Real-time Sign Language Letter and Word Recognition from Depth Data

Dominique Uebersax[1] Juergen Gall[1] Michael Van den Bergh[1] Luc Van Gool[1,2]
[1]BIWI, ETH Zurich [2]ESAT-PSI / IBBT, KU Leuven
uebersaxd@gmail.com {gall,vandenbergh}@vision.ee.ethz.ch vangool@esat.kuleuven.be

Abstract

In this work, we present a system for recognizing letters and finger-spelled words of the American sign language (ASL) in real-time. To this end, the system segments the hand and estimates the hand orientation from captured depth data. The letter classification is based on average neighborhood margin maximization and relies on the segmented depth data of the hands. For word recognition, the letter confidences are aggregated. Furthermore, the word recognition is used to improve the letter recognition by updating the training examples of the letter classifiers on-line.

1. Introduction

The American sign language (ASL) is a visual-gestural language used by deaf people in North America and in other countries around the globe. Over half a million people use ASL to communicate with each other as their primary language. ASL recognition systems can be used for education of children or newly hearing impaired, as well as for live interpretation applications to facilitate the communication between hearing and deaf people.

In this work, we present a system which translates gestures signed in front of a time-of-flight (TOF) camera into the corresponding letters of the ASL finger alphabet. Additionally, the system is able to guess the most likely word currently spelled with the individual letters in real-time. Exploiting the advantages of depth data, a hand segmentation algorithm is introduced that relies on the single assumption that the hand performing the gestures is the object closest to the camera. Using this premise, no further information like the skin color, markers for the hand, or a special recording setup are needed.

For letter classification, we have evaluated three methods. The first method relies on a codebook of hand gestures where each codebook entry contains only one single training example. The similarity of an extracted hand and a codebook entry is computed by the difference of normalized depth values. This method works only in a single-user setup where training data is provided by the user. The second method is based on average neighborhood margin maximization (ANMM) [26] that is more suited for classification of hand gestures in a multi-user environment, where the user does not provide any training data. The third method estimates the hand orientation and uses the orientation as additional cue for letter recognition. Based on the letter recognition system, we further propose a word recognition system. To this end, we combine the three letter classification methods and aggregate the letter confidences to recognize words out of a pre-defined lexicon. As an additional feature, we demonstrate that the word recognition can be used to improve the letter classifiers by updating the training samples when a word has been recognized with high confidence. To the best of our knowledge, this has not been previously investigated within this context.

While there exists previous work on gesture recognition systems operating in real-time and using depth data [7, 19, 20, 9, 25] with high recognition rates, the considered datasets are small and consist of well distinguishable gestures. Systems that consider larger datasets and especially finger alphabets [12, 1, 15, 22] still require special environments or markers to achieve high recognition rates. Recognition of spelled words with the use of finger alphabets has so far received very little attention. In [11], histograms of oriented gradients and a hidden Markov model are used to classify words in a single-user setup for the British sign language finger alphabet.

2. Related work

Our approach for recognizing letters of the sign alphabet is related to gesture recognition from depth data and optional color data [10, 14, 2, 20, 7, 19, 9, 22, 8, 25]. In particular, the ANMM classifier has been previously proposed for gesture recognition in [25]. However, gesture recognition is a simpler task since usually only a small set of distinctive gestures are used. In the case of sign languages, the signs for the letters are pre-defined and not very distinctive due to the noise and low resolution of current depth sensors.

978-1-4673-0063-6/11/$26.00

Recognizing signs of visual-gestural languages like ASL is a very active field [21, 3, 16, 18, 27, 23, 28]. For instance, the SignSpeak project [4] aims at developing vision-based technology for translating continuous sign language to text. However, many of these systems try to recognize an arbitrarily selected subset of a sign language, be it by motion analysis of image sequences or recognition of manually extracted static gestures. In the following, we structure comparable methods into single-user systems, *i.e.*, the systems are trained for a single user, and multi-user systems, *i.e.*, the user does not provide any training data:

Single-user systems. Polish finger alphabet symbols have been classified in [13] in an off-line setup. The input for each of the considered 23 gestures consisted of a gray-scale image at a relatively high resolution and depth data acquired by a stereo setup. In [5], a real-time recognition system has been developed for Spanish sign language letters where a colored glove was used. The real-time system [12] recognizes 46 gestures including symbols of the ASL. It assumes constant lighting conditions for training and testing and uses a wristband and special background for accurate hand segmentation. More recently, British sign language finger spelling has been investigated in [11] where the specialty is that both hands are involved in the 26 static gestures. Working on skin color, it is assumed that the signer wears suitable clothing and the background is of a single uniform color. The system recognizes also spelled words contained in a pre-defined lexicon, similar to the word recognition approach in this work.

