Sign Language Recognition using Leap Motion Controller

Makiko Funasaka, Yu Ishikawa, Masami Takata, and Kazuki Joe
Department of Information and Computer Sciences, Graduate School of Humanities and Sciences,
Nara Women's University, Nara, Japan

Abstract – *Making deaf people use an alternative method instead of current voice input to ICT equipment, we propose a sign language recognition method using Leap Motion Controller. As decisions using sign language, 16 kinds of decisions that focus on characteristic of hands and fingers are proposed. Sign language recognition algorithm is constructed using 16 kinds of decisions. The constructed flowchart is differing as order of decisions, recognition rate for all letters change from the difference accuracy rate of decisions. For sorting of decisions is enormous combination, genetic algorithm is applied to search for the optimal solution in the automatic construction of sign language recognition algorithm.*

Keywords: Leap Motion Controller, sign language recognition, Genetic Algorithm

1 Introduction

As smart phones and tablet devices have been improved, voice input and voice recognition interfaces have been widely used. The iPhone has Siri[1] that can answer questions by using natural language processing and web services. The Google Search[2] is a search engine that accepts voice input, too. The advantage of voice input is faster operations than by keybord or pad touch, so it is the minimal burden for ordinal users, in particular, for elderly users who are not familiar with text input. However, the voice input and the voice recognition interfaces is extremely difficult for deaf people. Therefore, any interfaces, which do not need voice input, should be developed for those poeple.

A sign language is a kind of communication means for deaf people. By using a sign language as an input interface to ICT devices, it is possible for deaf people who are considered very difficult to input by conventional keyboard and touch pad. A sign language use visual information with the use of finger, hand and arm operations. At the same time some finger operations are used with a part of the face such as line of sight and mouth. Fingerspelling can represent one of the alphabet 26 letters in the form of fingers.

In existing research for sign language recognition, the image recognition of color images, depth images and hand shapes are used[3]. Since it must be taken with colored gloves[3], the glove worn is not convenient. The image recogniton requires long computation time to detect the hand and fingers. Thus, it takes relatively a long interval to obtain the final recogntion result. In the case of the recognition with Kinect[4], large space is required for skeletal tracking. It is difficult to recognize the fingerspelling anywhere with Kinect. Therefore, sign language recogniton is required using a compact device that can directly recognize the shape of fingers or hands anywhere.

In this paper, we propose a fingerspelling language recogniton method using Leap Motion Contoroller[5][6]. Putting hands and fingers over a Leap Motion controller, the fingerspelling recognition is performed. Leap Motion has skeletal tracking that recognizes the framework of fingers to obtain a highly accurate various data such as the position of finger bones and the degree of the thumb and index finger. In addition, the use of Leap Motion allows fingerspelling recogniton without any physical contact.

The rest of the paper is constructed as follows. Section 2 provides related works of sign language recogniton in detail. Section 3 explains a fingerspelling recogniton method. Section 4 describes the search for the optimal solution in the automatic generation of fingerspelling recogniton algorithms. In Setion 5, we perform experiments using the proposed method.

2 Related works

As related works of sign language recognition, we explain three studies.

In the first study, colored gloves are adopted to obtain hand shape recognition applied to the conversion of a sign language [3]. Colored gloves dyed with 6 different colors are worn and the feature vectors are calculated using the Morphological Principal Component Analysis from the captured images. Hand detection and finger modeling are performed in natural background by using the colored gloves. The proposed feature vector extraction method makes the camera distance be highly independent. To analyze the performance of feature extraction, neural network is trained for the position recognition of a sign language with 26 alphabet letters. The neural network is a feed-forward three layers perceptron type with the backpropagation learning method. The numbers of neurons in the intermediate and output layers are 42 and 26, respectively. The 26 alphabet letters are characterized by a 20-dimensional vector from finger data without the palm of the hand. The result of experiments using a test set of 30 samples by each hand shows the recognition rate of 93.396%.

