Low-cost Intelligent Static Gesture Recognition System

Harini Sekar[α], Rajashekar R[ϕ], Gosakan Srinivasan[β], Priyanka Suresh[Σ], Vineeth Vijayaraghavan[Ω]
Research Assistant[αϕ], Undergraduate Student[βΣ], Director - Research and Outreach[Ω]
Solarillion Foundation[αϕΩ], Easwari College of Engineering[β], SSN College of Engineering[Σ]
(harini.sekar.in[α], rajashekar.r.in[ϕ], gosakan.s.in[β], priyanka.suresh.1994[Σ], vineethv[Ω])@ieee.org

***Abstract*—This paper presents prototype implementation of low-cost, open hardware, static - gesture recognition system. The implemented system has three major components: A Glove and Sensor Unit (GSU) – consisting of a pair of gloves embedded with custom made, low-cost flex and contact sensors, a Primary Supporting Hardware (PSH) that maps change in input values from GSU, a Secondary Supporting Hardware (SSH) that processes the input values and recognizes the gesture accurately. When a gesture is signed, the GSU tracks the change in orientation of the fingers, which results in a change in voltage levels of the sensors. This change is mapped by the PSH and passed on to SSH which comprises of two ATmega328P microcontrollers, one connected to each of the glove. The two microcontrollers are connected in a master-slave configuration and communication between them is facilitated through an XBee module. The performance of this gesture recognition system is evaluated using a data set comprising of 36 unique gestures. These gestures represent a total of 120 gestures that include all gestures across five globally used sign languages. A gesture recognition engine that resides in the master microcontroller processes the input and identifies the gesture. The gesture recognition engine comprises of a two stage selection-elimination embedded intelligence algorithm that is used to enhance the system efficiency from 83.1% to 94.5% without any additional hardware. The cost of the system is USD 30, which the authors believe on commercialization, could be brought under USD 9.**

***Keywords* — calibration; gesture recognition; master-slave; microcontrollers; open source hardware; sensors; sign language; XBee;**

I. Introduction

As on March 2015, over five percent of the world's population, which accounts for 360 million people are estimated to have disabling hearing loss [1]. Data from the World Health Organization (WHO) proclaims that majority of people with disabling hearing loss live in low and middle-income countries. About 70 million deaf people use sign language as their first language [2]. Sign language is also utilized by the audio-vocally impaired people for their communication. Just like other native languages, a variety of sign languages exist in different countries. One of the major challenges for people with hearing impairment is their fettered communication with the outside world. Limited access to technology due to their inability to communicate has a significant impact on their everyday life. Research is being done on several aspects to enhance their communication with the external world. One major area of research is gesture based recognition systems, which first came into existence with the advent of Sayre Glove [3]. Ever since, there has been development of significant devices and models that accurately interpret and translate the gestures of various sign languages.

The existing gesture based recognition devices utilize components such as accelerometers, gyroscopes and high-end sensors, which significantly increase the overall cost of the end system. The major demerit of such systems is their complexity and high-cost. Owing to the fact that majority of audio-vocally impaired people belong to the middle and bottom of the economic pyramid, such systems are not easily affordable by them and hence not viable. This brings about a need to develop low cost gesture recognition systems that identifies static gestures from various sign languages.

In this paper, a low-cost, open hardware, generic static gesture recognition system is implemented. The system is built using off-the-shelf components to facilitate easy implementation and lend itself to scale. It consists of a pair of gloves, which comprises of custom made, flex and contact sensors embedded on it. Commercially available flex sensors are unviable for this implementation as each such sensor typically retails at about USD 8. These custom sensors were built in the lab and cost about USD 0.02, a sensor. The glove is connected to PSH that maps the gesture input to corresponding voltage levels. These voltage levels are processed by the two ATmega328P microcontrollers, which is part of the SSH. The gesture recognition engine that runs on one of the two microcontrollers performs the identification and the result is displayed in the LCD. The proof of concept system has been built at a cost of USD 30 and can be realized under USD 9 upon commercialization.

