Thai Sign Language Translation Using Leap Motion Controller

Jirawat Tumsri and Warangkhana Kimpan

***Abstract*— In this article, algorithm for transforming sign language into Thai alphabets is presented to fill the gap in communication between the hearing impairment and normal people. Leap Motion Controller was applied to detect 5-finger-tip position and palm center in the form of X and Y axis. Then, decision tree was created by using the Quadtree technique and the research result on transforming Thai sign language into finite automata was applied to improve algorithm in creating finite automata of Thai alphabet sign language to increase efficiency and speed in processing sign language. The test result shows that it can discriminate 44-Thai alphabet sign language at 72.83% accuracy.**

***Index Terms*— Finite automata, Hearing impairment people, Leap Motion Controller, Quadtree, Thai sign language**

I. Introduction

In daily life, human beings always need interpersonal communication. Communication is, therefore, an essential basic factor to be in a society. Normal people communicate mainly through speaking but the hearing impairment people use the sign language. One of the barriers in communication between the hearing impairment people and the normal people is that the normal people do not understand the sign language thus making a little difference in a sign language expression can completely change the meaning in communication.

The statistics during 2012-2014 indicated that 51.0% of the hearing impairment people need help [1]. The public sector has set the policy in helping and promoting the hearing impairment people to be able to communicate with normal people by developing Thailand Telecommunication Relay Service (TTRS) machine or the communication machine for the hearing impairment people to fill the gap in communication [2]. However, there is still a problem of insufficient TTRS machine service as there are only 120 service points available in various places, moreover, there is an accessibility problem for the hearing impairment people. When using TTRS machine service, the hearing impairment people need provided sign language translators.

Manuscript received December 27, 2016; revised February 11, 2017.
Jirawat Tumsri is with the Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang (KMITL), Chalongkrung Rd., Ladkrabang, Bangkok, 10520 Thailand (e-mail: 57605074@kmitl.ac.th).
Warangkhana Kimpan is with the Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang (KMITL), Chalongkrung Rd., Ladkrabang, Bangkok, 10520 Thailand (e-mail: warangkhana.ki@kmitl.ac.th).

The survey on the performance of public sector during 2012-2014, shows that the average of the hearing impairment people's quality of life is at a stable level (35.3%) and that there are 37.3% of the hearing impairment people who still face problems and difficulties in living. This shows that the service of the public or private sectors is still insufficient for the hearing impairment people to improve their quality of life [1].

According to the problems mentioned above and in order to help hearing impairment people to be able to communicate with normal people, this research focuses in developing computer program that can translate the sign language into Thai alphabets by applying Leap Motion Controller. This equipment detects the fingers and then Quadtree and finite automata for classifying are used in order to transform the position of each finger into a Thai alphabet.

II. Theory and Related Works

A. Thai Sign Language

Communication can be divided into 2 types: verbal communication in which words and alphabets are commonly used in socialization and non-verbal communication in which actions, gestures, tunes, eyesight, objects, signals, environments, and other expressions are used in communication. Showing hand symbols or sign language is a type of non-verbal communication for hearing impairment people.

In 1956, Khun Ying Kamala Krailuek designed the standard Thai sign language (THSL) by applying the fingerspelling principle of American sign language (ASL) in single hand fingerspelling. THSL consists of 44 alphabets and 23 vowels [3] as shown in Fig. 1.

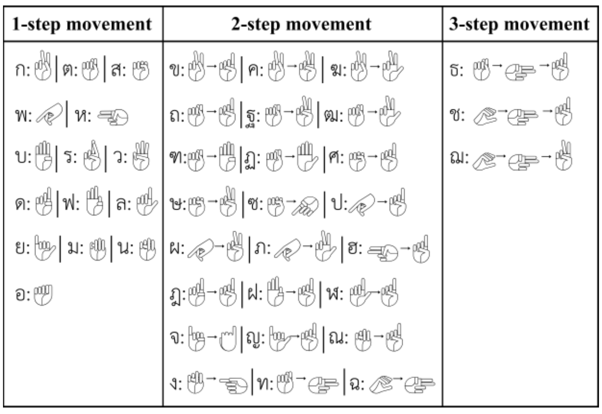


Fig. 1. Thai sign language diagram.
(This figure is derived from the one described in [4] which already had a copyright permission from EECON37.)

