Framework for Automated Customer Service in Sign Language

Filip Malawski
AGH University of Science and Technology
Department of Computer Science
Kraków, Poland
fmal@agh.edu.pl

Jakub Gałka
AGH University of Science and Technology
Department of Electronics
Kraków, Poland
jgalka@agh.edu.pl

ABSTRACT

Deaf people need the help of an interpreter in formal relations, such as visiting offices or medical institutions. We present a new framework for building systems for sign language interaction, which can provide basic automated customer service for the deaf. The framework covers all steps required to build such a system from scratch - the acquisition of scenario-specific corpora, extraction of features, training of models, recognition, user interface, integration and configuration of the final application. The usability of the framework has been evaluated by creating a proof-of-concept system for automated scheduling of doctor's appointments in sign language. The results indicate that the process of building a sign language interaction system with our framework is relatively quick and simple. Recognition efficiency was evaluated as well and proved to be sufficient for practical use.

Keywords

Sign Language Recognition, Kinect 2, HMM, Parallel HMM, HCI, Framework

1. INTRODUCTION

It is difficult for the deaf to function normally in society, as they communicate mainly by using sign language (SL), which most people do not know. Writing is not a viable solution, for several reasons. First of all, due to their different education levels, the deaf often do not know how to read and write. The grammar of a spoken and written language is considerably different from the grammar of sign language, therefore it is not easy for them to learn it. Moreover, even those who can read and write consider using sign language to be much faster and much more natural.

This problem with communication has an impact not only on the informal relations between deaf people and healthy persons, but in formal cases as well. In particular, when going to places such as offices or medical institutions, the deaf must rely on an interpreter. Some institutions provide translation services, and in other cases they need their own interpreter. The translation services offered by institutions can take the form of a special appointment with an interpreter on site or a web-conference with a remote interpreter. In either case, providing proper customer service for the deaf is time-consuming and expensive.

In some cases there is yet another concern – deaf people may want to keep their issues private, without involving another person as an interpreter. Particularly in the case of medical institutions, even making an appointment with certain types of doctors may be uncomfortable for deaf patients.

Automatic recognition and translation of sign language could significantly help in addressing the aforementioned issues. It would be less expensive than live interpreters, as well as easily and constantly available and comfortable for the deaf. The main concept of our framework is to provide various institutions with the means of creating a dedicated system, specific for their scenario, which would provide the deaf with at least basic customer self-service without any help from a live interpreter. The idea is similar to Interactive Voice Response (IVR) systems used in call centers. The customers are serviced based on the recognition of user sign-language gestures and pre-recorded or synthesized messages (prompts). Typical cases may be handled automatically and in non-typical ones it is possible to switch to a live interpreter.

We present a framework which allows for a quick and easy creation of automated customer service systems in sign language. It covers all necessary elements, from recording scenario-specific corpora, through data processing, training of models for the recognition module, front-end user application and the configuration of the whole system.

Data acquisition and processing is based on depth-image (Kinect 2) sensor, which provides the recognition of the user pose in the form of a fitted skeleton model. Sign language recognition is based on a Parallel Hidden Markov Model (PaHMM) classifier. The tools for the acquisition of structured corpora, training of models and the configuration of the final system are provided, together with a simple user application. The framework can be easily modified by the substitution of any of the modules. Therefore it is flexible and adaptable not only to various usage scenarios, but to different data sources, processing and recognition methods as well.

The evaluation of our framework was performed by creating a system which allows for automatic scheduling of doctor's appointments using sign language. We chose this particular area due to the results of a questionnaire among deaf people, in which they were asked to indicate situations where such a system would be most useful for them. We cover a simple scenario, consisting of a two step dialog – the selection of the doctor and the day of the appointment. Using our framework we recorded a database of 14 gestures for different doctors and 10 gestures for selecting days in Polish Sign Language. With these data, we trained the models for recognition, configured the dialog flow-chart and finally obtained a working prototype of the system. Based on this use-case we evaluated the usability of the framework and the efficiency of the recognition process. The initial results and user feedback are encouraging.

