An assistive interpreter tool using glove-based hand gesture recognition

Péter MÁTÉTELKI, Máté PATAKI, Sándor TURBUCZ, László KOVÁCS
Department of Distributed Systems
MTA SZTAKI, Institute for Computer Science and Control, Hungarian Academy of Sciences
Budapest, Hungary
{peter.matetelki, mate.pataki, sandor.turbucz, laszlo.kovacs}@sztaki.mta.hu

Abstract—An assistive tool (InterpreterGlove) for hearing- and speech impaired people is created, enabling them to easily communicate with the non-disabled using hand gestures and sign language. An integrated hardware and software solution is built to improve their standard of living, consisting of sensor network based motion-capture gloves, a low-level signal processing unit and a mobile application for high-level natural language processing. This paper introduces the overall system architecture and describes our automatic sign language interpreter software solution that processes the gesture descriptor stream of the motion-capture gloves, produces understandable text and reads it out as audible speech. The main logic of our automatic sign language interpreter consists of two algorithms: sign descriptor stream segmentation and text auto-correction. The software architecture of this time-sensitive complex application and the semantics of the developed hand gesture descriptor are described. We also present how the beta tester's feedback from the deaf community influenced our work and achievements.

Keywords—assistive technology; hearing impaired; social integration; gesture recognition; gesture descriptor; interpreter gloves; sign language; natural language processing; signal processing; Dactyl; fingerspelling

I. INTRODUCTION

Hearing- and speech-impaired people are not able to verbally communicate. They "speak" their own native language: the sign language. Like any other language, it can be learned by anyone, but is usually spoken only by the target group. Today sign language speakers need a human sign language interpreter to overcome the communication barriers with non-disabled people.

Our assistive tool improves life for hearing- and speech-impaired people by enabling them to easily get in touch with the non-disabled using hand gestures. The project achieves this by creating a hardware-software ecosystem consisting of wearable motion-capture gloves and a software solution for hand gesture recognition and text processing. This integrated tool operates as a simultaneous interpreter helping the disabled to communicate using the Dactyl international sign language – also referred to as fingerspelling – which includes 26 one-handed signs, one for each letter of the English alphabet.

The InterpreterGlove reads out the signed text loud, so the disabled and hearing-abled can fluently communicate. Deaf-mute people's communication potential is highly expanded by this device. The InterpreterGlove advances their social integration as they are enabled to express themselves to non sign language speakers and opens up possibilities like shopping, personal banking, taking part in education and beyond. It may also play a significant role in boosting the employment possibilities of the speech- and hearing-impaired people.

This paper concentrates on the application and high-level software aspects of the InterpreterGlove and does not discuss the hardware and low-level software characteristics in detail.

After the Related Work section, we give an overview of the system by describing the concept. Our custom developed Hagdil (HAnd Gesture DescrIption Language) gesture descriptor is presented next, followed by the algorithm descriptions. Evaluations and future work plans are described in the last sections.

II. RELATED WORK

In the related literature two major types of gesture recognition are presented: optical- and motion sensor-based detection.

Taiwoo Park et al. [1] describe a hand worn sensory system for capturing hand orientation. The study focuses on energy efficiency but the underlying system architecture and signal segmentation mechanisms bring up problems that are similar to our challenges. Similarly, In-Kwon Park et al. [2] worked out a glove based gesture recognizer system. Unlike our customizable InterpreterGlove, they used flex sensors to observe the deflection angles of the fingers and only recognized 17 distinct gestures.

Yun Li et al. [3] built a system to recognize the Chinese sign language based on a single accelerometer and EMG sensors (ElectroMyoGraphy evaluates the electrical activity produced by skeletal muscles). Their work implies that the recognition is based mainly on the hands' position, while the fingers only play a subsidiary role in the process.

Seungki Min et al. [4] describe a project with objectives – and obstacles – relevant to ours: to recognize Korean finger spelling. Analog flex sensors' and orientation sensors' measured data was used to classify the hand gestures by absolute orientation. They preferred the K-mean clustering algorithm instead of the much slower HMM (Hidden Markov Model) calculations and report about an 80% recognition rate (after user calibration). Cemil Oz and Ming C. Leu [5] used

The InterpreterGlove (Jelnyelvi tolmácskesztyű fejlesztése. KMR_12-1-2012-0024) project is performed as a consortium of MTA SZTAKI and Euronet Magyarország Informatikai Zrt. and supported by the Hungarian Government, managed by the National Development Agency, and financed by the Research and Technology Innovation Fund.