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Fig. 1 shows the Thai sign language in which the Thai alphabets are classified into 3 types according to the steps of movement as follows: Type I) 1-step movement in fingerspelling, consisting of 18 alphabets, e.g., ก, ต, ส, and so on. Type II) 2-step movement in fingerspelling, consisting of 23 alphabets, e.g., ข, ค, ฆ, and so on. Type III) 3-step movement in fingerspelling, consisting of 3 alphabets, ธ, ช, and ฌ [4].

B. Quadtree

Quadtree is a type of spatial data structures, similar to Q-tree in object-based image classification for solving the problem of extracting a large amount of data at the same time. Moreover, it can also reduce the number of irrelevant data extractions by continuously dividing the area in a two-dimension square into 4 equal parts in the form of plus sign, resulting in four and sixteen squares respectively. Then the decision tree is created from the results of the square division as shown in Fig. 2.

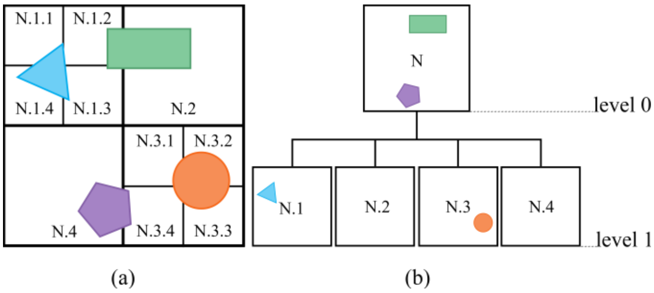


Fig. 2. Quadtree division for storing objects in decision tree.

Fig. 2 (a) shows the square N before dividing. The decision tree creates 1 node in level 0, called "root node", which is the node N as shown in Fig. 2 (b). After dividing the square N into 4 parts and comparing it to the decision tree in Fig. 2 (b), it can be indicated that the node N in level 0 consists of 4 child nodes in level 1 which are N.1, N.2, N.3, and N.4. The division of nodes in level 1 is the last division, therefore the child nodes in this level are called "leaf node". Every node in the decision tree stores the objects which can be the member of only one node. However, in each node, many objects can be stored. The objects can be the members of node N when the criteria are as follows: [5]

1) The objects are placed in N (triangle, rectangle, circle, and pentagon).

2) If the objects are placed in any child nodes of N, they are the members of those child nodes (triangle and circle).

3) If the objects are placed astride child nodes of N, they are the members of node N (rectangle and pentagon).

The Quadtree is mostly used in an object-based image classification, such as finding various positions on the map, providing spatial details of games, and so on. It is good for systematic data storage and reduction of the number of the other irrelevant space examinations.

C. Finite Automata

Finite automata is a model for studying the operation of computer. The overall elements of automata are divided into 3 parts consisting of input alphabet tape, processor, and data storage. The input tape stores alphabets and sends them to the processor, the processor controls internal state that changes according to the input, whereas the storage stores the outcome obtained during the calculation or processing [6].

Finite automata consists of 2 types: deterministic finite automata and non-deterministic finite automata. The definition of deterministic finite automata can be explained by substitution of 5 factors as shown in (1) [7].

$$M = (Q, \Sigma, \delta, q_0, F) \qquad (1)$$

When Q is any state finite sets

Σ is alphabet finite sets

$\delta : Q \times \Sigma \rightarrow Q$ is a transition function

q_0 is a initial state, for $q_0 \in Q$

F is a set of acceptance state for $F \subseteq Q$ and F is a final state assuming that finite automata accepts the input alphabet tape

The transitive diagram definition of finite automata in each diagram is the graph that identifies the direction as the followings:

1) There is Q as a set of all points and there are arrows pointing to the starting point.

2) There are 2 circles enclosing all points in F.

3) There is a line connecting point p to point q with the input symbol a, written as follows: $\delta(p,a) = q$

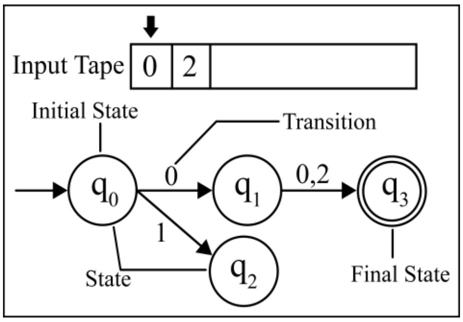


Fig. 3. Diagram of automata elements.