II. Related Work

For several years there has been advent of new gesture based recognition systems for translating and interpreting sign languages. Wang Jingqiu et al [4], Celestine Preetham et al [5], Tushar Chouhan et al [6], in their systems have developed data gloves constructed using bend/flex sensors and accelerometers for interpretation of sign languages. The usage of accelerometers in the design of the glove increases the cost of the overall system. Others [6], have made use of Hall Effect sensors which add to the cost of the overall system. The prototype system proposed in this paper uses a pair of gloves embedded with custom made flex and contact sensors to minimize the cost of the overall system.

978-1-4673-9519-9/16/$31.00 ©2016 IEEE

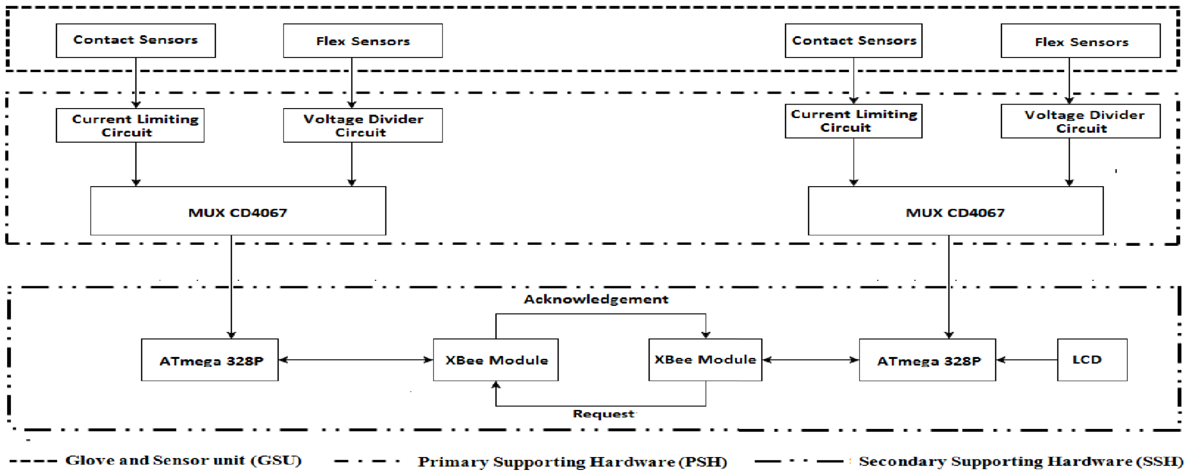

Fig. 1: System Block Diagram

Duy Bui et al [7], in their paper make use of MEMS accelerometers for recognition of Vietnamese sign language. Jong-Sung Kim et al. [8], in their paper have developed a dynamic gesture recognition system for identifying the vocabulary of the Korean Sign language. Tan Tian Swee et al [9], developed a gesture recognition system that identifies 25 gestures of the Bahasa Isyarat Malaysia sign language. In the field of gesture based recognition, systems that recognize a particular sign language or a predefined vocabulary set have been evolving since the advent of Sayre glove in 1977. The system implemented in the paper is generalized to accurately identify static gestures from five of the globally used sign languages.

III. Problem Statement

The hearing impaired face difficulties in communicating with their surroundings. A lot of research work is being done for their improved interaction. Gesture based recognition systems that accurately translate sign language is one such area of research. A number of such systems developed are not economically viable for the hearing impaired as research shows that the majority of them belong to low income groups. Hence there is a need to develop low cost affordable systems that lend itself to scale.

Lot of existing systems translate a particular sign language with a limited gesture set. This necessitates a need to develop a system that is generic in nature and can identify a number of sign languages. This paper expounds the prototype implementation of a low cost generic gesture recognition system that broadly addresses the aforementioned needs.