Postal address of authors: MTA SZTAKI DSD, H-1111 Budapest, Lágymányosi u. 11. Hungary

2014 IEEE Canada International Humanitarian Technology Conference - (IHTC)
978-1-4799-3996-1/14/$31.00 ©2014 IEEE

Cyber Gloves to detect finger positions and Flock of Birds to sense hand orientation to recognize the American finger spelling gestures. ANN (Artificial Neural Networks) algorithms were used on the software side – also considered in our project but rejected because of the need of intensive training.

Farid Parvini and Cyrus Shahabi [6] describe a custom algorithm to recognize the static finger spelling gestures using Cyber Gloves. Their algorithm uses biomechanical validation to check finger positions. 75% recognition rate was reached without user calibration prior to signing, which would not have been possible using ANN. In another work Farid Parvini et al. [7] created a comparative study of ANN-based and biomechanical characteristics based solutions and state that the latter is more efficient. Following an exhaustive evaluation and planning phase we also decided to take a similar path with InterpreterGlove's Hagdil gesture descriptor and recognition solution, based on biomechanical approach.

Susanna Ricco and Carlo Tomasi [8] report about a novel approach to fingerspelling recognition through classification of letter-to-letter transitions instead of segmenting letter states. Disadvantage of the approach from the computation side is that the domain cardinality expands quadratically. Also, users need to train each sign-pair transition when adding a custom gesture.

Kinematic analysis of sign language - spoken language translation was executed by Chemuttaai Koech [9] on the same hardware setup as [5] to specify the segmentation theory of the signs. W.W. Kong and Surendra Ranganath [10] investigated the hand trajectories for sign language and created segmentation algorithms using Bayes-detectors to identify segmentation points. Hunwei Han et al. [11] also used visual detection solutions for hand motion analysis based on the hand kinematics. They state that hand-dynamics based segmentation algorithms tolerate best the gesture and signing speed variations.

III. Concept

The system overview is shown in Fig. **1**. InterpreterGlove system overview. Main building blocks of the software system are the glove, the training environment, the mobile applications and the backend server.

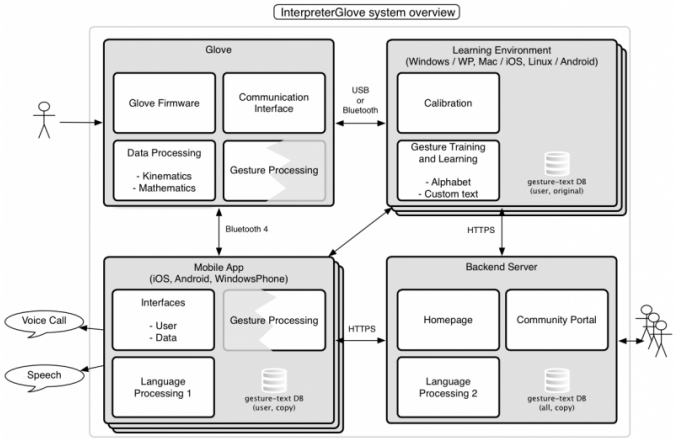

Fig. 1. InterpreterGlove system overview

The glove prototype is made of breathable elastic material, the electronic parts and wiring are ergonomically designed to ensure seamless daily use. The hardware consists of twelve integrated nine-axis motion-tracking sensors and incorporates a wireless communication module. The integrated sensors involve a three-axis accelerometer, a three-axis gyroscope and a three-axis magnetometer. As none of the sensors, by themselves, are capable of determining spatial orientation stably, signal fusion algorithms are used to calculate the absolute pitch, roll and yaw components of the sensors fixed on the glove. To determine the joints' deflections, relative angles are calculated per sensor pairs. The glove firmware uses these relative 3D angles to compose the digital copy of the hand, which is denoted by our custom gesture descriptor. Connected to the user's cell phone, the glove transmits the gesture descriptor stream, which is used by the mobile application to perform all the complex high-level computational tasks.