2. RELATED WORK

Automatic sign language recognition is recently a much-investigated topic. State-of-the-art works differ mainly in terms of data modalities, processing and recognition methods. On the other hand, practical deployments and usage are rarely addressed as research objectives.

Considering data modalities and processing, multiple approaches can be found in literature. Although some works depend solely on RGB data [ThPM14, YaSL10], the usage of depth sensors, such as the Kinect, proved to be greatly beneficial [DoDZ13, KaKh15]. Depth data allows for easy and efficient extraction of the person in the image, as well as body segmentation and tracking. Moreover, the Kinect provides skeleton data, which by itself is often reliable enough for efficient gesture recognition [ASSM16, ISTC14]. In our work, we employ Kinect 2 skeleton data, using normalized positions and orientations of selected joints.

Various machine-learning methods have been employed for the recognition process. The most popular approaches use Dynamic Time Warping [BaDr13, JaKh14, MQSA14] and Hidden Markov Models (HMM) [GFZC04, LiKK16, VeAC13]. Both were shown to provide high accuracy of recognition [RMPS15]. Other approaches include: Support Vector Machines [KoRa14], Neural Networks [AnKS15] and Random Forests [RAWD13]. We employ Parallel HMM, which we found to be superior to the traditional HMM method especially in terms of higher robustness to feature distortions and better classification confidence levels than in classical HMMs. Similar conclusions are reported in [ThPM14].

One particular issue with sign language recognition is that there are different languages in each country, just as in the case of spoken languages. Usually each research team works with SL native to their country – there are works on American SL[DoLY15], Arabic SL[ASSM16], Chinese SL [WCZC15], Indonesian SL[RAWD13], etc. This makes the comparison of results difficult, as there is no international SL database. The situation is no different in our case – we conduct experiments with a custom database, recorded by us specifically for our scenario, using

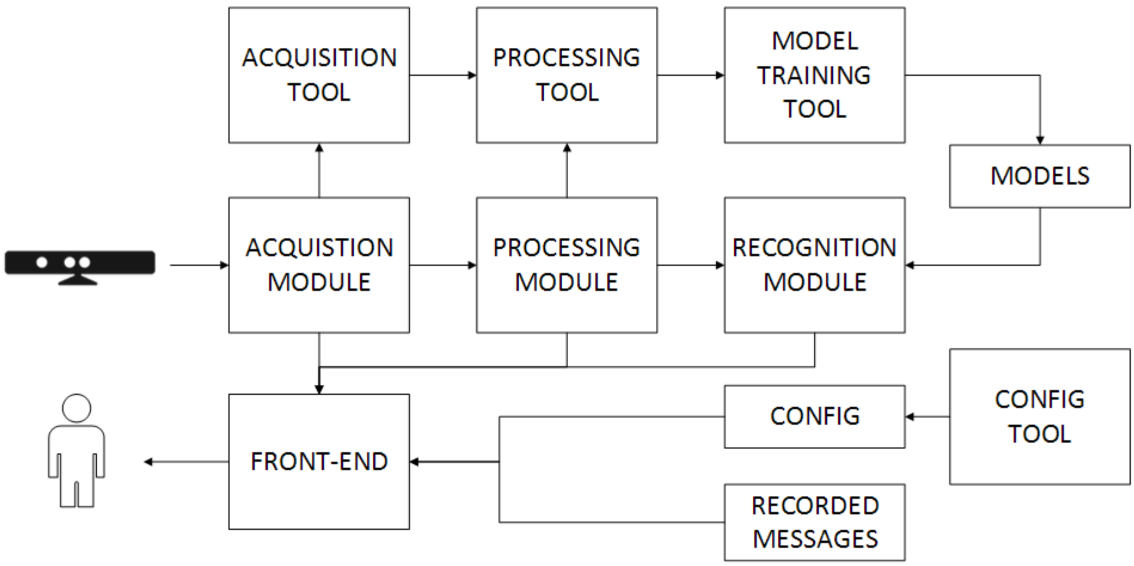

Figure 1: Framework architecture

Polish SL. So far Polish SL has been analyzed in [OsWy13]. It is also important to note, that we report results obtained using a practical system.