IV. Methodology

The block diagram of the gesture recognition system is shown in figure 1. The gesture recognition system consists of three major components: the Glove and Sensor Unit (GSU), the Primary Supporting Hardware (PSH) and the Secondary Supporting Hardware (SSH). The GSU comprises of a pair of gloves embedded with flex and contact sensors. When the user signs a gesture, the change in orientation of the fingers is recognised and the corresponding values are passed on to the PSH. The PSH consists of supporting circuits, namely, the current limiting resistor circuit connected to the contact sensors and voltage divider circuit connected to the flex sensors. These supporting circuits map the change in voltage levels from the sensors. The mapped voltage levels are passed on to the SSH from the PSH through the multiplexer, MUX CD4067. The SSH consists of a pair of ATmega328P microcontrollers, each connected to a glove. One of the microcontrollers is denoted as the master microcontroller and other, the slave. The communication between the two microcontrollers happens through an XBee module. Once the master microcontroller receives the input from the glove connected to it, it requests the slave microcontroller for its corresponding input value. The slave acknowledges this request and transmits the equivalent value of the input obtained by it to the master microcontroller. On receiving this input, the gesture recognition engine in the master microcontroller processes the values obtained and identifies the gesture. The identified gesture is then displayed in the Liquid Crystal Display (LCD).

A. *Glove and sensor unit [GSU]*

The GSU consists of a pair of gloves embedded with low-cost custom made bidirectional flex and contact sensors that track the change in orientation of the fingers when the user signs a gesture [10]. The cost of each flex sensor is USD 0.02.

1) Positioning of sensors: The positioning of sensors in the glove plays a crucial role in accurate gesture recognition. It was observed that a total of 17 unique sensor positions were required to map all gestures in the 5 globally used sign languages as explained in section D. Accordingly, the gesture recognition glove along with the embedded sensors was designed. The design consists of 9 flex sensors and 8 contact sensors on each glove. The positioning of the sensors is shown in figure 2.

The flex sensors are classified into 2 types based on their positioning on the glove. The flex sensors F_1 to F_5 used to capture the changes in orientation on top of the glove are termed as ‘outer flex sensors’ and the flex sensors F_6 to F_9

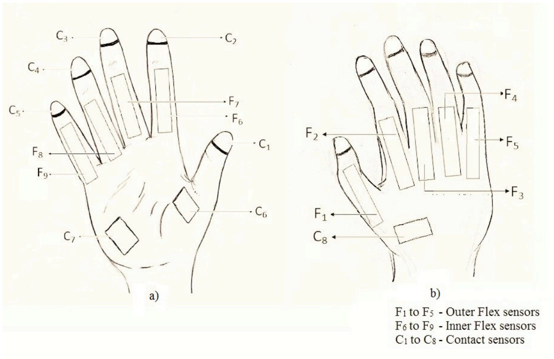

Fig. 2: Gesture Recognition Glove – Flex and Contact sensor postioning – a) Bottom view b) Top view

used to capture the changes in orientation beneath the glove are termed as 'inner flex sensors'. A contact is detected if two or more contact sensors come in contact with each other. If two or more contact sensors of the same hand come in contact with each other, it is termed as 'same hand contact' whereas, if they belong to both the hands it is termed as 'opposite hand contact'.

2) Calibration:

a) Flex sensors: The flex sensors were calibrated to detect the change in orientation due to the signing of a gesture. This change in orientation of the flex sensor is referred to as a bend. The calibration of flex sensors was performed using a calibration program loaded in the master microcontroller during the system design. The analog output from the flex sensors was converted to its equivalent digital value (0-1023) by a 10 bit ADC. Each value from the sensor is an average of 1000 readings sampled at a rate of 0.0625 μs.

