we look at all combinations of distances between each of the extended fingertip positions. We define the width, *w*, as the maximum distance between all combinations of extended fingertip positions. The length is calculated as the distance from the palm center to each of the extended fingertip positions. The length, *l*, is defined as the max calculated distance. Our new finger length-width ratio, *r*, is defined as $r = l/w$. See figure 2(b) for an example.

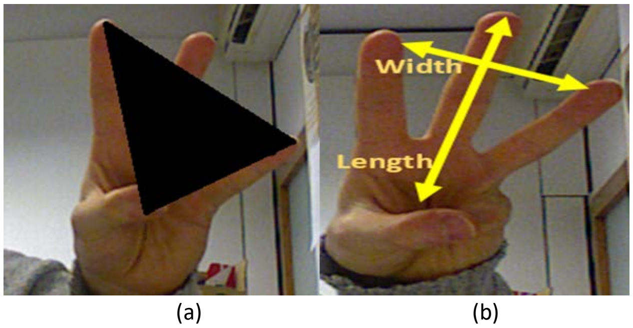

Figure 2(a). Total finger area on RGB image. Figure 2(b). Finger length-width ratio on RGB image. Note: these images modified from [12][13] for illustration purposes only.

2.6 Fingertip directions and distances

The direction that the extended fingers are pointing, along with the distance between the palm center and the fingertip positions can be an important feature for classifying gestures. This allows us to tell if the fingers are curled under to pointing straight up, and each direction in between. As previously detailed, the Leap SDK gives us the direction that each of the fingers are pointing. The directions are directly inserted into our feature vector. We also propose the use of another complimentary feature to the fingertip directions. We refer to this new feature as the fingertip distances. During calculation of our other Leap features, we must calculate the distance from the palm center to each of the fingertip positions. We use these calculated distances as a feature descriptor to help us classify different gestures.

3. EXPERIMENTS AND EVALUATION

We conducted experiments on the publicly available University of Padova Microsoft Kinect and Leap Motion dataset [12][13], which consists of hand gestures captured from both the Leap and Kinect. It contains 10 different gestures performed by 14 different subjects a total of 10 times each for 1400 different samples. Each sample includes depth and RGB images, the raw depth data, and a csv file that includes 20 features to classify gestures. See figure 3 for sample images of the 10 gestures from this dataset.

We also constructed an in-house dataset using the 24 static letters of the alphabet in American Sign Language. The dataset consists of 14 subjects performing each gesture 10 times for a total of 3360 samples. This dataset contains Leap skeletal data in CSV format. See table 2 for listing of features and figure 4 for example images. This dataset is publicly available for comparisons.

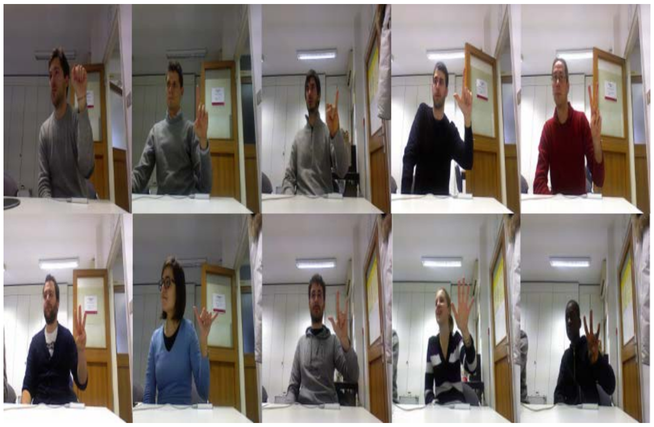

Figure 3. Sample images from [12][13]. Note: RGB images shown only for illustration purposes.

3.1 Random regression forests

Regression trees [2] are a powerful tool that can be used for classification. Regression trees work by splitting the problem into smaller ones that can be solved with simple predictors. Each node of the tree is a question whose answer directs towards the left or right child. During training the data is clustered so that simple models can achieve good results. While regression trees can give good results, they are prone to overfitting. Breiman [1] found that the overfitting can be overcome by a collection of randomly trained trees. Random forests work by constructing multiple, randomly trained, regression trees and taking the mean classification, of those trees, as the output. Schmidt et al. [15] were able to successfully use random regression forests to determine a set of gestures from Leap data. Due to this and the ability of random forests to overcome overfitting, their speed, and power for classification they are a natural fit for our gesture classification scheme.