Second, in order to recognize a static sign language by hand, a robust approach using a novel combination of features

is proposed [7]. The features are the color of images, the depth of images and the hand shape. Obtaining depth, color and skeleton data by Kinect, the proposed accurate hand segmentation divides them into a hand color image and a hand depth image. Color and depth feature vectors are characterized by the Local Binary Pattern (LBP) histogram calculated from the hand color image and the hand depth image. Extracting a hand shape from the hand depth image, the shape feature vector is characterized. The combination of these three feature vectors are used for the recognition using template matching and Support Vector Machine (SVM). Two experiments are conducted. First, to examine the classification accuracy in combinations of different features, some experiments are conducted with 4 types of feature combinations; depth only, color only, color and depth or color, depth and hand shape. The three conditions of skin color, hand size and distance of camera are changed at data collection, and the proposed hand segmentation algorithm is used with different lighting conditions. The dataset composed of 8 different hand poses denoting letters of the fingerspelling alphabet from 'A' to 'H' is generated. Each sign is performed 10 times by 12 non-expert signers. It gives a total of 120 color and depth images for each of the 8 alphabet signs. As the result of the experiment, when the proposed feature combination of color, depth and hand shape is applied, the recognition rate of 95% is higher than other combinations. Next, the target signs from 'A' to 'Y' excluding sign 'J' and 'Z' which involve motion are used for the experiment. In a fingerspelling data set, 500 color and depth image pairs per sign are obtained and five data sets are used for the experiment. The result of the experiment is the recognition rate of 92.14%.

The third is a study to recognize the alphabet fingerspelling 26 letters using Leap Motion Controller [6]. In this study, finger data is obtained from the Leap Motion API. Palm data consists of the unit direction vector of the palm, the position of the palm center, the velocity of the palm and the accuracy of the data. At the same time, the grab strength, the pinch strength, the sphere center and the sphere radius are obtained. The finger data consists of the direction of each finger, the length of each finger, the tip velocity and the position of joints for distal phalanges, intermediate phalanges, proximal phalanges and metacarpals. To apply machine learning the data obtained from the Leap Motion API, feature vectors are calculated. As the features of palm, the pinch strength, the grab strength, the average distance, the average spread and the average tri-spread are used for the machine learning. The average distance is calculated as the sum of the distances between the fingertips in adjacent frames. The average spread for the palm is estimated based on the distance between adjacent fingertips. The tri-spread is the triangular area between two adjacent finger tips and the midpoint of the two finger's metacarpal positions. The average tri-spread is calculated by adding the triangle area of all pairs of fingers and dividing by the total number of the frames. For each finger, extended distance, dip-tip projection, orderX and angle are derived. The extended distance is the maximum distance of all points of the finger from the palm center. The Dip-tip projection is the projection of the dip-to-tip vector onto the palm normal vector. The OrderX is the order of the finger along the x-z plane with respect to other fingers. For the experiment, data are collected from two (deaf and normal) teachers of deaf education. In order to classify the 26 letters using, a k-nearest neighbor method is applied and the recognition rate is 72.78%. The use of SVM improves the recognition rate to 79.83%.

Among the above studies, in the case of colored glove based recognition, the user may feel a troublesome to wear them and a photograph picture must be taken with wearing the colored glove, which is not very convenient. In the case of Kinect based recognition, a large space is required to obtain depth as well as and color image for the skeletal tracking, which is not easy for ordinal use. In the case of Leap Motion based recognition, the machine learning requires large computation for a new person.

3 Sign language recognition using a sensor device

3.1 Detection of finger using Leap Motion Controller

Hardware and software of Leap Motion Controller are used for get the following five functions: 1) hands detected in a frame including rotation, position, velocity and movement from the last frame, 2) all fingers and pointing tools recognized by hand with rotation, position and velocity, 3) the exact pixel location on a display pointed at by a finger or a pointing tool, 4) recognition of basic finger gestures such as swipes and taps, and 5) detection of position and orientation changes between frames [6]. In this paper, hands and fingers are detected to obtain the normal vector of the palm, coordinates of fingertips and finger bone, the direction vector of arm and the direction vector of the fingertip.