TABLE I. INFLUENCE ON FLEX SENSOR BY ADJACENT FINGER BEND

Flex sensor (F_n)	Base value $(T_{Base})_n$	Maximum Value $(T_{Max})_n$	Maximum influence by (n-1)th finger $(T_{Base}+\Delta T_n)_{n-1}$	Maximum influence by (n+1)th finger $(T_{Base}+\Delta T_n)_{n+1}$
F1	512	610	NA	512
F2	512	600	512	530
F3	512	640	535	520
F4	512	610	550	530
F5	512	570	520	NA
F6	512	680	512	512
F7	512	710	512	512
F8	512	660	512	512
F9	512	640	512	NA

The output of each flex sensor F_n during no bend condition is the initial base threshold value $(T_{Base})_n$. F_n is subjected to its maximum bend and the corresponding maximum value is denoted as T_{Max}. It was observed from table I that $(T_{Base})_n$ of the outer flex sensors (F_2 to F_5) increased due to its adjacent bends. The maximum deviation (ΔT_n) from the base value of F_n due to the bending caused by the sensors (F_{n-1}) (left) and (F_{n+1}) (right) are denoted as $[(T_{Base}+\Delta T_n)_{n-1}]$ and $[(T_{Base}+\Delta T_n)_{n+1}]$ respectively and is shown in table I.

The base threshold values are recalibrated to new threshold values $(T_{Base})_n$ using the formula in equation 1 to account for the influence on the base threshold value by its adjacent bends and is shown in table II. A tolerance factor (τ) is added to $T_{Base,new}$ to arrive at the calculated threshold $(T_{cal})_n$ as denoted in equation 2. The tolerance factor is set to 20% of the difference between T_{Max} and $T_{Base,new}$ and the resulting final threshold is shown in table II.

$$(T_{Base,new})_n = max[(T_{Base}+\Delta T_n)_{n-1}, (T_{Base}+\Delta T_n)_{n+1}] \quad (1)$$

$$(T_{cal})_n = (T_{Base,new})_n + \tau \quad (2)$$

TABLE II. FLEX SENSOR - THRESHOLD VALUES

F_n	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9
$(T_{Base})_n$	512	512	512	512	512	512	512	512	512
$(T_{Base,new})_n$	512	530	535	550	520	512	512	512	512
$(T_{cal})_n$	537	548	561	533	538	548	559	556	538

b) Contact sensors: In order to identify the types of contacts present in the gesture, a separate calibration procedure was performed for the contact sensors. The output voltage range of the contact sensors for various types of contacts as explained in section B, are converted to their digital equivalent value (0-1023). A tolerance of $\pm$ 0.05V was factored during calibration. The result of calibration for the contact sensors is tabulated in table III.

TABLE III. CONTACT SENSOR THRESHOLD VALUES

	Reference Voltage		Same hand contact		Opposite hand contact	
Contact Sensor (C_n)	Hand 1	Hand 2	Hand 1	Hand 2	Hand 1	Hand 2
C_1	1023	1023	-	-	1023	1023
C_2 to C_8	368	0	489 to 705	1023	51 to 304	51 to 304

B. Primary Supporting Hardware [PSH]

The PSH consists of current limiting resistor circuit and voltage divider circuit, which receives values from contact sensors and flex sensors respectively as shown in figure 3. The obtained values are then multiplexed using a multiplexer MUX CD4067. The output of the multiplexer is then passed on to the microcontroller in the SSH for further processing.

1) Design of current limiting resistor circuit: The voltage for the contact sensor C_1 is used as reference and is set at 5V. The sensors C_2 to C_8 on both the hands were initially assigned 0V (V_2). This hardware design resulted in production of overlapping sensor output values for different orientations, thus making it difficult to differentiate between gestures containing same hand and opposite hand contacts. In order to identify the type of contact and hence the gesture, a design enhancement was made in the circuit, which involved setting a threshold for the contact sensors in one of the hands. This threshold was

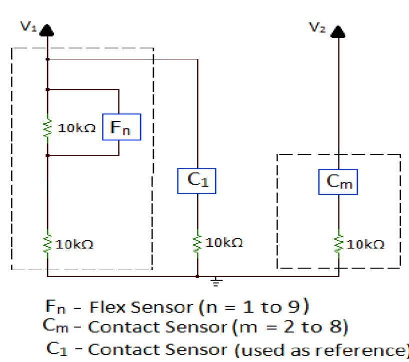

Fig. 3: Primary Supporting Hardware

empirically arrived at by varying the voltage from 0V to 5V with 0.2 V increments. It was observed that at 1.8V, gestures could uniquely be identified without any overlap in values. The output voltage of contact sensors for various types of contact is shown in table IV.