Multi-user systems. Using a stereo camera to acquire 3D and color data, Takimoto et al. [22] proposed a method for recognizing 41 Japanese sign language characters. Data was acquired from 20 test subjects and the achieved classifier runtime is about 3fps. Although the approach does not require special background or lighting conditions, segmenting the hand, which is a challenging task by itself, is greatly simplified by the use of a black wristband. Colored gloves have been used in [6] for recognizing 23 symbols of the Irish sign language in real-time. A method for recognizing the ASL finger alphabet off-line has been proposed in [1]. Input data was acquired in front of a white background and the hand bounding box was defined for each image manually. A similar setup has been used in [15]. While these works rely on markers like wristbands or gloves to avoid the most challenging task for hand segmentation, namely the detection of the wrist, our approach relies only on raw depth data acquired with a low-resolution depth sensor.

3. ASL word recognition

An overview of the system is given in Fig. 1. The depth data is used for hand localization and segmentation. After rotating and scaling the segmented hand image, the letter is recognized using classifiers based on average neighborhood margin maximization (ANMM), depth difference (DD), and hand rotation (ROT). The confidences of the letters are then combined to compute a word score. The most likely word is accepted if the ratio of its score and the score of the second most likely word surpasses a predefined threshold.

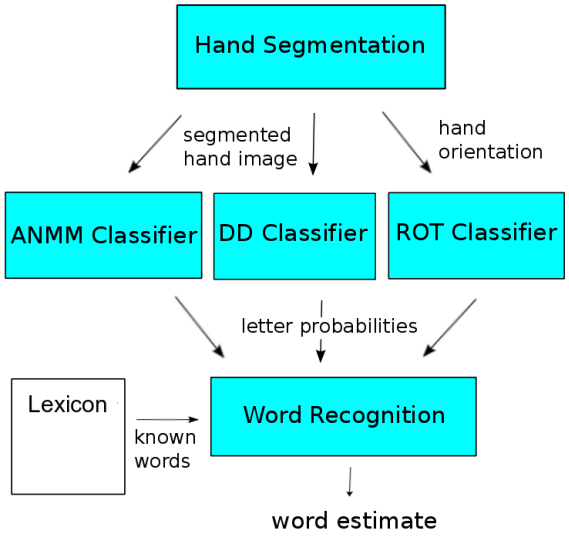
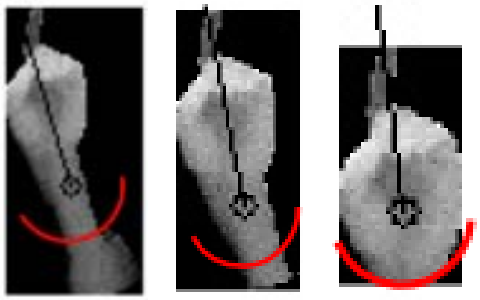


Figure 1. (a) ASL word recognition system setup.

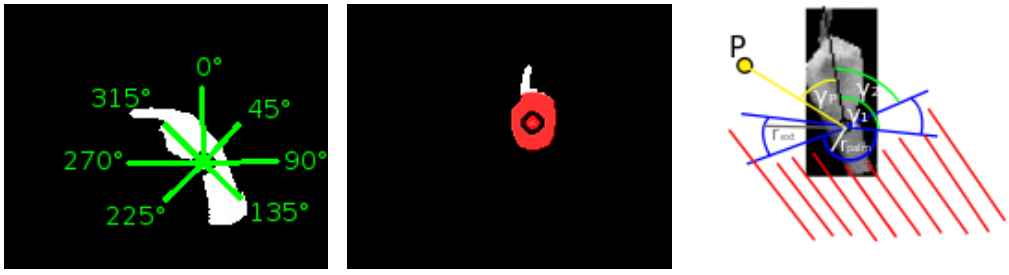


(a) (b) (c)

Figure 2. (a-c) During hand segmentation, the center of the hand (black circle), the palm size (red semicircle), and hand orientation (black line) are iteratively estimated. Starting with a rough segmentation based on thresholding depth data (a), the heuristic iterates and converges in most cases to the correct solution (c). The segmented hand is then normalized and used for classifying the shown letter.