3.2 Decision tree and the recognition rate for fingerspelling letters

The Recogniton target is fingerspelling 24 letters without movement out of 26 letters. Although there are several alphabetic representations for fingerspelling, we use the fingerspelling representation as shown in Fig.1 [9].There are differences in fingerspelling representations ; if the palm of hand is facing the opponent, if the back of hand is facing the opponent, or which finger is bent how. Considering these representations, the conditional branches constructing a decision tree are 16 kinds of 'a' to 'p' as shown in Table 1, and the decision tree is suitable for programming. The detail of each conditional branch is shown in Table 1. The conditional branch is just two ways of Yes or No. Figure 2 is an example of decision trees. In Figure 2, left arrow is Yes and right arrow is No. It should be noted that, "a finger is fully extended" means all of the first, the second and the third

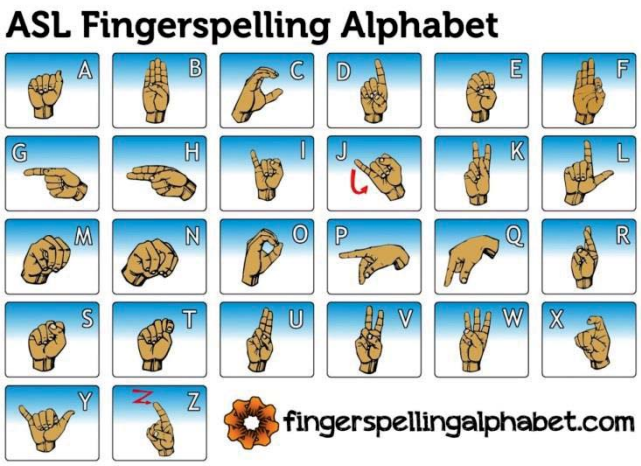

Figure 1 : Alphabetical fingerspelling

Table 1 : Detail of each conditional branch

	Conditional Branch
a	The palm of hand is facing the opponent.
b	Hand orientation is above.
c	There is a finger that makes a wheel with the first and the second joints
d	The third joints are extended except the thumb.
e	The back of hand is facing the opponent.
f	Hand orientation is below.
g	All of the first and the second joints are bent.
h	The thumb is fully extended.
i	The index finger is fully extended.
j	The middle finger is fully extended.
k	The ring finger is fully extended.
l	The pinky finger is fully extended.
m	The thumb is contact with the middle finger.
n	The ball of the middle finger is on the nail tip of the index finger.
o	The index finger and the middle finger are separated.
p	The thumb and the index finger are open.

joints are extended. Also, "hand orientation" means the direction of the fingertips from the wrist.

We perform preliminary experiments for the recognition rate at each conditional branch. A right-handed examinee puts her hand over a Leap Motion Controller to make the controller recognize her fingerspelling. The recognition rate is calculated

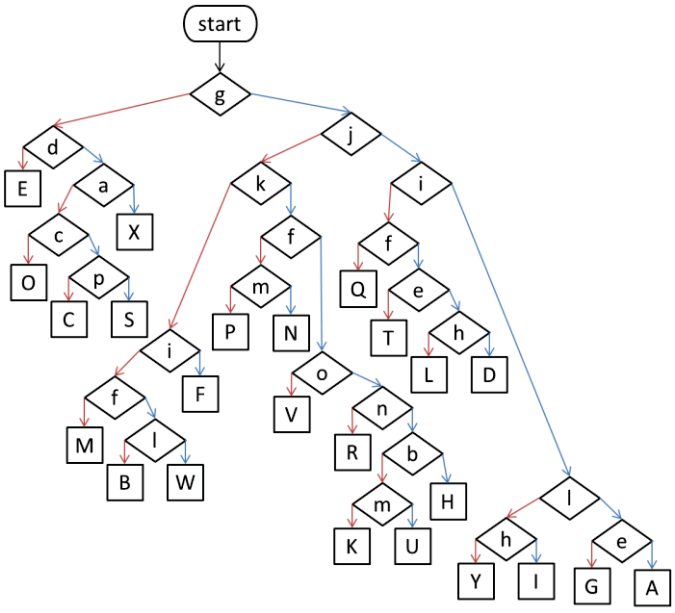

Figure 2 : An example of decision trees

Table 2 : The correct answer rate of each condition (%)

a	95.00	e	96.67	i	96.67	m	90.00
b	92.92	f	97.08	J	95.83	n	92.92
c	90.83	g	99.58	k	93.33	o	97.50
d	95.83	h	95.83	l	95.83	p	100.0

by 10 times experiments for all decision tree nodes.