As mentioned before, most papers focus on data processing or machine-learning methods, while little work has been done considering the practical usage of these methods. Contrary to that, our framework's main focus is on making these methods available for real-life responsive applications.

3. PROPOSED FRAMEWORK

Overview

Our framework aims at facilitating the creation of interaction systems where the general idea is to display questions to the user and recognize his answers, recorded by a dedicated sensor, such as Kinect 2. Currently, in order to improve the accuracy of the system, the user is actively prompted to answer with one word or phrase only. Therefore, the dialogs must be constructed accordingly, with a set of answers expected by the recognizer grammar related to the specific prompt. Nevertheless, even building such a relatively simple interaction system is challenging and our framework focuses on simplifying the development process.

The architecture of our framework is presented in Figure 1. Core elements include the acquisition, processing and recognition modules. The acquisition module interfaces with the sensor and provides the data. The processing module extracts specific features from raw data. The recognition module classifies a given sample by using previously trained models. These core elements are used by the tools and the front-end application.

The acquisition tool is a dedicated application which allows for a quick and easy recording of a specific database of gestures. The processing tool applies any feature extraction method chosen from the processing module to a given database and produces files with computed features. The model training tool generates models for specific sets of gestures, based on the feature files. The models are then used by the recognition module. The configuration tool provides a user-friendly graphical interface for defining dialogs and editing paths in the configuration file.

The front-end application integrates the acquisition, processing and recognition modules and provides a simple user interface for interaction with the system. It requires recorded messages which are to be displayed. Alternatively these messages can be synthesized by employing a virtual signing avatar, although this feature is still being developed. The settings of the front-end application (paths to models and recordings, dialog flow-chart, etc.) are defined in the configuration file.

Two programming platforms are employed in the current implementation, namely MATLAB and .NET. The front-end application, acquisition, processing and configuration tools as well as all three core modules are implemented in .NET. The model training tool is implemented in MATLAB, as it allows for relatively quick verification of different approaches. Both the recognition module and the model training tool employ an external implementation of HMM, namely HTK[1].

The elements of the framework which are loosely coupled communicate by creating specific files. The acquisition tool produces avi files with RGB and depth data as well as MATLAB .mat files with skeleton data. The processing tool produces .mat files with the features. We employ .mat files for storing both the skeleton data and the features, since this format can be easily handled not only in MATLAB, but in .NET as well, due to a dedicated library. This also facilitates the potential substitution of the elements of the framework. The models are saved in an HTK-specific format and the configuration file uses the XML format. Communication between the tightly-coupled elements (core modules, front-end application) employs dedicated .NET classes.

Acquisition and processing modules

In current implementation of the acquisition module we employ the Microsoft Kinect 2 sensor. It provides multiple data streams, namely RGB images, depth images and skeleton data. The Kinect 2 skeleton model consists of 25 joints, which is an improvement compared to the first version of the Kinect, which provided only 20 joints. Additional joints include the spine between the shoulders, the tip of the left and right hand, and thumbs. The newly added tip and thumb joints are particularly interesting in the context of gesture recognition. For each joint x, y, z the coordinates in the camera frame-of-reference space are provided. Additionally, for all joints except for the tips and thumbs, orientations are given in the form of quaternions.