3.1. Letter Recognition

After normalizing and thresholding the depth data where we assume that the closest connected object is the hand of interest, we iteratively estimate the size and position of the palm and the orientation of the hand as shown in Fig. 2(a-c). The depth values are normalized to be in the range of 0 to 1.



(a) (b) (c)

Figure 3. (a) Star model to approximate radius of the palm. (b) The red colored area is the palm. The hand center is computed only for the palm to stabilize the hand segmentation. (c) Illustration of the segmentation refinement step.

3.1.1 Palm detection

After normalization of the hand image $\mathcal{I}$, the gravity $\vec{c} = \frac{1}{|\mathcal{I}|}\sum_{\vec{p}\in\mathcal{I}}\vec{p}$ is computed where $\vec{p}$ are the pixels belonging to the hand. Having the center, we use a star-like profile around the hand center to estimate the radius of the palm, r_{palm}. The profile is rotated with the hand's orientation, leading to seven directions as shown in Fig. 3(a). In each direction, the largest distance from the center to a contour point is measured. As radius, we take the median of the distances scaled by $\alpha = 1.065$ to compensate for a small bias of the median towards smaller hand sizes. Having the radius and previous center of the hand, we re-compute the center $\vec{c}$ by taking only the pixels of the palm into account as shown in Fig. 3(b). Estimating the center of the palm and not of the full hand with fingers is necessary since otherwise the center migrates to the direction of the extended fingers. After the estimation of the palm, depth values that do not belong to the palm or the fingers are removed. A point $\vec{p}$ is discarded if:

$$\begin{aligned}&(\gamma_{\vec{p}} > \gamma_1 \ \wedge\ \|\vec{p}-\vec{c}\| > r_{\text{palm}}) \quad \vee \\ &(\gamma_{\vec{p}} > \gamma_2 \ \wedge\ \|\vec{p}-\vec{c}\| > \eta \cdot r_{\text{palm}})\end{aligned} \tag{1}$$

where $\eta = 1.75$. This is illustrated in Figure 3(c). While the region $\gamma_{\vec{p}} > \gamma_1$ is assumed to contain no fingers and thus all pixels that do not belong to the palm are removed, $\gamma_{\vec{p}} > \gamma_2$ describes the regions left and right of the hand. In this regions, only pixels that are far away from the center are assumed not to be part of fingers.

3.1.2 Orientation estimation

The first estimate of the hand orientation $\vec{d}$ is obtained by principal component analysis (PCA). However, PCA is not always very accurate and we detect finger tips to refine the orientation. Similar to the segmentation refinement step, we define a region of interest based on the current estimated center $\vec{c}$ of the palm:

$$\mathcal{F} = \left\{\vec{p}\in\mathcal{I} : \|(\vec{p}-\vec{c})-\vec{d}\|^2 < \beta\left(\|(\vec{p}-\vec{c})\|^2 + \|\vec{d}\|^2\right)\right\}, \tag{2}$$

where $\beta \geq 1$ is a scale factor to widen the region of interest over 90° on both sides of the orientation vector $\vec{d}$. In our experiments, we use $\beta = 1.1025$, which corresponds to an angle of 130°. Within this region, we greedily search for up to three finger tips. The first finger tip is the pixel $\vec{p}_0 \in \mathcal{F}$ with the largest distance to the center $\vec{c}$ and $\|(\vec{p}_0 - \vec{c})\| > \xi_0 \cdot r_{\text{palm}}$. If a finger tip has been detected, we continue with the second one $\vec{p}_1 \in \mathcal{F}$ with $\|(\vec{p}_1 - \vec{c})\| > \xi_1 \cdot r_{\text{palm}}$ and the angles of the vectors $\vec{p}_0 - \vec{c}$ and $\vec{p}_1 - \vec{c}$ being larger than 18°. The threshold for the three finger tips have been set to $\xi_0 = 1.5$, $\xi_1 = 1.275$, and $\xi_2 = 1.02$. In case that at least one finger tip has been detected, the direction vector $\vec{d}$ is redefined by the average position of the finger tips $\vec{p}_{\text{finger}}$: $\vec{d} = (\vec{p}_{\text{finger}} - \vec{c})/\|\vec{p}_{\text{finger}} - \vec{c}\|$.