The decision tree consists of nodes (non-terminal) and leaves (terminal). The node represents 16 conditional branches as shown in Table 1 while the leave represents fingerspelling 24 letters. Starting from the root (top node), there is a pass to get to each leaf. Each pass consist of one or more nodes. A leaf may have a lot of combinations of nodes to reach. However, depending on the combination of nodes that is required for each fingerspelling, the overall recognition rate is different. For example, when we want fingerspelling 'M', the node 'j', 'k', 'i' and 'f' are Yes and the node 'g' is No. By using the recognition rates of nodes 'j', 'k', 'i', 'f' and 'g' of Table 2, the recognition rate of fingerspelling 'M' in Figure 2 is calculated as $0.9583 * 0.9333 * 0.9667 * 0.9708 * 0.9958 * 100 = 83.58$. Calculating the recognition rates for other fingerspelling in the same manner, the average recognition rate of a decision tree shown in Figure 2 is 81.97%.

Note that there is a decision tree where a fingerspelling has multiple passes to reach the terminal node. Namely, there is a conditional branch whose decision does not affect the recognition result. In this case, the recognition rate of the conditional branch is 100%.

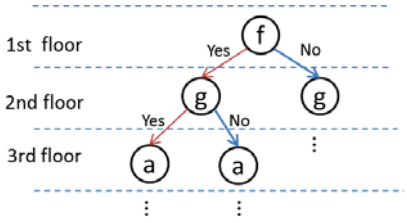

Figure 3: Structure of decision tree

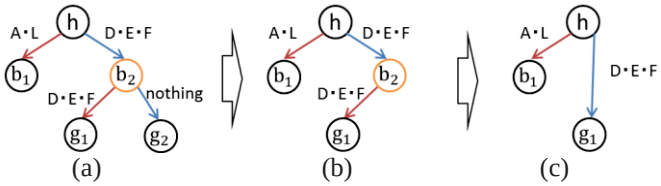

Figure 4: Unnecessary nodes: (a) shows a case including an unnecessary node. (b) shows a state deleting 'g_2'. (c) shows a state deleting 'b_2'.

3.3 Fingerspelling recognition algorithm

The structure of the decision tree is shown in Figure 3. The left arrow represents Yes while the right arrow represents No, where each node has a condition from "a" to "p". The depth of the decision tree to be generated is 17 with 16 layers for nodes (conditional branch) and 1 layer for leaves (final fingerspelling letter). At the root node of a decision tree, all the fingerspelling letters are candidates. When visiting each node, the candidates are sieved by the condition to select less candidates until the number of candidates becomes one, namely it is a leaf. The leaf represents a fingerspelling letter that is determined uniquely by the decision tree.

In the fingerspelling recognition algorithm, first 16 kinds of conditions are arranged in an appropriate order. This order corresponds to the hierarchical order of the decision tree. In other words, the nodes in the same hierarchy have a conditional branch from the same node. In addition, when generating a decision tree in some hierarchical order, it may include unnecessary nodes. The unnecessary node has a condition that all the fingerspelling letters from the parent node are sent to a child node without splitting the fingerspelling letters. Figure 4 is a diagram showing the process of deleting an unnecessary node. Figure 4 (a) shows a state including the unnecessary node (b_2). Node 'h' has two branches for the left side of node 'b_1' and the right side of node 'b_2'. In addition, the node 'b_2' has two branches for the left side of node 'g_1' and the right side of node 'g_2'. 'g_1' has 3 letters 'D', 'E' and 'F' while 'g_2' has nothing. Namely, 'g_2' is removable. Figure 4 (b) is the state after removing 'g_2'. In Figure 4 (b), since all the fingerspelling letters of 'b_2' from the parent node ('D', 'E' and 'F') are sent to the child node (g_1) without splitting the letters, 'b_2' is unnecessary. Figure 4 (c) is a state that 'b_2' is deleted. In this way, if unnecessary nodes are included, it is possible to shorten the processing of fingerspelling recognition by deleting them as shown in Figure 4.

Since there are 16 kinds of conditions, the possible number of decision trees to be generated is 16!, which is about 20 trillion ways. In addition, different decision trees have different recognition rates for fingerspelling. Therefore, it is necessary to obtain an optimum decision tree in consideration of the correct answer rate of Table 2.