Fig. 3 is a written example of finite automata. It is merely one-way. In each movement of pointer at the input tape, automata will always be in the set of some states. When the reading starts, the input is 0. Therefore, the direction is from the starting state (q_0) to state (q_1). Then, the pointer is moved to the right for 1 alphabet. The process is repeated until all the input is completely read. If the current state is in the set of acceptance state (q_3), it shows that the obtained outcome is the acceptance state.

From (1), it can be substituted as follows: $M = (\{q_0, q_1, q_2, q_3\}, \{0, 1, 2\}, q_0, q_3)$ which can be written in the table format showing the state of receiving the input of 0 and 2 as in Table I:

TABLE I
STATE TRANSITION

State / Event	0	1	2
q_0	q_1	q_2	-
q_1	q_3	-	q_3
q_2	-	-	-
q_3	-	-	-

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

D. Leap Motion Controller

Leap Motion Controller is a sensor as shown in Fig. 4. It is used for detecting the movement position of hands and fingers. It can detect the movement of every knuckle and palm center with the precision at the level of 0.01 mm.

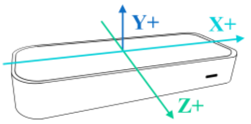


Fig. 4. 3-dimension detecting direction.

The limitation of Leap Motion Controller application is that too much or too little light affects the precision of infrared in the equipment in detecting the objects, and it cannot detect the object within 10 millimeters from the equipment, also the overlapped objects cannot be detected as the only largest one can be detected. Besides, the objects placed too close to each other may be detected as only one object.

E. Related Works

In 2007, Qutaishat Munib, Moussa Habeeb, Bayan Takruri, and Hiba Abed Al-Malik [8] presented the study on the development of automatic system for transforming American sign language by applying image processing to transform image into vector and comparing to the vector using neural network technique in learning the gesture set of sign language, which the rotation of gesture in the image did not affect the data extraction. The result of system test, by using 300 sample images of hand signals, revealed that the system could perceive the sign language gesture with 92.3% accuracy.

In 2008, Suwannee Phitakwinai, Sansanee Auephanwiriyakul, and Nipon Theera-Umpon [9] proposed the study on Thai sign language translation using Fuzzy C-Means (FCM) and Scale Invariant Feature Transform (SIFT), which was the transform of sign language into 15 alphabets and 10 words by using camera as an input in processing. The algorithms used in this study were FCM and SIFT. The experiment result revealed that the system could transform sign language into alphabets and words with the accuracy of 82.19% and 55.08%, respectively. Whereas the nearby neighbor was at 3 and SIFT threshold was 0.7.

In 2014, Wisan Tangwongcharoen and Jirawat Tumsri [4] presented the study on decoding Thai sign language pattern using Quadtree and using Leap Motion Controller in detecting 5-fingertip positions. Those fingertip positions were analyzed by using Quadtree technique. The outcome consisted of the number codes for 44 alphabets, 3 digits for each one.

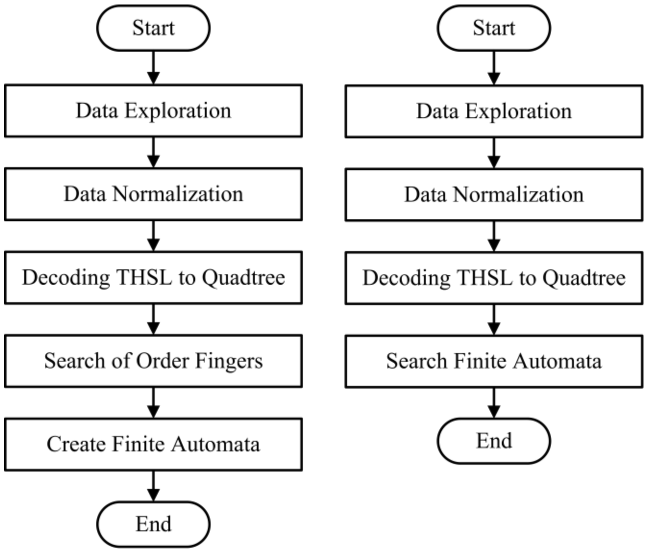
In 2015, Jirawat Tumsri and Warangkhana Kimpan [10] presented the study on code translation from the Thai sign language pattern to finite automata by creating finite automata from 1-step fingerspelling codes of 18 alphabets, one finger for 3-digit code. The obtained result was 1-step fingerspelling finite automata structure.