3.1.3 Classification

Having the hand image $\mathcal{I}$ segmented and normalized, we can classify the letter signed by the hand. To this end, we use three classifiers. The first is based on a codebook containing one example for each of the N letters. It simply compares the pixel-wise depth distance between the codebook entries $\mathcal{C}_i$ and the observed hand $\mathcal{I}$, *i.e.*,

$$\underset{i\in\{1,\ldots,N\}}{\text{argmin}}\ \Delta^{(i)}_{\text{DD}} \quad \text{where} \quad \Delta^{(i)}_{\text{DD}} = \sum_{\vec{p}} |\tilde{\mathcal{C}}_i(\vec{p}) - \tilde{\mathcal{I}}(\vec{p})|. \tag{3}$$

$\tilde{\mathcal{C}}_i$ and $\tilde{\mathcal{I}}$ are the depth images of $\mathcal{C}_i$ and $\mathcal{I}$ normalized such that the average is 0. The second classifier relies on the hand orientations $\vec{d}_i$ stored for each letter:

$$\underset{i\in\{1,\ldots,N\}}{\text{argmin}}\ \Delta^{(i)}_{\text{ROT}} \quad \text{where} \quad \Delta^{(i)}_{\text{ROT}} = |\vec{d}_i - \vec{d}|. \tag{4}$$

The third classifier is more powerful and is based on average neighborhood margin maximization (ANMM) [26]. The idea is to find a linear projection W to maximize the distance to local neighbors x_k of a data point x_i with different class labels $\mathcal{N}^e_i$ and minimize the distance to neighbors with the same class label $\mathcal{N}^o_i$:

$$\underset{\mathbf{W}}{\text{argmax}}\ tr[\mathbf{W}^T(\mathbf{S}-\mathbf{C})\mathbf{W}], \tag{5}$$

where

$$\mathbf{S} = \sum_{i,k:\vec{x}_k \in \mathcal{N}_i^e} \frac{(\vec{x}_i - \vec{x}_k)(\vec{x}_i - \vec{x}_k)^T}{|\mathcal{N}_i^e|}, \quad (6)$$

$$\mathbf{C} = \sum_{i,k:\vec{x}_k \in \mathcal{N}_i^o} \frac{(\vec{x}_i - \vec{x}_k)(\vec{x}_i - \vec{x}_k)^T}{|\mathcal{N}_i^o|}. \quad (7)$$

The ANMM features are computed as the eigenvectors of the l largest eigenvalues of $\mathbf{S} - \mathbf{C}$, and stored in vector form in a matrix $\mathbf{W}$. In order to keep the system real-time, these ANMM features are approximated with Haarlets [17, 24]. These Haarlets are stored in vector form in a matrix $\mathbf{F}$. During classification, we extract the feature coefficients $\vec{f}$ for the segmented hand and compute the ANMM coefficients by $\vec{y} = \mathbf{C}\vec{f}$ where

$$\mathbf{C} = \mathbf{W} \cdot \left(\left(\mathbf{F}^T \mathbf{F} \right)^{-1} \mathbf{F}^T \right)^T, \quad (8)$$

which is computed during training. In our experiments, we set the number of ANMM feature vectors to $l = 13$. After mapping the feature coefficients $\vec{f}$ to the ANMM coefficients $\vec{y}$, classification is performed by nearest neighbor search, *i.e.*, $\mathcal{I}$ is assigned to the letter with the nearest mean ANMM coefficients.

For combining the classifiers, we compute the confidence for letter i by the weighted sum of the normalized confidences:

$$c_{\text{letter}}^{(i)} = \lambda_{\text{ANMM}} \cdot c_{\text{ANMM}}^{(i)} + \lambda_{\text{ROT}} \cdot c_{\text{ROT}}^{(i)} + \lambda_{\text{DD}} \cdot c_{\text{DD}}^{(i)}, \quad (9)$$

where

$$c_{\text{class}}^{(i)} = \frac{\max_j \Delta_{\text{class}}^{(j)} - \Delta_{\text{class}}^{(i)}}{\sum_{j=1}^{N} \Delta_{\text{class}}^{(j)}}. \quad (10)$$

For letter recognition, the letter with the highest confidence $c_{\text{letter}}^{(i)}$ is taken.