4 Search the optimal fingerspelling recognition method

The conditions for fingerspelling recognition proposed in Section 3 are changeable by order. Since the correct answer rates in Table 2 are different, the average recognition rate for fingerspelling letters obtained in decision trees is changeable by the order of conditions. We propose an algorithm for searching the optimal solution for automatically generated decision trees with varying the order of 16 types of conditions.

As the optimal solution search for combinatorial optimization problem, there are a Branch and Bound method that gives a strict solution and a Genetic algorithm [10] that gives approximate solutions. In the case of Branch and Bound, although limited operations narrow the search space, it requires huge computation to search all over the limited search space. Therefore the amount of computation is unrealistic time consuming in order to obtain the result.

ID3 is known as a decision tree learning algorithm. It is for each independent variable with determining the average amaount of information and the expected value in the case of determining the value of variable. The largest variable is selected so that the operation of the variable to the node of the tree is performed recursively. In the case of ID3, creating a decision tree considering the average recognition rate is too difficult.

Using the Genetic algorithm, it is not possible to find the optimum solution because it gives approximate solutions while it is determined with less computation time to get solutions close to the optimal solution. We employ a Genetic alogorithm for the combinatorial optimization problem of 16 kinds of conditions to find the quasi optimal solution in realistic computation time. The constraints of the combinatorial optimization problem to be handled in this paper are the following.

1. 16 kinds of conditions are arranged so that the maximum average recognition rate of all fingerspelling is obtained.
2. Each condition is selected only once.

In this paper, we use a simple genetic algorithm with the most basic configuration. One point crossover and a roulette selection are adopted. The Order Representation [11] is used for the gene expression. Using the Order Representation, it is possible to satisfy the above constraint 2 as well as the one point crossover without causing lethal genes. The order Representation consists of the list L_1 where conditions are arranged alphabetically and the list L_2 where conditions are

arranged along a phenotype. The phenotype gives the order to perform the conditions. Each conditional branch destination of list L_2 is searched in list L_1 to be replaced, namely the initial conditions are removed from the list L_1. The above operation is repeated. Finally list L_2 becomes a list represented by a genotype.

Fitness is the recognition rate calculated with the automatic generation algorithm for fingerspelling recognition described in subsection 3.2 and 3.3. In the case of mutation operations, the i-th number from the beginning of the gene list by the Order Representation is rewritten by a number less than $N + 1 - i$ to avoid a lethal gene.

By the genetic and evolution operations, the individuals with high fitness to the target problem are increased. Finally, the fitness of the best individual becomes a suboptimal solution, and the genes of the best individual represent the optimal order for the conditions.

5 Experiments

5.1 Experimental method

We apply a Genetic algorithm to the automatically generated fingerspelling recognition algorithms described in subsection 3.3. To obtain suboptimal solutions, we perform three experiments: 1) Find appropriate crossover probability, 2) Find appropriate mutation probability, and 3) Obtain suboptimal solutions using 1) and 2).

In 1), the crossover probability is varied 0.7 to 0.9 by 0.05 while the mutation probability is fixed to 0.08. In 2), the mutation probability is varied from 0.02 to 0.1 by 0.02 while the crossover probability is fixed to 0.8. For both 1) and 2), the number of individuals and the maximum number of generations are 1,000 and 300, respectively.

Both 1) and 2) are performed 10 times for each parameter. In 3), the appropriate parameters obtained in 1) and 2) are used. 3) is performed 100 times with 1,000 individuals and 300 generations.

5.2 Experimental results and discussion

Figure 5 shows the fitness values of the best individual at the 300th generation with various crossover probabilities of 1). As the result of 1), the quasi-optimal solution is 82.71% achieved 3, 3, 5, 4 and 4 times out of 10 times trials with the crossover probability from 0.7 to 0.9 by 0.05, respectively.