In this study, Quadtree technique [4] was applied in transforming finger positions into finger codes for 44 alphabets, and finite automata algorithm [10] was improved for supporting 3-step fingerspelling.

III. Proposed Method

A. Overview of the Method

The methods is divided into 2 main parts: creating finite automata prototype for Thai sign language as shown in Fig. 5 (a) and transforming sign language into Thai alphabet pattern as shown in Fig. 5 (b).



(a) (b)

Fig. 5. 2 main parts of overview method.

From Fig. 5 (a) the creation of finite automata for Thai sign language starts by collecting the samples of Thai sign language in the pattern of the position on X and Y axis. Then, the sample data values are adjusted into a proper pattern, then Thai sign language is decoded. The obtained result in each finger is a 3-digit number. Then, the 3-digit code is used to find out finger order and number for creating finite automata prototype for Thai sign language in order to reduce the number of data or fingers. After getting the finger order and number, the finite automata prototype for Thai sign language is created.

Fig. 5 (b) shows the steps in transforming sign language into Thai alphabet pattern. It starts by the user showing the sign language through Leap Motion Controller and then the data are sent to the program. The value of the data, which are in the pattern of positions on X and Y axis, is adjusted to be in a proper pattern. Then, the positions on X and Y axis with the adjusted value are decoded into Thai sign language and the outcome of 1 alphabet is found out from finite automata prototype for Thai sign language.

B. Data Exploration

In order to collect the prototype data or the transformation of sign language into Thai alphabet pattern, the hand gestures must be in the parallel position to Leap Motion Controller. First, hand gesture, open hand or waving hand, must be shown for 3 seconds so that the hand sizes of both the master and examiner can be adjusted. After that the position of X and Y axis on fingertip and palm center are stored for 3 seconds for each gesture after started. The data of positions on X and Y axis on fingertips which derived from Leap Motion Controller, are used in this study as follows:

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

1) The creation of finite automata prototype for Thai sign language consists of positions on X and Y axis of 5 fingertips and on X and Y axis of palm center.

2) For transforming sign language into Thai alphabet pattern, receiving data value of each finger relies on the outcome of finding out finger order and number in the step of creating finite automata prototype for Thai sign language. For instance, the outcome of finding out finger order and number consists of thumb, index finger, and middle finger, while the position of X and Y axis are in the palm center.

C. Data Normalization

The adjustment of data value into the proper pattern is the master's and the examiner's hand size adjustment in order that their hands are in the same size, and determination of position on the same palm center is also done for accuracy in transforming sign language into Thai alphabet pattern.

The equation for finding the distance between 2 points in direct line or Euclidean distance [11] is shown in (2):

$$distance = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \qquad (2)$$

Where $distance$ is the length from middle fingertip to palm center

X_1, Y_1 is X and Y axis on middle fingertip

X_2, Y_2 is X and Y axis on palm center

After the distance between middle fingertip and palm center is obtained, the X and Y axis positions are used in finding out the ratio as shown in (3) and the outcome of (3) is used to adjust the X and the Y axis positions of each finger as shown in (4) and (5) respectively. This adjustment is conducted to solve the problem of different hand sizes of the master and the examiner and reduce the error in transforming sign language into the Thai alphabet pattern.

$$ratio = distance_1 / distance_2 \qquad (3)$$

Where $ratio$ is the ratio between sign language of the master and examination

$distance_1$ is the length from middle fingertip to palm center of the master

$distance_2$ is the length from middle fingertip to palm center of the examination

$$X_{post} = X_{pre} * ratio \qquad (4)$$

$$Y_{post} = Y_{pre} * ratio \qquad (5)$$

Where X_{post}, Y_{post} is X and Y axis positions after adjusting the ratio

X_{pre}, Y_{pre} is X and Y axis positions before adjusting the ratio

$ratio$ is the ratio between sign language of the master and examination

In addition to the adjustment of finger positions, there is also the shift of finger positions presenting the proper Quadtree table, which the position on palm center is always set as a fixed point in referring to other fingers by setting the palm center position on the midpoint of X and Y axis in the Quadtree table. In doing so, it can cover all of the hand even though there is a shift position or changes of the positions of other fingers.