Prior to signing, the glove needs to be configured and adapted to the user's hand, and gestures need to be recorded and assigned to meanings (letters or expressions). Both tasks are executed in the learning environment. Bended and extended fingers, crossed, closed and spread fingers, wrist positions and hand absolute orientation are recorded so that the glove can generate the correct gesture descriptor for any hand state. This is followed by training the fingerspelling alphabet by signing each letter. Users also have the possibility to assign any textual expression to custom hand gestures. The resulting database is stored in a file and transferred to the mobile device.

The mobile application processes the gesture descriptors streamed by the motion-capture glove to produce understandable text and reads it out as audible speech.

The backend server can operate as the central node for community features (gesture exchange) and can provide higher-level language processing capabilities than the offline text processing built into the mobile application.

IV. Hagdil Gesture Descriptor

To create a valuable gesture descriptor it is vital to understand the structure and kinematics of the human hand [12]. The thumb consists of two links, other fingers have three phalanges, all fingers are connected to the palm, and the palm to the wrist by joints. The thumb movement has 5 DoF (Degree of Freedom), other fingers have 4 DoF and the wrist rotational and translational movement has 6 DoF, this adds up as a 27 DoF system for the human hand.

We specified our custom gesture descriptor called Hagdil with these requirements in mind. Its semantics describe the general human biomechanical characteristics and also align to the specific requirements of the glove's capabilities. A Hagdil descriptor stores all substantial characteristics of the human hand: parallel and perpendicular positions of the fingers relative to the palm, wrist position and absolute position of the hand. We use it to encode the users' gestures to be transmitted to the mobile device.

Hagdil semantics differentiate 18 hand parameters. Ten parameters describe the perpendicular state of the thumb relative to the palm and the bended states of finger phalanges. The next four denote the parallel position of the thumb relative

to the palm and the crossed-closed-spread state of the fingers. Two parameters describe the absolute orientation of the hand and another two give the position of the wrist. Each parameter can have 16 different values but the protocol leaves the opportunity to extend it to 256 for more sophisticated application scenarios.

We also experimented with a different approach for gesture coding. Instead of taking the anatomic-kinematic model of the hand as a basis we analyzed the fingerspelling alphabet and determined the minimum necessary set of distinct states for each finger – we call this approach KML. We found various pros and cons when comparing Hagdil to KML. The number of possible states of KML is much smaller than that of Hagdil, making it easier to process. However, it hardly allows users to record custom gestures to their personal "fingerspeaking" dictionary. So Hagdil is a better choice for a customizable product, while KML might be a better decision when creating a control solution based on predefined gestures.

V. GLOVE-TO-TEXT

30 Hagdil descriptors are generated by the InterpreterGlove and transmitted to the mobile application every second. During testing sessions, we logged every data packet – containing a Hagdil descriptor and metadata – to a file for later evaluation. We tested the algorithms discussed below using these logs. We not only made conclusions about the algorithm qualities, but the signing timings were also analyzed using the timestamps in the logs. We found that a deaf user typically uses two signs each second while testing the prototype. During signing, the hand is almost still – forming a valid sign – for about an equal amount of time as it is in transitional state moving from one sign to another. As an English word typically consists of 5 letters, our testers communicated one word every 2.5 seconds average.

Two types of algorithms are applied on the generated Hagdil descriptor stream to transform it into understandable text. First, raw text is generated as a result of the segmentation by finding the best gesture descriptor candidates. The raw text may contain spelling errors caused by the human user's inaccuracy and natural signing variations. Then, the auto-correction algorithm processes this raw text and transforms it into understandable words and sentences that can be read out loud by the speech synthesizer.

A. Raw text generation via segmentation

The *simple Hagdil similarity based* algorithm iterates all inputs and searches the nearest valid Hagdil descriptor in the Hagdil definitions' database. If one exists within a given Levenshtein distance, the corresponding letter is appended to the raw output text. Repeating letters are omitted. This algorithm was quick to implement but its output quality proved to be highly dependent on the quality of the input stream.

The *repetition-based* algorithm also searches for the nearest neighbor but instead of omitting the repetitive letters it uses repetition as the measure of probability. Only letters with at least n occurrences are used in the raw output. For f input frequency, the best results were obtained using $n = f/4$. This algorithm is much more resistant to small errors in the input stream but is still too sensitive.