Figure 6 shows the fitness values of the best individual at the 300th generation with various mutation probabilities of 2). As the result of 2), the quasi-optimal solution is 82.71%

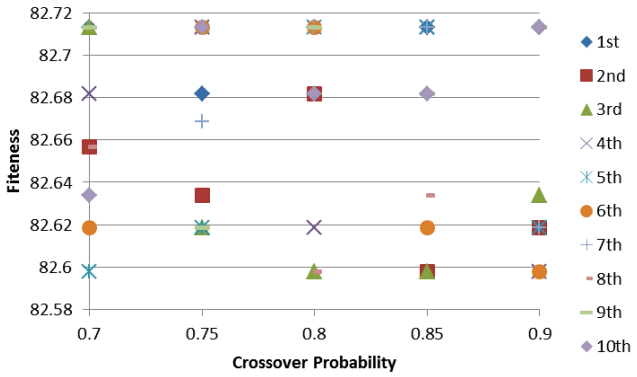

Figure 5: Fitness of the best individuals at the 300th generation with various crossover probability

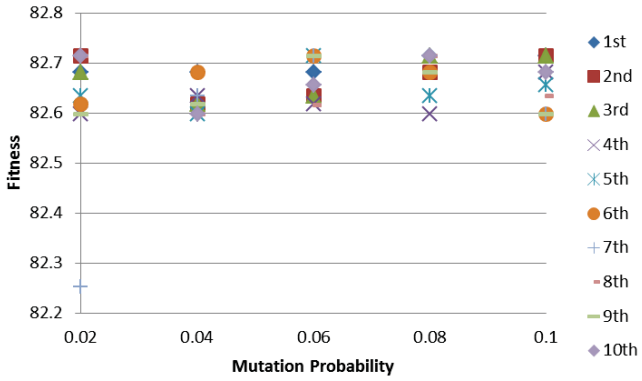

Figure 6: Fitness of the best individuals at the 300th generations with various mutation probability

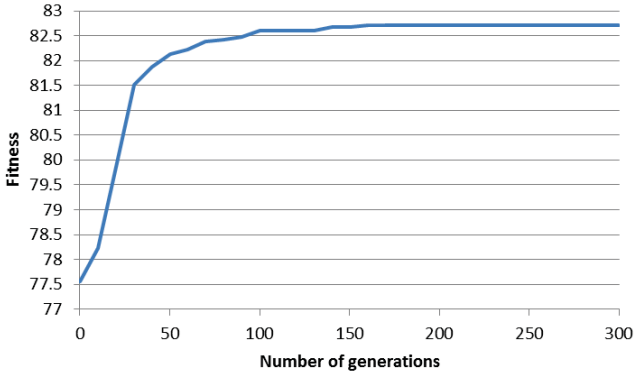

Figure 7: The fitness change by generation in the case of quasi-optimal solution.

achieved 3, 4, 5 and 3 times out of 10 times trials with the crossover probability 0.02, 0.06, 0.08 and 0.1, respectively.

The results of 1) and 2) show that the quasi-optimal solution for the fingerspelling recognition is 82.71%. So the

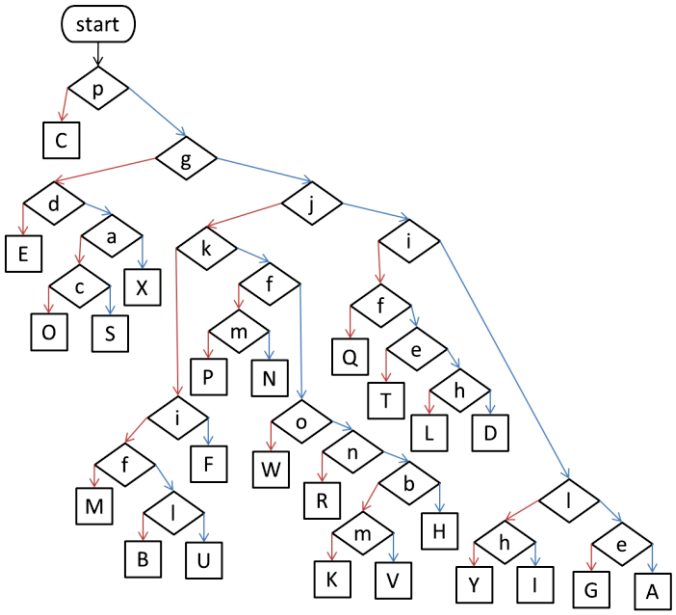

Figure 8 : An example of quasi-optimal solution of decision tree

optimum parameters for the genetic algorithm are the crossover probability 0.8 and the mutation rate 0.08.