The shift of finger position is done by finding out the distance between the former palm center position (X_1, Y_1) and palm center position after being shifted to the midpoint of X and Y axis (X_2, Y_2) as shown in (2). After finding out the distance of the shift of palm center position, X and Y axis positions of each finger are adjusted as shown in (6) and (7).

$$X_{post} = X_{pre} + distance \qquad (6)$$

$$Y_{post} = Y_{pre} + distance \qquad (7)$$

Where X_{post}, Y_{post} is X and Y axis positions after adjusting the ratio

X_{pre}, Y_{pre} is X and Y axis positions before adjusting the ratio

$distance$ is the length of shifting palm center position toward the midpoint of Quadtree table

The creation of finite automata prototype for THSL starts by finding out the value between middle fingertip and palm center as shown in (2). Then, X axis position is moved as shown in (6) and Y axis position as shown in (7) to transform sign language into the correct Thai alphabet pattern.

In adjusting data value into the proper pattern in the step of transforming sign language into Thai alphabet pattern, the values of X and Y axis positions on middle fingertip and on palm center are used in finding out the distance between 2 points as shown in (2). Then, the ratio of the distance is found out between middle fingertips and palm centers of the master and the examiner as shown in (3). After the ratio is obtained, the value of X axis position is adjusted as shown in (4) and of Y axis position as shown in (5). The next step is to move X axis position as shown in (6) and Y axis position as shown in (7).

D. Decoding THSL to Quadtree

Algorithm for decoding THSL by using Quadtree technique is a method for decoding the gesture of one finger. The input consists of X and Y axis positions on each finger, the number of alphabet codes, and the size of Quadtree square table, whereas the obtained outcome consists of 3-digit number code of each finger that can be transformed into the decision tree pattern as shown in Fig. 6. This result was extended from the one obtained in [4] which already had a copyright permission from EECON37.

E. Search of Order Fingers

Finger order search is a step in creating a finite automata prototype of THSL to reduce data storage and increase processing efficiency in transforming sign language into Thai alphabets. The data on X and Y axis of 5 fingers be entered and the outcome will be obtained in form of the finger order as described in the following algorithm.

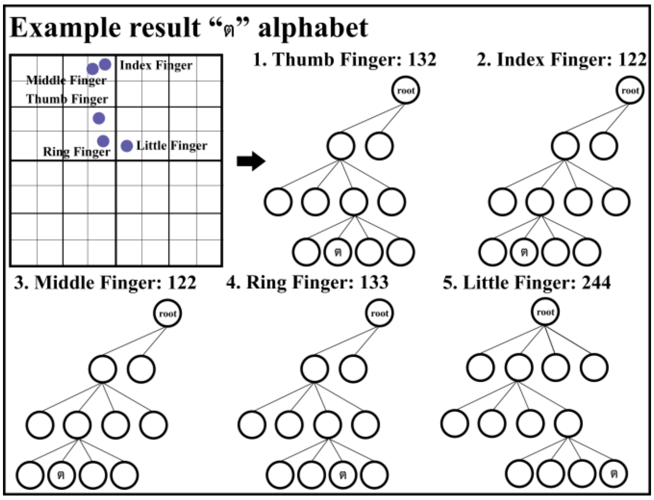


Fig. 6. Example of decision tree prototype for the alphabet “ต” of 5 fingers.

```
Algorithm: Search of Order Fingers
found duplicate = true, number of finger = 5, finger number i = 1
While found duplicate
    Set temp finger number = 1, minimum code = 44
    While temp finger number <= number of finger
        number duplicate finger array = find code of finger duplicate
        If minimum code > number duplicate finger array Then
            minimum code = number duplicate finger array
            finger number i = type of finger
        End if
        Next temp finger number
    End while
    If found duplicate code of finger array Then
      Remove code of finger in temp finger number
        number of finger = number of finger - 1
        Next i
    Else
        found duplicate = false
    End if else
End while
Return finger number
```

The above algorithm is the steps of finger order and number search. First, the first finger order is searched by selecting to store each finger code of duplicate alphabet. For example, the codes on a thumb finger for the first, second, and third alphabets are 244, 123, and 244, respectively. Therefore, 244 is stored as an alphabet code of the thumb finger. Secondly, after all alphabets are completely searched, the algorithm examines whether the initial finger has more finger codes in storage than the current one. If so, the number of duplicated code is stored and the current finger is set as the first finger. Finally, the stored finger codes are examined whether there are any duplicate codes. If there are, the next finger or the second finger is then searched. In the search of the second finger, only the stored duplicate code of the first finger is considered and the outcome of the first finger from the initial search will not be considered. For example, if the outcome of the first finger is the thumb finger, the second round is, therefore, to consider the codes of index finger, middle finger, ring finger, and little finger. This process kept repeating until the duplicate finger code is no longer found or all of 5 fingers are completely considered. For example, the final outcome reveals that there are 3 fingers: the first one is the thumb finger, the second is the index finger, and the third is the middle finger.