TABLE IV. VOLTAGE RANGE FOR CONTACT SENSORS

Reference Voltage			Same hand contact		Opposite hand contact	
Contact Sensor (C_n)	Hand 1	Hand 2	Hand 1	Hand 2	Hand 1	Hand 2
C_1	5V	5V	-	-	5V	5V
C_2 to C_8	1.8V	0V	2.44V to 3.4V	5V	0.3V to 1.44V	0.3V to 1.44V

2) Design of voltage divider circuit: The voltage divider circuit comprises of two 10KΩ resistors connected in series. The circuit is connected to a constant supply of 5V (V_1). The flex sensors are connected across one of the resistors. During the signing of a gesture, a change in orientation causes a change in the resistance value of the flex sensor. This is realized as change in voltage across the flex sensors.

C. Secondary Supporting Hardware [SSH]

The SSH consist of an ATmega328P microcontroller connected with an XBee, one for each glove respectively. The controllers receives the values from the respective multiplexers and processes them. The master microcontroller requests the slave for its value through an XBee communication channel. The gesture recognition engine then processes the values and recognizes the gesture based on the mode of operation. The result is displayed in a Liquid Crystal Display.

D. Gesture Set Formation

The gesture recognition system is designed to recognize the static gestures across various sign languages. In order to validate the performance of the system, five different sign languages were chosen. They were Australian Sign Language, British Sign Language, Indian Sign Language, New Zealand Sign Language and Standard Sign Language. The choice of sign languages were made based on its usage pattern globally. The five sign languages comprised of 120 static gestures. It was observed that these 120 gestures comprised of 36 unique gestures (G1 to G36) which formed the master set for the gesture recognition system.

1) Bitmap template: The output of each sensor while signing a gesture has a binary value assigned by the software comparator based on its threshold. This binary value stored is represented as a bit map template shown in figure 4. A gesture has two bitmap templates denoting values from each of the hands, uniquely. Each of the flex sensors are assigned one bit (0 or 1) as they comprise of only two possible states (no bend or bend), whereas the contact sensors are assigned two bits (00, 01 or 11) in order to differentiate between the three possible states: no contact, opposite hand contact and same hand contact respectively. No bits were assigned to contact sensor C_1 as it is considered the reference.

TABLE V. GESTURE LOOKUP TABLE

Gesture	Equivalent Decimal		Gesture	Equivalent Decimal	
	Hand 1	Hand 2		Hand 1	Hand 2
G1	989	0	G19	10752	10752
G2	990	514	G20	991	1024
G3	508	0	G21	989	32768
G4	1020	989	G22	989	989
G5	989	514	G23	991	1048576
G6	2975	2975	G24	10752	524288
G7	510	510	G25	514	1538
G8	989	8200	G26	991	991
G9	509	477	G27	477	408
G10	991	524288	G28	508	477
G11	11009	524288	G29	477	0
G12	2945	524288	G30	476	0
G13	989	8192	G31	1538	0
G14	988	988	G32	2945	991
G15	989	2016	G33	508	508
G16	1021	524288	G34	480	0
G17	33007	32768	G35	1	1
G18	3065	524288	G36	992	989

2) Gesture lookup table: The gesture lookup table contains 36 gestures which form the master set for the gesture recognition system as mentioned above. Each gesture comprises of two bitmap template values, one for each hand. The bitmap template values of each hand are converted to their decimal equivalent and stored in the gesture lookup table as shown in table V.