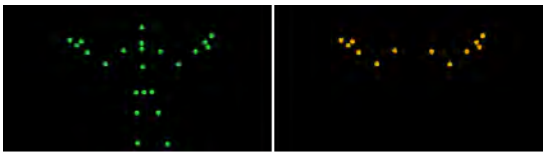

Figure 2. Kinect 2 skeleton joints (sitting person). Left: all tracked joints. Right: joints selected for sign language recognition.

We decided to use skeleton data, as it enables relatively accurate tracking of joint trajectories during signing. Moreover it is robust to illumination and background changes. We identified all joints from both hands and arms to be the most relevant for

[1] http://htk.eng.cam.ac.uk/

gesture classification. The selected joints include (for each hand): shoulder, elbow, wrist, hand, tip, and thumb (see Figure 2). We use x, y, z coordinates for all selected joints as well as all available orientations. In order to better adapt the system to various users, the positions of joints are normalized relative to the position of the head. Many gestures in sign language are performed in a specific position relative to the head, e.g. a dentist requires performing the gesture near the mouth. Therefore the head was chosen as a good reference point for the other joints. After normalization, the first and second derivatives of the joint positions are computed as additional information for the recognition module.

Recognition module

Hidden Markov Models are often used for gesture recognition, due to their high efficiency in temporal pattern recognition. The idea of HMM is to build a model of hidden states, based on known observations. Parallel HMM contains multiple channels, each with its own model. During the classification process, outputs from all channels are combined into a single result. There are multiple approaches to both grouping features into channels and combining the results.

In our system we employ PaHMM with grouping based on the skeleton joints. The positions and orientations of each joint form separate joint-feature channels for the PaHMM (see Figure 3). This method of grouping features corresponds directly to the nature of the modeled phenomenon - various gestures engage various joints to a different extent. Particularly, some gestures differ in the positions of joints during movement, while other ones differ only in orientations.

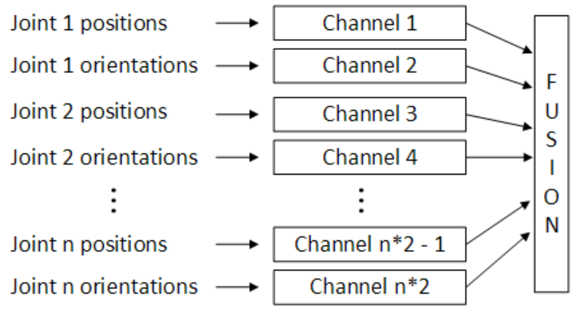

Figure 3: Parallel HMM channels

We perform a weighted fusion of the modeling results from all channels by assigning fusion weights during training. The classification is then performed based on the weighted log-likelihood channel fusion. The weights are computed based on the accuracy of each separate channel. For the sake of comparison we also evaluate the database with a classical, single channel full joint-feature HMM.

The key assumption in our framework is that each question (system prompt) has its own set of expected possible answers and therefore, for each question, we train a separate recognition model. This allows to achieve high recognition efficiency while providing a convenient interaction method for the users.

Tools

Our framework includes a number of tools, which are not directly used in the final front-end application, but are essential in adapting it to a given scenario during deployment phase.

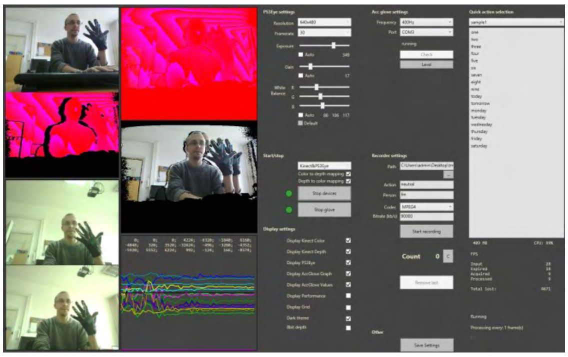

Figure 4: Acquisition tool

A proper amount of data is crucial for the recognition and the database acquisition step can be optimized only by efficient recording procedures. For the purpose of effective data acquisition we have created dedicated software (see Figure 4), which allows to record data from Kinect 2. The operator needs only to press 'start/stop' button, therefore during the recording sessions only a few seconds are needed for each sample. The data is recorded in a format and structure compatible with the rest of the framework and can be directly sent to the processing tool.