3.2 Exhaustive feature evaluation

In order to evaluate the efficacy of our proposed features, we did an exhaustive evaluation of all combinations of our feature descriptors. For all possible combinations of our Leap features we create a separate feature vector. For the six Leap features proposed in section 2, there were a total of 63 possible feature vectors. All feature vectors were constructed for all 1400 samples available in [12][13]. A random regression forest was used for our hand gesture classification, where 10-fold cross validation was used. In this classification scheme, the data is randomly split into 10 subsets. In these subsets, one is used for testing and the other nine are the training data. This is done for all of the subsets, where each is used as the testing data. The average error across all trials is then used.

Using this classification scheme resulted in 100% classification for all 1400 samples. This classification rate can be attributed to the proposed extended finger binary representation. When the gestures classified use different fingers, as is the case for all of [12][13], this feature is highly accurate. When this feature was included in any of the 63 tested feature vectors the classification rate was always 100%. Due to this, we also make note of our highest classification rate without the extended finger binary representation, which was 81.71%. This classification rate was obtained by using the combination of max finger range, total finger area, and finger length-width ratio. See table 1 for comparisons of our results with those in [13]. As can be seen from this table, even without our extended finger binary representation feature, our Leap features still outperform current state of the art, showing the classification accuracy of our proposed approach.

Table 1. Comparisons with [13].

Method	**Classification Rate**
Marin et al. [13]	81.5%
Proposed with binary representation	**100%**
Proposed without binary representation	81.71%

3.2 ASL alphabet recognition

In order to test our proposed method on more varied gestures, we use our in-house dataset containing the 24 static letters of the alphabet. The alphabet in ASL can be challenging for automatic hand gesture recognition due to the similarity of some of the letters, however, our proposed method was still able to accurately classify the majority of the 24 tested letters. We used the same 10-fold cross validation classification pipeline as our experiments on [12][13]. The classification rate on all 3360 samples was 98.36%. There are some instances in the American Sign Language where the letters look extremely similar, yet our proposed classification scheme was still able to accurately classify those letters.

In classifying each letter, *t* was incorrectly classified as *n* more than any other letter. While *t* has the lowest classification rate with 92.8%, with it being incorrectly classified as *n* 3% of the time, the classification rate for *n* was not the same. It had a classification rate of 98.5% and it was only misclassified as *t* once. This disparity could be attributed to the Leap. While the Leap is a powerful tool for gesture recognition, it can have some incorrect data as it can get confused as to which finger is where, as well as incorrectly showing a finger as bent or extended when the gesture is showing the opposite.

The confusion matrix of all 24 evaluated letters of our ASL dataset is shown below in table 3. Each letter has 140 samples. The confusion matrix details the number of times each of the letters were both correctly and incorrectly classified and with what letters.

Table 2. Listing of Leap features from ASL dataset.

Feature	**Data Type**	**Feature type**
Extended fingers	Binary	Fingers (5)
Finger directions	3D vector	Fingers (5)
Fingertip positions	3D vector	Fingers (5)
Extended fingertip positions	3D vector	Fingers (5)
Hand direction	3D vector	Hand
Palm normal	3D vector	Hand
Palm Position	3D vector	Hand
Number of fingers	Unsigned	Range (1-5)

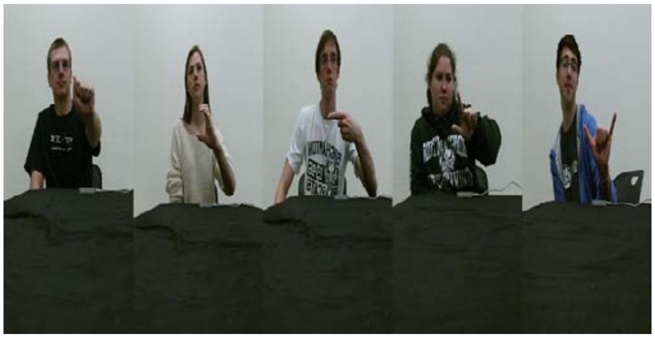

Figure 4. Example images from in-house ASL dataset (from left to right A, C, G, L, Y). NOTE: RGB images shown only for illustration purposes.

Table 3. Confusion matrix of the 24 evaluated letter of ASL.

	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
A	139	1																						
B		139																				1		
C			139								1													
D				138													2							
E				1	139																			
F					1	139																		
G							137	3																
H								140																
I				2					137										1					
K										137											1	1	1	
L					1						139													
M		2										136							1			1		
N	1												138						1					
O														140										
P															140									
Q																140								
R				1									1				133			2			3	
S	1			2											1			136						
T	1									1			4	1			2		130				1	
U										1							2			134	2	1		
V				1																2	137			
W																	2					138		
X																							140	
Y																								140

As can be seen from the confusion matrix many of the letters were misclassified as other similar letters. For example the letter *d* was only misclassified twice with the letter *r*. This misclassification can be attributed to the similarities of the two letters the incorrect data that can be acquired from the Leap.