TABLE VI. PSEUDOCODE FOR INTELLIGENT ALGORITHM

1.Obtain the decimal equivalents of both the hands while signing the gesture.

2.Calculate the binary equivalent of each hand and store them in two separate bitmap templates.

3.Compare B_2 and B_3 of both hands in the signed gesture with the corresponding B_2 and B_3 of the gestures from the lookup table.

4.Form a subset of the gestures (S) which satisfies the condition mentioned in step 3.

5.If S consists of single gesture, then display the gesture through LCD.

6.Else, compare B_1 of each hand of the signed gesture, with corresponding B_1's in subset S.

7.Find the gesture (s) with the least number of mismatches in bits of B_1.

8.If more than one gesture satisfies this condition, then display 'Gesture not found'.

9.Else display the gesture through LCD.

E. Design of two-stage selection-elimination intelligence algorithm

The intelligence algorithm is used by the gesture recognition engine when the system is in the Intelligent Operating Mode (IOM) as defined in section F. The algorithm was implemented in a two-stage selection-elimination process, taking into consideration, observations during hardware testing and calibration.

Signed gestures from each hand are subdivided into three blocks B_1, B_2 and B_3 as indicated in figure 4, which denote the bits corresponding the outer flex sensors, inner flex sensors and contact sensors respectively. In the first stage (selection stage), B_2 (inner flex sensor bits) and B_3 (contact sensor

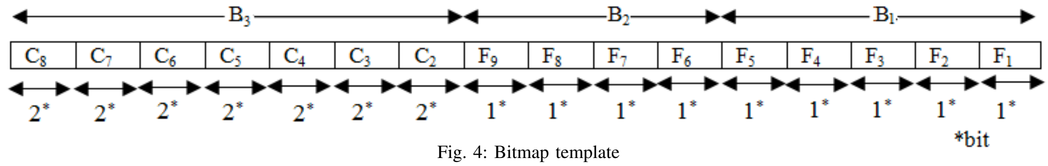

Fig. 4: Bitmap template

TABLE VII. SYSTEM EFFICIENCY ACROSS INDIVIDUAL GESTURES - DOM, IOM

Gesture		G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18
Efficiency %	DOM	88	80	83	79	87	77	78	90	94	91	87	85	87	81	70	83	64	91
	IOM - I	96	89	0	0	98	77	0	98	0	100	91	100	97	0	88	95	84	100
	IOM - II	96	89	100	97	98	77	100	98	100	100	91	100	97	89	88	95	84	100
Gesture		G19	G20	G21	G22	G23	G24	G25	G26	G27	G28	G29	G30	G31	G32	G33	G34	G35	G36
Efficiency %	DOM	91	81	81	81	80	97	80	87	83	94	76	96	84	76	74	92	72	72
	IOM - I	98	100	100	0	97	97	94	0	100	100	0	0	0	89	74	0	0	0
	IOM - II	98	100	100	81	97	97	94	100	100	100	95	100	100	89	74	92	90	97

bits) of the signed gesture are matched correspondingly with gestures in the gesture lookup table. The resulting matches in gestures of this stage (IOM-I) are stored in a selection table. This was based on the observation that less than 19% of the errors were contributed by B_2 and B_3, or a combination thereof. The final signed gesture was obtained in the second stage (IOM-II) as a process of elimination, from the selection table. In the elimination process, B_1 of the signed gesture is compared with corresponding B_1's of gestures in the selection table. The engine intelligently performs closest match between the two B_1's being compared, by finding the mismatches in the corresponding bits of B_1. The gestures with the least number of mismatches are stored in an elimination table. If the result in the elimination table is a unique gesture, the signed gesture is identified. However, if the elimination table results in likelihood of more than one gesture, the gesture signed by the user is unidentified. The pseudocode for this algorithm is shown in table VI.

F. Gesture Recognition Engine

The gesture recognition engines runs on the master microcontroller and is used to perform gesture identification. The gesture recognition engine operates in two modes - Default Operating Mode (DOM) and Intelligent Operating Mode (IOM).