Although we currently employ skeleton data only, the software can record all Kinect 2 data streams. Moreover it allows for the acquisition of data from other devices, namely two PS3Eye cameras and an accelerometer glove and provides data stream synchronization mechanisms. These additional data may be used in different approaches or deployment scenarios. The software and its source code are freely available under the GPL license[2].

The processing tool allows to easily process an entire recorded database with a method selected from the processing module and to produce a properly structured database of feature files. The model training tool employs these features to generate Parallel HMM models, which can be directly used in the recognition module. The training process is parallelized in order to accelerate this step in case of multi-core CPUs. The algorithms in this tool can be modified in order to validate different learning and

[2] https://github.com/fmal-pl/MultiSourceAcquisition

recognition approaches. Finally, the configuration tool allows to easily edit the XML configuration file, which defines the dialog, as well as paths to models and recordings, etc. It provides a convenient graphical tool for editing the conversation flow-chart (see Figure 5).

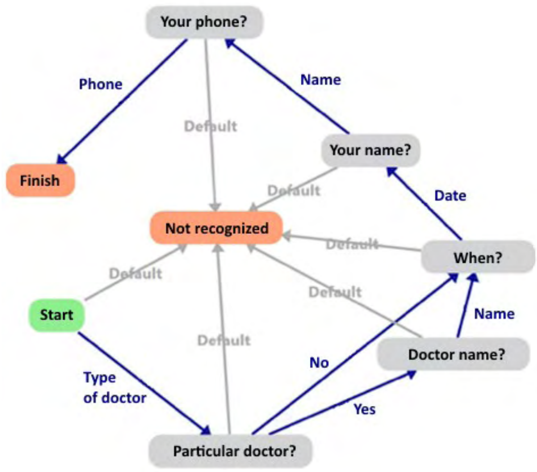

Figure 5: Dialog definition flow-chart interface in the configuration tool

Front-end

The translation site is equipped with a display, a computer and the Kinect 2 sensor mounted below or above the display. The user stands or sits in front of the display and interacts with the system (see Figure 6). The user interface of the front-end application has a single window, where recorded messages and video feed from the camera are displayed interchangeably. The application can also display subtitles, mostly for the sake of demonstrational purposes, for people with poor or no knowledge of sign language.

Figure 6: Translation site

The interaction consists of a series of steps. In each step, a message ending with a question is displayed and the user is prompted to respond by using sign language. During the response stage, the video feed from the Kinect RGB camera is displayed, so that the user can see how his answer is recorded. The time available for answering is set by default to 3 seconds, although in can be configured differently.

The response is recorded, processed and classified. The system has a database of recorded messages with all available answers for each question. It displays a message corresponding to the recognized gesture as feedback to the user and then proceeds to the next question accordingly to the conversation flow-chart. The system may ask the user to repeat the gesture if the confidence of the recognition is below the expected threshold. It prevents false positives and increases system reliability significantly. In the case of multiple failed recognitions the system displays a message asking to switch to a live interpreter and ends the interaction. In case when answers to all questions are recognized an end message is displayed after the last question and the system is ready to send the gathered information.

4. EVALUATION

Proof-of-concept

We evaluated our framework by using it to implement a proof-of-concept system for the automatic scheduling of doctor's appointments using sign language. The scenario assumed a simple, yet functional dialog, where the user is asked which doctor she or he would like to see and when.