4. DISCUSSION AND FUTURE WORK

We have presented a method for hand gesture recognition that uses a random regression forest with feature descriptors created from Leap data including an extended finger binary representation, finger ratio and area, a finger length-width ratio, and finger directions and distances. We have shown that our proposed method outperforms current state of the art on a publicly available dataset [12][13]. We have also created a new dataset that consists of 24 static letters of the American Sign Language. Our experiments on this dataset are encouraging with a classification rate of 98.36%.

While the current results are encouraging, we are looking at multiple extensions of this work, including working with more challenging datasets and recognizing dynamic gestures by incorporating the gesture’s velocity vector as [5] has had some success with this. We are interested in creating 3D statistical shape models from the 3D hand data captured from the Leap. In doing this we could accurately model variations in the different gestures to use for classification. We are also investigating real-time functionality in virtual reality applications, as devices such as the Oculus Rift can easily be integrated with the Leap. There has also been success with using deep learning and 3D depth data captured from the Microsoft Kinect to recognize hand gestures [16][17]. We are currently investigating using deep learning with our features collected from the Leap.

5. REFERENCES

[1] L. Breiman, “Random forests,” *Machine Learning*, 45(1), pp.5-32, 2001.

[2] L. Breiman, J. Friedman, R. Olshen, and C. Stone, “Classifcation and regression trees,” *Wadsworth and Brooks*, Monterey, CA, 1984.

[3] F-S. Chen, C-M. Fu, and C-L. Huang, “Hand gesture recognition using a real-time tracking method and hidden marov models,” *Image and Vision Computing*, 21, pp. 745-758, 2003.

[4] Y. Chen, Z. Ding, Y. Chen, and X. Wu, “Rapid recognition of dynamic hand gestures using leap motion,” *IEEE Conf. on Information and Automation*, 2015.

[5] C-H. Chuan, E. Regina, and C. Guardino, “American sign language using Leap Motion sensor,” *International Conference on Machine Learning and Apps*, 2014.

[6] Z. Ding, Z. Zhang, Y. Chen, and X. Wu, “A Real-time dynamic gesture recognition based on 3D trajectories in distinguishing similar gestures,” *Conference on Information and Automation*, 2015.

[7] F. Dominio, M. Donadeo, and P. Zanuttigh, “Combining multiple depth-based descriptors for hand gesture recognition,” *Pattern Recognition Letters*, 2013.

[8] M. Funasaka, Y. Ishikawa, M. Takata, and K. Joe, “Sign language recognition using Leap Motion controller,” *Intl. Conf. on Parallel and Distributed Processing Techniques and Applications*, 2015.

[9] K. Hu and L. Yin, “Multi-scale topological features for hand posture representation and analysis,” *Intl. Conference on Computer Vision*, 2013.

[10] X. Liu, and K. Fujimura, “Hand gesture recognition using depth data,” *IEEE Conference on Automatic Face and Gesture Recognition*, 2004.

[11] Z. Lu, X. Chen, Q. Li, and X. Zhang, “A Hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices,” *IEEE Trans. On Human-Machine Systems*, 44(2), 2014.

[12] G. Marin, F. Dominio, and P. Zanuttigh, “Hand gesture recognition with leap motion and kinect devices,*” Intl. Conf. on Image Processing*, 2014

[13] G. Marin, F. Dominio, and P. Zanuttigh, “Hand gesture recognition with jointly calibrated Leap Motion and depth sensor,” *Multimedia Tools and Applications*, pp. 1-25, 2014.

[14] A. Pandit, D. Dand, S. Mehta, S. Sabesan, and A. Daftery, “A simple wearable hand gesture recognition device using iMEMS,” *International Conference of Soft Computing and Pattern Recognition*, 2009.

[15] T. Schmidt, F.P. Araujo, G.L. Pappa, and E.R. Nascimento, “Real-Time hand Gesture Recognition Based on Sparse Positional Data,” *Brazalian Workshop on Computer Vision*, 2014.

[16] A. Tang, K. Lu, Y. Wang, J. Huang, H. Li, “A real-time hand posture recognition system using deep nerual networks,” *ACM Trans. On Intelligent Sys. and Technology*, 6(2), 2015.

[17] D. Wu, L. Pigou, P-J. Kindermans, N. Do-Hoang Le, L. Shao, J. Dambre, and J-M. Odobez, “Deep dynamic neural networks for multimodal gesture segmentation and recognition,” *IEEE Trans. On Pat. Analysis and Mac. Intelligence*, 38(8), 2016.