The *sliding window algorithm* uses a window size of $f/2$ to evaluate the input. The frequency of each occurring gesture descriptor is calculated. Those reaching the – previously mentioned – threshold n are used in the raw output text. This algorithm also omits repetitions.

As learnt from the pilot tests with deaf users, about half of the input data represents transitions between two valid hand states. During these transitions, descriptors are too distant from any defined Hagdil state therefore cannot be matched. We call these descriptors unmatched Hagdil descriptors. The *segmentation along unmatched Hagdil descriptors* is based on a finite-state machine with a transition state and a recording state. When at least two matched descriptors follow each other it switches to recording state. If it encounters an unmatched Hagdil descriptor it falls back to transition state. In recording state it records all matched Hagdils – typically belonging to the same letter. Before changing back to transition state the algorithm outputs the letter corresponding to the most frequent Hagdil descriptor during the recording period.

In case of *segmentation along known transitions* all possible defined-Hagdil to defined-Hagdil transitions are pre-recorded. The input stream is searched for these transitions. This algorithm is very robust as each gesture is recognized twice: once in the "previous gesture to current gesture" transition, and once in the "current gesture to following gesture" transition. The big disadvantage of this algorithm is that all transitions need to be pre-recorded and stored. Adding a new gesture would require the user to re-sign at least $2k$ transitions (where k is the number of all stored gestures, typically between 26 and 50). As customizability is a fundamental feature of the InterpreterGlove, this algorithm was rejected and not implemented for evaluation.

Our next segmentation algorithm is based on the principle that the fingers' motion is much faster while in transition state and is quite stable while showing a sign. The *kinetics based* algorithm calculates the fingers' dynamics examining each descriptor pair in the input stream. If more than p Hagdil positions change or if the sum of the absolute differences at each Hagdil position is greater than s than the algorithm assumes that the hand is moving, otherwise it assumes that the hand is still, showing a sign (best results were produces using $p=2$ and $s=50$). In still state, the nearest matchable Hagdil is selected for the raw output text.

To evaluate segmentation algorithms we used a text similarity measure to compare the original text to the output of each algorithm. The table below shows the average similarity percentages after several signing sessions.

Name of the algorithm	**Similarity**
Original text	1
Hagdil similarity based	0.58
Repetition-based	0.58
Sliding window	0.79
Unknown Hagdil descriptors	0.72
Kinetics based	0.77

The above results show that the best methods are the *sliding window* and the *kinetics based algorithms*. Comparing the two we found that the sliding window approach outperformed the kinetics based method in good conditions (steady signing pace, gloves in place, etc.). In worse conditions the latter algorithm produced better results, generating more robust, stable output. The sliding window algorithm seems a better choice for known environments whereas the kinetics based solution is more reliable to changes to fluctuating input quality.

B. Text auto-correction

The output of the segmentation may contain spelling errors. Although most letter duplicates are eliminated by the segmenter algorithms, letter omission or false letter insertion may occur in the raw text. We found that the InterpreterGlove errors do not resemble to typical human typing errors. While spell checkers are good at correcting the most common misspelled words, mistakes that arise from mistyping some letters on the keyboard and common typos, the glove errors show different characteristics.

For the reasons above a text auto-correction algorithm has been created based on Levenshtein distance calculation supplemented with n-gram and confusion matrix searches and operations. The algorithm examines each word in the output of the segmentation. Words that do not exist in the database are compared to a set of resembling words. The correction cost – changing the original word to an existing one – is calculated for each pair of words.

For computing distance we used a modified Levenshtein algorithm where distances represent the substitution cost between two words. Substitution costs are calculated by using a two-dimensional confusion matrix that incorporates all possible letter pairs. Letter substitution costs are calculated using Hagdil descriptor distances with our aforementioned Hagdil similarity algorithm.

VI. Evaluation and Field Testing

At the planning phase of the project we consulted the deaf-blind about their needs. Albeit we learned fingerspelling, our basic Dactyl knowledge fell short for evaluating the prototypes, so we involved members of the targeted community as pilot testers at an early stage. We learned a lot from their reactions, which deeply influenced our work and achievements concerning many characteristics of the final prototype: optimal output frequency, Hagdil semantics, adequate Hagdil value domain, definition of hand gestures and so on.