3) is performed 100 times using the crossover probability 0.8 and the mutation probability 0.08. As the result, the maximum, average and minimum values of the fitness are 82.71%, 82.66% and 82.25%, respectively. The dispersion is 0.004. Also, 37 trials out of 100 reach the maximum value.

From the above results, by using the optimal parameter (the crossover probability is 0.8 and the mutation probability is 0.08) the quasi-optimal solution are stably obtained. Figure 7 shows the fitness changes by generation in the case of quasi-optimal solution 82.71%. The fitness converges at the 160th generation.

An example of the quasi-optimal solution is "p→g→j→k→i→f→d→o→n→b→l→m→e→h→a→c", and the decision tree is shown in Figure 8. By using the decision tree, it is possible to recognize the 24 letters with average recognition rate of 82.71%. In addition, the decision tree as the quasi-optimal solution can be generated in 3.4 minutes on average.

Although the quasi-optimal decision tree generation requires more than three minutes, the resultant decision tree can be used in any way. The real-time fingerspelling recognition for 24 letters is achieved using the resultant decision tree and Leap Motion Controller without any compute intensive processes such as image processing or neural networks. The real-time fingerspelling recognition is required for human interface by fingerspelling.

6 Conclusions

In this papeer, we propose the fingerspelling recogniton using Leap Motion Controller to give an alternative method of voice input for deaf people. The recogniton target is fingerspelling 24 letters exclude 2 fingerspelling letters that require finger movement.

As conditional branches to be used for the decision tree, we use 16 kinds of conditions that forcuse on the characteristics of hand and finger. By changing the order of the conditional branches, a different decision tree is generated with a different average recogniton rate for the fingerspelling 24 letters. The decision tree is automaticall generated by a Genetic algorithm to obtain quasi-optimal solutions.

We perform several experiments for the application of the Genetic algorithm and we obtain the quasi-optimal solution of the recogniton rate 82.71%. From the above, we validate the proposed method is very effective.

In the future work, it is necessary to include the fingerspelling two letters with movement. We think the decision tree should not be fixed. As a user makes use of the fingerspelling recognition, the recognition rates at conditional branches may change. In this case, some incremental learning mechamism should be performed.

7 References

[1] iOS8 Siri, Apple, https://www.apple.com/jp/ios/siri/ (last access: 2015-04-13)

[2] Google, https://www.google.co.jp/ (last access:2015-04-13)

[3] Marcus V. Lamar, Md. Shoaib Bhuiyan, Akira Iwata. "Hand alphabet recognition using morphological PCA and neural networksNeural Networks"; IJCNN '99. International Joint Conference on, vol.4, 2839-2844,(1999)

[4] Xbox 360 – Kinect, Microsoft, http://www.xbox.com/ja-JP/Kinect (last access: 2015-04-13)

[5] Leap Motion, https://www.leapmotion.com/ (last :access: 2015-02-03)

[6] Mischa Spiegelmock. "Leap Motion Development Essentials"; Packt Publishing (2013)

[7] C. S. Weerasekera, M. H. Jaward, N.Kamrani. "Robust ASL Fingerspelling Recognition Using Local Binary Patterns and Geometric Features"; Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on, 1 – 8, (2013)

[8] Ching-Hua Chuan, Eric Regina, Caroline Guardino. "American Sign Language Recognition Using Leap Motion

Sensor"; Machine Learning and Applications (ICMLA), 2014 13th International Conference on, 541 – 544, (2014-12)

[9] Fingerspellingalphabet.com, http://www.fingerspellingalphabet.com/fingerspelling-chart-print-pdf-download/ (last access:2015-04-13)

[10] David E. Goldberg. "Genetic Algorithms in Search, Optimization, and Machine Learning"; Addison-Wesley Professional (1989)

[11] John J. Grefenstette, Rajeev Gopal, Brian J. Rosmaita, Dirk Van Gucht. "Genetic Algorithms for the Traveling Salesman Problem"; Proceedings of the 1st International Conference on Genetic Algorithms, 160 – 168 (1985)