Note that the DD and ANMM classifiers only take the shape of the hand but not the global hand orientation into account due to the normalization. Since some letters like 'H' and 'U' are similar in shape but differ mainly in hand orientation, the additional ROT classifier helps to distinguish these gestures and improves the recognition accuracy as our experiments show.

3.2. Word Recognition

For word recognition, we can use the letter confidences for recognizing finger spelled words. To this end, a lexicon containing all known words is used to correct possible errors of the letter classifiers and to determine word boundaries as well. The straight forward structure of the presented approach allows for very easy addition of new words by simply adding them to the lexicon. The proposed approach aggregates the letter confidences $c_{\text{letter}}^{(i)}$ and computes a confidence value for each word w by:

$$c_w^{(k)} = \frac{1}{k} \left(\sum_{l=1}^{k-1} c_w^{(l)} + c_{\text{letter}}^{(i_k)} \right), \quad (11)$$

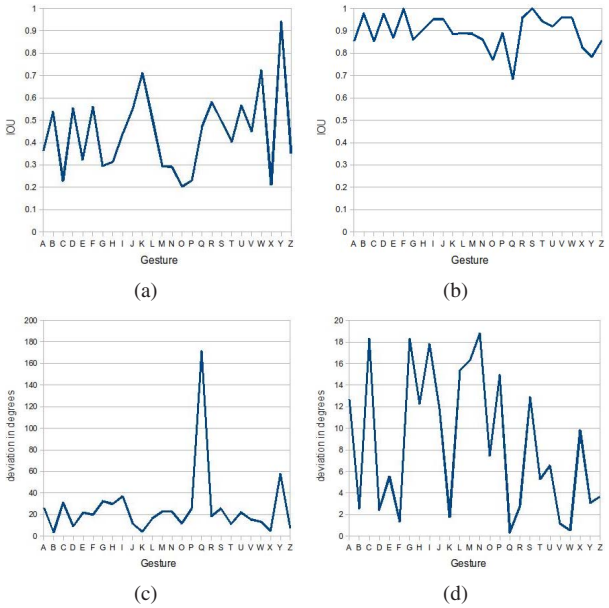
where i_k is the letter at the kth position of the word w. As soon as the confidence ratio of the word with the highest and the second highest confidence is larger than 1.04, the word is accepted. When the ratio is even larger than 1.2, *i.e.*, the confidence of the word is 20% higher than the confidence of any other word of the lexicon, we update the codebook for the DD and ROT classifiers by replacing $\mathcal{C}_{i_k}$ and $\vec{d}_{i_k}$ for each letter i_k of the word. The transitions between signed letters are detected by a movement of the segmented hand, i.e., a letter is only recognized when the observed hand movement over the last 10 frames is small.

4. Experiments

Test data was collected from 7 test subjects at a distance of approximately 80cm from the MESA SR4000 TOF camera. Except of subject 7, the users were not experienced with ASL. For the unexperienced users, a brief introduction to ASL was given before the recordings and the symbols were shown on a display during the recordings. For each user, at least 50 samples are available per letter. In the single-user setup, one hand example per letter is used for building the codebook of the DD and ROT classifiers. In the multi-user setup, we used the data of subject 7 as training data for the DD and ROT classifiers and tested on subjects 1-6. For testing on subject 7, we used the training data from subject 6. The ANMM classifier is trained on the remaining dataset by leaving out the individual test subject's data in each case. The parameters of the system that are specified in Section 3 and not evaluated in Section 4 were empirically determined on a small validation set.

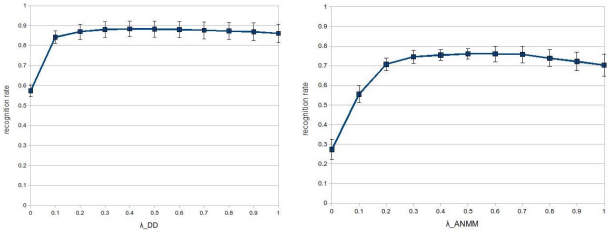
4.1. Hand segmentation and orientation accuracy

The accuracy of the hand segmentation and estimation of the hand orientation for all letters is given in Fig. 4. To this end, we have manually annotated 7 hand gestures for each letter. The segmentation quality is measured by the intersection over union (IOU) ratio of the annotated bounding box and the bounding box estimated by our approach. The hand orientation error is measured as error angle between the annotated and the estimated orientation vector where the results of the first iteration are obtained by PCA. The accuracy of the hand segmentation and hand orientation estimation increases with the number of iterations. After 50 iterations, the method has converged to a reasonable accurate solution, considering the small resolution of the depth sensor.