For this scenario we recorded a database of 24 different gestures, signed by 7 persons, who were coached and supervised by a professional signer. The database consists of 14 gestures associated with various medical specializations (allergist, cardiologist, dentist, dermatologist, diabetologist, dietician, gastrologist, gynecologist, laryngologist, ophthalmologist, oncologist, psychologist, psychiatrist, and surgeon), and 10 gestures associated with selecting days (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, today, tomorrow, and day after tomorrow). Figure 7 presents the key frames from some of the recorded signs. Each gesture was repeated by each person 10 times. The dataset will be available for academic community since October 2016 (please contact the authors for details).

The two categories of gestures included in the database (doctors and days) were chosen not only because they fit the scenario, but also due to their distinct characteristics. In the case of doctors the gestures are considerably different from each other, often performed using two hands and with various kinds of hands movements (see Figure 7, top row). Gestures for days, on the other hand, are very similar. In some cases they differ only in the positioning of the fingers, while the movement of the hands remains the same (see Figure 7, bottom row). This constitutes a much more difficult classification problem, in particular for the methods based on Kinect 2 skeleton data, which contain the positions of only two fingers (index and thumb) and even so, these data are often not quite accurate. Therefore different recognition accuracy is to be expected for each category - an issue discussed in the recognition evaluation section.

The recorded database was run through the processing tool in order to extract the features - the normalized positions and orientations of selected joints. In the next step, the model training tool was employed to prepare the models (separate for each category) for the recognition module. The dialog was defined in the graphical editor of the configuration tool and the video messages required for the dialog were recorded using a standard camera. Finally, the remaining part of the configuration was prepared (paths to models and recorded messages) and uploaded to the front-end application.

Figure 7: Key frames of selected gestures from recorded evaluation database. Top row: allergist, dermatologist. Bottom row: Tuesday, Wednesday

Usability

In this section we evaluate the usability of the framework in context of the task of preparing a proof-of-concept system, by analyzing each step of the process.

The time needed for a recording session with one person was approximately one hour, including teaching the person each gesture prior to signing it. Therefore, the acquisition of the entire database of 24 gestures with 7 persons required approximately 7 hours. More complex scenarios would naturally require more time to record all of the necessary gestures. Compared to our previous experiences with general purpose recording software, our dedicated acquisition tool significantly reduced the time needed for both recording and preparing the data for further processing.

In order to extract the features, the processing tool required only to enter the path to the recorded database and the path to the main directory of the feature files. The extraction lasted only a few minutes, since the processing of skeleton data is not very time-consuming. The model training tool operated in a similar fashion – it required input and output paths and also a division of the dataset, since doctors and days were trained as separate models for separate questions in the dialog. We used our default settings for the learning process, although depending on the approach, the configuration of the model training tool may require selecting the proper parameter values. The time needed for training both models was approximately 1 hour on a computer with a 4-core 3.8GHz processor.

Recording the video messages for the dialog (with a standard camera) took a approximately 2 hours, mostly due to multiple repetitions required to achieve satisfactory quality of each video. The configuration of the dialog and the remaining settings of the front-end application took about an hour. The front-end application, although simple, was positively rated by two professional signers, who found the system to be convenient and innovative.

Summarizing, the framework allowed to quickly and easily create a functional system for basic customer-service in sign language. In the case of the proof-of-concept system it took only several hours to record the database and another few to setup the rest of the system. All steps were easy to perform and required little specific knowledge. Although complex scenarios would naturally require more work, we conclude that our framework greatly facilitates the creation of such systems.

Recognition results

Proper recognition efficiency is essential for the practical use of interaction systems build with our framework, which is why it was also analyzed. As mentioned before, we trained separate models for each question, therefore the evaluation of recognition was performed separately for both categories (doctors and days). For comparison, we used both HMM and PaHMM classifiers.

We evaluated only the user-independent case, as this is relevant to our practical application. We used the leave-one-out cross-validation scheme - samples from each person were tested with a model trained on all the other persons. Since we have 7 persons in the database, there were 7 folds of the cross-validation. The presented results were averaged from all folds.