1) Default Operating Mode (DOM): In DOM, a gesture is recognized when an exact match between the B_1, B_2, and B_3 of the signed gesture is obtained with one of the reference gestures in the gesture look up table. Mismatch in any one of the blocks results in the gesture being unidentified. It was observed that the errors were due to mismatches in detection of flex and contact sensor's orientation. However 81% of the errors were due to mismatches of outer flex sensors, as seen in figure 5. To enhance the system efficiency and system performance, an intelligent operating mode was devised.

2) Intelligent Operating Mode (IOM): System efficiency enhancement is done by adding intelligence to the existing gesture recognition engine, without the addition of any external hardware, as explained in Section E.

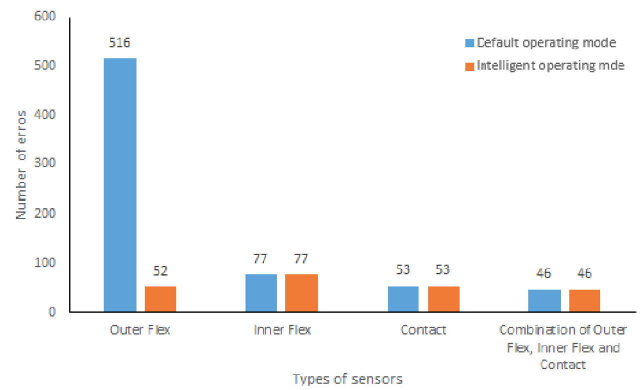

Fig. 5: Error Distribution

G. System efficiency and results

1) Overall efficiency of gesture recognition system: Each of the 36 gestures from the master set was signed 100 times for a total of 3600 samples. This was performed by a single operator with right hand as the dominant hand. The efficiency of the system was calculated in the two operating modes. The overall efficiency of the system was found to be 83.1% in DOM. This is higher than the efficiency that the system achieved in IOM-I (60.05%) and lower than system efficiency in IOM-II (94.5%).

2) System efficiency across individual gestures: The efficiency of individual gestures varied from 64% (G17) to 97% (G24) in DOM while it varied from 74% (G33) to 100% (G3, G7, G9, G10, G12, G18, G20, G21, G26, G27, G28, G30 and G31) in IOM-II. It is observed that 23 gestures performed at their final efficiency during IOM-I while 13 gestures were undetected as shown in table VII. Table VIII illustrates the efficiency improvement for each of the individual gestures in the IOM over the DOM. No gesture performed with an efficiency lower than that of the DOM.

3) System efficiency across multiple data sets: In order to check the consistency of the system efficiency in DOM and IOM, the system was operated in both the modes for ten data sets. Each data set consists of 1000 gesture samples that was synthesized from the master data set. The system was operated in both modes. The overall efficiency of the system was found to be 82.9% across the ten data sets in DOM which was higher

TABLE VIII. EFFICIENCY IMPROVEMENT OF INDIVIDUAL GESTURES

Gesture	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18
%Improvement	9.09	11.25	20.48	22.78	12.64	0	28.20	8.88	6.38	9.89	4.59	17.64	11.49	9.87	25.71	14.45	31.25	9.89
Gesture	G19	G20	G21	G22	G23	G24	G25	G26	G27	G28	G29	G30	G31	G32	G33	G34	G35	G36
%Improvement	7.69	23.45	23.45	0	21.25	0	15.5	14.94	20.48	6.38	25	4.16	19.04	17.10	0	0	25	34.72

than the system's efficiency in IOM-I (60.5%) and lower than the efficiency in IOM-II (94.7%) which is comparable to the overall system efficiencies observed in section G-I. Figure 6 shows the efficiency of each of the modes across the ten data sets.