F. Create Finite Automata

From (1), the factors of finite automata are set to be the code numbers of sign language in each digit of each finger and final state is the set of one alphabet. Therefore, the alphabet is encoded into sign language. Then, each digit is decoded by starting at the outcome of finger order and number search, which each finger has 3-digit code.

In algorithm for creating finite automata prototype, the input is THSL prototype for decoding 44 alphabets. The outcome is finite automata structure of THSL. The algorithm for creating finite automata prototype is shown below.

```
Algorithm: Create Finite Automata
alphabet = 1, number of alphabet = 44
While alphabet <= number of alphabet
    finger number = 1, current state = initial state
    While finger number <= size finger number
        Read code of finger number 3 integer
        For i = 1 to 3
            Get code of finger number integer i
            Find state code finger at current state
            If not found state code finger Then
                Create state finite automata
            End if
            Next i
        End for
        Next finger number
    End while
    state automata = final state
    Next alphabet
End while
Return finite automata
```

The above algorithm is finite automata creating algorithm. Firstly, the current state pointing at the initial state and receiving 1-alphabet code is set. Then, the 3-digit code is read from the first finger, the first digit code is read, the current state is searched whether there is a movement direction toward another state of the first digit code. If it is not found, the state is created with the movement direction toward another state to be the first digit and the current state is set to point to the created state. The process is repeated until all of three digits are completely done. Secondly, when the state is created for all 3 digits, the code of the next finger is read and repeated the same process as creation steps of automata of the first finger code. Finally, when the creation is completed according to finger order and number for one alphabet, the automata of next alphabets are created until all of 44 alphabets are completely done.

```
Algorithm: Search Finite Automata
current state = first state of THSL finite automata
While EOF of input tape
    found = false
    Read value at current pointer of input tape
    While consider all transition or not found
        If value on transition == value of input tape Then
            Next state of current transition
            found = true
        Else
            Next transition of current state
            End if
            Next i
        End if else
    End while
    Next pointer of input tape
End while
If state of current state == final state Then
    Get alphabet of current state
Return alphabet
```

G. *Search Finite Automata*

The inputs of search finite automata for Thai alphabets are the examiner's and the master's codes of each finger, and code for decoding in THSL pattern. The obtained outcome consists of Thai alphabets which is described in the algorithm of search finite automata.

At the starting, the current state is set to point to the first state. Then, the value is read from the pointer that points to the input tape by considering all directions of transition of current state. If value on transition is equal to value read from input tape, the current state point to transition state in that direction. Then, the pointer of the input tape is moved to the right and the direction of the transition is considered as done before. This process is repeated until the end of the input tape pointer, then this algorithm outcome from the acceptance state of current state is considered. If it is the acceptance state, the alphabet of current state is retrieved. If it is not, it reveals that no Thai alphabet is found.

IV. Experimental results and discussion

From the steps of THSL finite automata creation for 44 alphabets, the outcome from the step of decoding THSL to Quadtree was the code of 3-digit number of each alphabet on each finger. Then, the outcome from the step of decoding THSL to Quadtree was used for the search of finger order which reveals that three fingers namely thumb finger, index finger, and middle finger, can sequentially classify sign language of 44 Thai alphabets. The results of this step are shown in Table II. Finally, these results are used for creation of finite automata.