The Hagdil semantics – similarly to the project as a whole – has been created as the result of an iterative development process. We planned, implemented, tested, measured and analyzed each version from multiple aspects: whether it fully describes the fingerspelling gestures as expected, if the generated descriptors overlap, if there are sufficient distances in between, etc. We also researched the necessary and sufficient value-domain for each descriptor position to ensure fault tolerance and at the same time enable using any finger position when training and recording custom gestures. Every Hagdil version has been checked for consistency programmatically, evaluated by the research group and also reviewed by a sign-language interpreter and a deaf person through live testing with the prototype.

VII. Future Work

As we aim to support people with special needs, we allow high-level software customizability. Beyond gesture training and custom gesture-text definition possibilities, we let users interact with the software using hand gestures and a graphical interface.

Although originally targeted to the deaf people, we realized that this tool has high potential for many other target groups like speech-impaired, physically disabled people or those coming through a stroke during rehabilitation. We plan to address their special needs in future projects.

We identified two major improvement areas that can have a huge impact on the application-level possibilities of this complex system. Integrating the capability of detecting dynamics – to perceive finger and hand movement direction and speed – opens up new interaction possibilities. Expanding the motion capture coverage by including additional body parts opens the door for more complex application scenarios. For gloves expanded to both arms and hands we will need to solve many challenges among which one of the most important is the synchronization of the hands.

References

[1] Taiwoo Park, Jinwon Lee, Inseok Hwang, Chungkuk Yoo, Lama Nachman, Junehwa Song. 2011. E-Gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems* (SenSys '11). ACM, New York, NY, USA, 260-273.

[2] In-Kwon Park, Jung-Hyun Kim, Kwang-Seok Hong. 2008. An implementation of an FPGA-based embedded gesture recognizer using a data glove. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication* (ICUIMC '08). ACM, New York, NY, USA, 496-500.

[3] Yun Li, Xiang Chen, Jianxun Tian, Xu Zhang, Kongqiao Wang, Jihai Yang. 2010. Automatic recognition of sign language subwords based on portable accelerometer and EMG sensors. In*International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction* (ICMI-MLMI '10). ACM, New York, NY, USA, Article 17 , 7 pages.

[4] Seungki Min et al., "Simple Glove-Based Korean Finger Spelling Recognition System", Lecture Notes in Computer Science Volume 4705, 2007, 1063-1073.

[5] Cemil Oz, Ming C. Leu. 2005. Recognition of finger spelling of American sign language with artificial neural network using position/orientation sensors and data glove. In *Proceedings of the Second international conference on Advances in neural networks - Volume Part II* (ISNN'05), Berlin, Heidelberg, 157-164.

[6] Parvini, F.; Shahabi, C., "Utilizing Bio-Mechanical Characteristics For User-Independent Gesture Recognition," *Data Engineering Workshops, 2005. 21st International Conference* , pp.1170,1170, 05-08 April 2005

[7] Farid Parvini, Dennis Mcleod, Cyrus Shahabi, Bahareh Navai, Baharak Zali, Shahram Ghandeharizadeh. 2009. An Approach to Glove-Based Gesture Recognition. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques*, Julie A. Jacko (Ed.). Springer-Verlag, Berlin, Heidelberg, 236-245.

[8] Susanna Ricco, Carlo Tomasi. 2009. Fingerspelling recognition through classification of letter-to-letter transitions. In *Proceedings of the 9th Asian conference on Computer Vision - Volume Part III* (ACCV'09), Vol. Part III. Springer-Verlag, Berlin, Heidelberg, 214-225

[9] Chemuttaai Koech, 2007, A kinematic analysis of sign language

[10] W. W. Kong, Ranganath Surendra, "Automatic hand trajectory segmentation and phoneme transcription for sign language," *Automatic Face & Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on* , vol., no., pp.1,6, 17-19 Sept. 2008

[11] Junwei Han, George Awad, Alistair Sutherland. 2009. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recogn. Lett.* 30, 6 (April 2009), 623-633.

[12] Craig L Taylor, Ph.D., Robert J. Schwarz, M.D, The Anatomy and Mechanics of the Human Hand