We employed multiple performance measures in order to be able to better assess how our system would suit the given scenario. They are based on the following values: P (positive) – number of examples in the given class, N (negative) – number of examples in other classes, TP (true positive) – number of correctly classified positive examples, TN

- number of correctly classified negative examples, FP - number of negative examples classified incorrectly as positive examples, FN - number of positive examples classified incorrectly as negative examples Employed performance measures include:

$$accuracy = \frac{TP + TN}{P + N} \quad (1)$$

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

We also use error equal rate (EER), which is the rate at which both the false acceptance and false rejection rates are equal.

Results for the 'doctors' category are given in Table 1 and for the 'days' category in Table 2. These include both HMM and PaHMM results.

%	Acc	EER	F1	Prec.	Recall
HMM	83.16	5.40	83.41	83.64	83.17
PaHMM	91.22	4.08	91.22	91.22	91.23

Table 1. Results for 'doctors' category

%	Acc	EER	F1	Prec.	Recall
HMM	67.00	14.95	66.95	66.90	67.00
PaHMM	75.57	9.92	75.51	75.46	75.57

Table 2. Results for 'days' category

Several conclusions may be drawn from the results. First of all, there is a considerable discrepancy between the results for the 'doctors' and 'days' category. As indicated in the database description, 'days' gestures were expected to be a more difficult case due to their similarity. As they differ mostly in the positioning of the fingers, there is a clear need to enhance the recognition process with features corresponding to hand shapes in order to properly handle this type of gestures. On the other hand, Kinect 2 skeleton data is sufficient to achieve satisfactory results for hand-shape-independent gestures.

In both categories, the results for PaHMM are superior to the classical HMM. The absolute difference in accuracy, precision, recall and F1 score is approximately 8%, which constitutes a significant improvement. Similar values of precision, recall and F1 score indicate that the recognition model is well balanced. Low EER is particularly important in case of practical applications. It indicates that the confidence threshold of the classifier may be set in such a way that almost all false positives are rejected, while at the same time almost all true positives are accepted. This corresponds directly to the usability of the final system. In the case of the 'doctors' category the EER is sufficiently low to put the model into a practical use. In the case of the 'days' category, as mentioned before, hand shape features need to be added to achieve comparable results. It is worth mentioning, that the results were obtained with relatively small database, and increasing number of subjects and samples is likely to improve recognition efficiency as well.

5. CONCLUSIONS

We presented a complete framework for building sign language interaction systems for providing basic customer service for the deaf. We evaluated the feasibility and usability of the framework by creating a simple system for making appointments with a doctor in sign language. We concluded that the framework enables easy and quick creation of sign language interaction self-service systems, while providing significant use case flexibility. It can be easily adapted to different scenarios and different recognition approaches. The recognition efficiency indicates that in the case of the 'doctors' gestures, which are not strictly dependent on hand shapes, the results are sufficient to put the model into a practical use. Low EER corresponds to the high usability and robustness of the final system. In the case of the 'days' gestures, additional features are required in order to handle the different hand shapes better. In the future we intend to add hand shape descriptors extracted from depth images and also extend the framework with a virtual signing avatar, which would provide an alternative to the pre-recorded video messages.

6. ACKNOWLEDGMENTS

This work was supported by the Polish National Centre for Research and Development - Applied Research Program under Grant PBS2/B3/21/2013 titled "Virtual sign language translator".