TABLE II
Result of Decoding THSL to Quadtree Method

Alphabet	Thumb Finger	Index Finger	Middle Finger
ต	132	122	122
พ	422	133	422
บ	422	132	333
ข (ก + 1)	322 + 122	133 + 133	332 + 422
ถ (ต + 1)	132 + 122	122 + 133	122 + 422
ฝ (ฟ + 1)	422 + 122	422 + 133	132 + 422
ธ (ท + 1)	(132 + 422) + 122	(122 + 132) + 133	(122 + 133) + 422
ช (ฉ + 1)	(122 + 422) + 122	(132 + 132) + 133	(332 + 133) + 422
ฌ (ฉ + 2)	(122 + 422) + 422	(132 + 132) + 112	(332 + 133) + 333

Table II shows the code of 3-digit number of each alphabet on each finger. For example, the outcome in the second-movement group of alphabet "ข" was the results from the mixture of alphabet "ก" (thumb: 322, index finger: 133, middle finger: 332) and the number "1" (thumb: 122, index finger: 133, middle finger: 422). Whereas the outcome in the third-movement group of alphabet "ธ" was the results from the mixture of alphabet "ท" in the second movement (the first movement, thumb: 132+422, index finger: 122+132, middle finger: 122+133) and the number "1" (thumb: 122, index finger: 133, middle finger: 422).

On the steps of transforming sign language into Thai alphabet pattern, the accurate outcome in transforming sign language into Thai alphabets is shown in Table III.

TABLE III
Example Result of Search Finite Automata

Alphabet	Accuracy	Alphabet	Accuracy	Alphabet	Accuracy
ต	80.00%	ข	63.33%	ธ	75.56%
พ	73.33%	ถ	80.00%	ช	73.33%
บ	86.67%	ฝ	83.33%	ฌ	66.67%

V. Conclusions

This research presented the transformation of sign language into Thai alphabets to help the hearing impairment people use to communicate with normal people. Two steps were applied in the study: 1) creation of the prototype of finite automata for sign language on 44 Thai alphabets which classified into 3 groups according to the finger movements and 2) transformation of sign language into Thai alphabets by using Quadtree and finite automata. The average score of obtained outcome accuracy in sign language transformation was 72.83%. The alphabets with the highest accuracy score, 86.67% were "พ" and "บ". Whereas the alphabets with the lowest accuracy score, 53.33% were "ย", "ค", and "ฆ".

References

[1] J. Bunyarattanasuntorn, Proceeding Articles in 7th National Academic Conference on the Deft People, "Quality of Life of the Deft: Reflection of Monitoring and Evaluation of the 4th National Development Plan for Quality of Life of The Hearing Impairment People, 2015-2016", vol.2, pp. 33-58, 2015.

[2] Thai Telecommunication Relay Service. TTRS Machine for the hearing impairment people: Reduction of communication gap [Online]. Available: http://ttrs.or.th/index.php/submenu1/news/156-it24hrs-ttrs [Nov 22, 2016].

[3] M. Thammasaeng, "Sign Language," Bangkok: Jong Chareon Printing, Sign Language, vol. 2, 1996.

[4] W. Tangwongcharoen and J. Tumsri, "Decoding Thai Sign Language Pattern using Quad Tree," *in Proc. of the 37th Electrical Engineering Conf. (EECON37)*, Khon Kaen, Thailand, vol. 2, pp. 873-876, 19-3 Nov, 2014.

[5] ajangz. 2015. Quadtree [Online]. Available: http://ajangz.exteen.com/20150528/quad-tree [Nov 10, 2016].

[6] K. Jiaranaithanakit, "Theory of Computation," 2nd ed. Bangkok: Technology Promotion Association (Thailand-Japan), 2014, ch. 2, pp. 20-23.

[7] P. Linz, "An Introduction to formal languages and Automata," 4th ed. London, UK: Jones and Bartlett Publishers, 2006, ch. 1, pp. 26-28.

[8] Q. Munib, M. Habeeb, B. Takruri, and H. A. Al-Malik, "American sign language (ASL) recognition based on Hough transform and neural networks", *Science Direct, Expert Syst. with Applicat*., vol. 32, pp. 24-27, 2007.

[9] S. Phitakwinai, S. Auephanwiriyakul, and N. Theera-Umpon, "Thai Sign Language Translation Using Fuzzy C-Means and Scale Invariant Feature Transform," *ICCSA 2008*, pp. 1107-1119, 30 Jun.-3 Jul. 2008.

[10] J. Tumsri and W. Kimpan, "Code Translation from Thai Sign Language Pattern to Finite Automata," *in Proc. of the 7th Conf. of Electrical Engineering Network of Rajamangala University of Technology 2015 (EENET 2015)*, A-one The Royal Cruise Hotel, Pattaya, Chonburi, Thailand, pp. 125-128, 27-29 May, 2015.

[11] M. M. Deza and E. Deza, "Encyclopedia of Distances," Springer, Sci. Bus. Media, p. 94, 2009.