Fig. 6: System Performance - DOM, IOM

TABLE IX. SYSTEM EFFICIENCY ACROSS MASTER & MULTIPLE DATA SETS

Description	η_{sys}	$(\eta_{(S1-S10)})_{avg}$	$(\eta_{(S1-S10)})_{max}$	$(\eta_{(S1-S10)})_{min}$
DOM	83.11	82.97	84.4	81.6
IOM-I	60.05	60.49	61.8	59
IOM-II	94.5	94.7	95.7	92.9

Conclusion

The overall system efficiency (η_{sys}) for the master data set, the average ($\eta_{(S1-S10)avg}$), maximum ($\eta_{(S1-S10)max}$) and minimum ($\eta_{(S1-S10)min}$) efficiency across ten data sets for the two operating modes are tabulated in table IX. It is observed that there is a very high correlation between η_{sys} and $\eta_{(S1-S10)avg}$ in DOM, IOM-I and IOM-II operating modes thereby validating the robustness of the system across data sets. Further the system demonstrates less than 2% deviation in efficiency from η_{sys} and $\eta_{(S1-S10)avg}$ across all ten data sets in the DOM mode. This deviation was found to be less than 3% in IOM-I and less than 1.9% in IOM-II displaying consistent performance. Thus a low-cost, open hardware, two stage selection-elimination intelligence based embedded gesture recognition system was developed using off-the-shelf components for under USD 30 at laboratory conditions. The system design and selection lends itself to easy commercialization and scale upon which, an expected production cost of USD 9 can be achieved.

Acknowledgement

The authors would like to acknowledge Solarillion Foundation for its support and funding of the research work carried out for the design and implementation of the gesture recognition system.

References

[1] Fact Sheet No.300, "Deafness and Hearing Loss", World Health Organization (WHO), March 2015, [Online]
Available : http://www.who.int/mediacentre/factsheets/fs300/en/

[2] "Sign Language",wfdeaf.org [Online]
Available : http://wfdeaf.org/human-rights/crpd/sign-language

[3] Dipietro, L.; Sabatini, A.M.; Dario, P., "A Survey of Glove-Based Systems and Their Applications," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , vol.38, no.4, pp.461,482, July 2008

[4] Wang Jingqiu; Zhang Ting, "An ARM-based embedded gesture recognition system using a data glove," in Control and Decision Conference (2014 CCDC), The 26th Chinese , vol., no., pp.1580-1584, May 31 2014-June 2 2014

[5] Preetham, C.; Ramakrishnan, G.; Kumar, S.; Tamse, A.; Krishnapura, N., "Hand Talk-Implementation of a Gesture Recognizing Glove," India Educators' Conference (TIIEC), 2013 Texas Instruments , vol., no., pp.328,331, 4-6 April 2013

[6] Chouhan, T.; Panse, A.; Voona, A.K.; Sameer, S.M., "Smart glove with gesture recognition ability for the hearing and speech impaired," in Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS), 2014 IEEE , vol., no., pp.105-110, 26-27 Sept. 2014

[7] Bui, T.D.; Nguyen, L.T., "Recognizing Postures in Vietnamese Sign Language With MEMS Accelerometers," Sensors Journal, IEEE , vol.7, no.5, pp.707,712, May 2007

[8] Jong-Sung Kim; Won Jang; Zeungnam Bien, "A dynamic gesture recognition system for the Korean sign language (KSL)," in Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on , vol.26, no.2, pp.354-359, Apr 1996

[9] Swee, T.T.; Salleh, S.-H.; Ariff, A.K.; Chee-Ming Ting; Siew Kean Seng; Leong Seng Huat, "Malay Sign Language Gesture Recognition system," inIntelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on , vol., no., pp.982-985, 25-28 Nov. 2007

[10] Rishikanth, C.; Sekar, H.; Rajagopal, G.; Rajesh, R.; Vijayaraghavan, V., "Low-cost intelligent gesture recognition engine for audio-vocally impaired individuals," Global Humanitarian Technology Conference (GHTC), 2014 IEEE , vol., no., pp.628,634, 10-13 Oct. 2014