7. REFERENCES

[AnKS15] Anand, V., Keskar, A. G., and Satpute, V. R.: Sign Language Recognition Through Kinect Based Depth Images And Neural Network. 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 194–198, 2015

[ASSM16] Amin, O., Said, H., Samy, A., and Mohammed, H. K.: HMM based automatic Arabic sign language translator using Kinect. 10th International Conference on Computer Engineering and Systems, ICCES 2015, pp. 389–392, 2016

[BaDr13] Barczewska, K. and Drozd, A.:

Comparison of methods for hand gesture recognition based on Dynamic Time Warping algorithm. 2013 Federated Conference on Computer Science and Information Systems, pp. 207–210, 2013

[DoDZ13] Dominio, F., Donadeo, M., and Zanuttigh, P.: Combining multiple depth-based descriptors for hand gesture recognition. Pattern Recognition Letters, Elsevier B.V., 2013

[DoLY15] Dong, C., Leu, M. C., and Yin, Z.: American Sign Language Alphabet Recognition Using Microsoft Kinect. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 44–52, 2015

[GFZC04] Gao, W., Fang, G., Zhao, D., and Chen, Y.: A Chinese sign language recognition system based on SOFM/SRN/HMM. Pattern Recognition vol. 37, Nr. 12, pp. 2389–2402, 2004

[ISTC14] Ibañez, R., Soria, Á., Teyseyre, A., and Campo, M.: Easy gesture recognition for Kinect. Advances in Engineering Software vol. 76, pp. 171–180, 2014

[JaKh14] Jambhale, S. S. and Khaparde, A.: Gesture recognition using DTW & piecewise DTW. 2014 International Conference on Electronics and Communication Systems, ICECS 2014, pp. 1–5, 2014

[KaKh15] Kane, L. and Khanna, P.: A framework for live and cross platform fingerspelling recognition using modified shape matrix variants on depth silhouettes. Computer Vision and Image Understanding vol. 141, Elsevier Ltd., pp. 138–151, 2015

[KoRa14] Kong, W. W. and Ranganath, S.: Towards subject independent continuous sign language recognition: A segment and merge approach. Pattern Recognition vol. 47, Elsevier, Nr. 3, pp. 1294–1308, 2014

[LiKK16] Li, T. H. S., Kao, M., and Kuo, P.: Recognition System for Home-Service-Related Sign Language Using Entropy-Based K -Means Algorithm and ABC-Based HMM. IEEE Transactions on Systems, Man, and Cybernetics: Systems vol. 46, Nr. 1, pp. 150–162, 2016

[MQSA14] Masood, S., Qureshi, M. P., Shah, M. B., Ashraf, S., Halim, Z., and Abbas, G.: Dynamic time wrapping based gesture recognition. 2014 International Conference on Robotics and Emerging Allied Technologies in Engineering, iCREATE 2014 - Proceedings, pp. 205–210, 2014

[OsWy13] Oszust, M. and Wysocki, M.: Polish sign language words recognition with Kinect. 2013 6th International Conference on Human System Interactions, HSI 2013, pp. 219–226, 2013

[RAWD13] Rakun, E., Andriani, M., Wiprayoga, I. W., Danniswara, K., and Tjandra, A.: Combining depth image and skeleton data from Kinect for recognizing words in the sign system for Indonesian language (SIBI [Sistem Isyarat Bahasa Indonesia]). 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 387–392, 2013

[RMPS15] Raheja, J. L., Minhas, M., Prashanth, D., Shah, T., and Chaudhary, A.: Robust gesture recognition using Kinect: A comparison between DTW and HMM. Optik vol. 126, Elsevier GmbH., Nr. 11-12, pp. 1098–1104, 2015

[ThPM14] Theodorakis, S., Pitsikalis, V., and Maragos, P.: Dynamic-static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition. Image and Vision Computing vol. 32, Elsevier B.V., Nr. 8, pp. 533–549, 2014

[VeAC13] Verma, H. V., Aggarwal, E., and Chandra, S.: Gesture recognition using kinect for sign language translation. 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), pp. 96–100, 2013

[WCZC15] Wang, H., Chai, X., Zhou, Y., and Chen, X.: Fast sign language recognition benefited from low rank approximation. 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015, 2015

[YaSL10] Yang, R., Sarkar, S., and Loeding, B.: Handling Movement Epenthesis and Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming. IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 32, Nr. 3, pp. 462–477, 2010