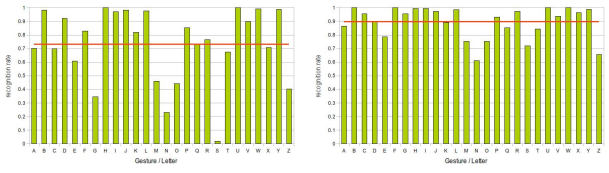
(a) (b) (c) (d)

Figure 4. (a) Segmentation accuracy (IOU) after 10 iterations. (b) Segmentation accuracy (IOU) after 50 iterations. (c) Hand orientation error after 10 iterations. (d) Hand orientation after 50 iterations.



(a) (b)

Figure 5. (a) Single-user: Average gesture recognition rates for different combinations of the DD and ROT classifiers, achieved by varying λ_{DD} and $\lambda_{\mathrm{ROT}} = 1 - \lambda_{\mathrm{DD}}$. (b) Multi-user: Average gesture recognition rates for different combinations of the ANMM and ROT classifiers.



(a) (b)

Figure 6. Letter recognition rates showing differences between the individual letters ($\lambda_{\mathrm{ANMM}} = \lambda_{\mathrm{ROT}} = \lambda_{\mathrm{DD}} = 0.33$). (a) In the multi-user setup, some letters like 'S' are often not correctly recognized. (b) In the mixed setup, the recognition rates of the letters that were difficult to classify in the multi-user setup are significantly improved.

4.2. Letter recognition

The impact of the parameters λ_{ANMM}, λ_{ROT}, λ_{DD} (9) for letter recognition is evaluated in Fig. 5(a-b). Fig. 5(a) shows the error for the single-user case, *i.e.*, training and testing is performed for the same subject. The plot shows that the ROT classifier alone ($\lambda_{\mathrm{DD}} = 0$) is not very useful, but it improves slightly the DD classifier ($\lambda_{\mathrm{DD}} = 1$) for $0.2 \leq \lambda_{\mathrm{DD}} < 1$. The multi-user case shown Fig. 5(b) is more challenging since none of the testing subjects is part of the training data. While the ROT classifier alone ($\lambda_{\mathrm{ANMM}} = 0$) fails again, it improves the ANMM classifier ($\lambda_{\mathrm{ANMM}} = 1$) up to 9%. Note that $\lambda_{\mathrm{ANMM}} = 1$ is comaparable to the recognition method [25]. We have also evaluated a mixed setup where the ANMM classifier is in multi-user mode, i.e., it is not trained on the test subject, while the DD and ROT classifiers are initialized by a single training example for each letter provided by the test subject. Table 1 lists the recognition performance for different setups and shows that the DD classifier does not work well when it is not trained on the testing subject. However, if the DD and ROT classifiers are trained on the same subjects (mixed setup), the recognition accuracy can be increased to outperform the single-user system. This is very practical since DD and ROT use only one example for each gesture and can be updated on-line without additional supervision by replacing the example based on the output of the word recognizer as explained in Section 3.2, whereas ANMM needs to be trained off-line for optimal performance. The impact of the classifier updates is discussed in Section 4.3. For the multi-user and mixed setups, the average letter recognition rates for individual letters are shown in Fig. 6.

Although a direct comparison to related work is difficult since the methods are evaluated on different sets of gestures and datasets, we give an overview in Table 3. If we used a similar amount of gestures as in [19, 20] or [7], namely 6 or 12, we would get comparable recognition rates, namely 0.99 or 0.95. Methods that achieve a higher recognition rate on a large set of gestures [22, 6, 1, 15] do not run in real-time and use markers for a clean segmentation.

4.3. Word recognition

For word recognition, 56 words were selected randomly out of a lexicon of 900 words. The results for different setups are reported in Table 2. Although some letters are difficult to classify (Fig. 6), the word recognition is very reliable. The system has some problems with words like 'us' that are short and contain letters with low recognition rates. In order to overcome this problem, we have proposed in Section 3.2 to update the codebooks for the DD and ROT classifiers based on the output of the word recognizer. Fig. 7 illustrates the improvement that one obtains by this procedure. When comparing the increase of the average letter recognition rate from 0.8 to about 0.9 (Fig. 7(c)) with the

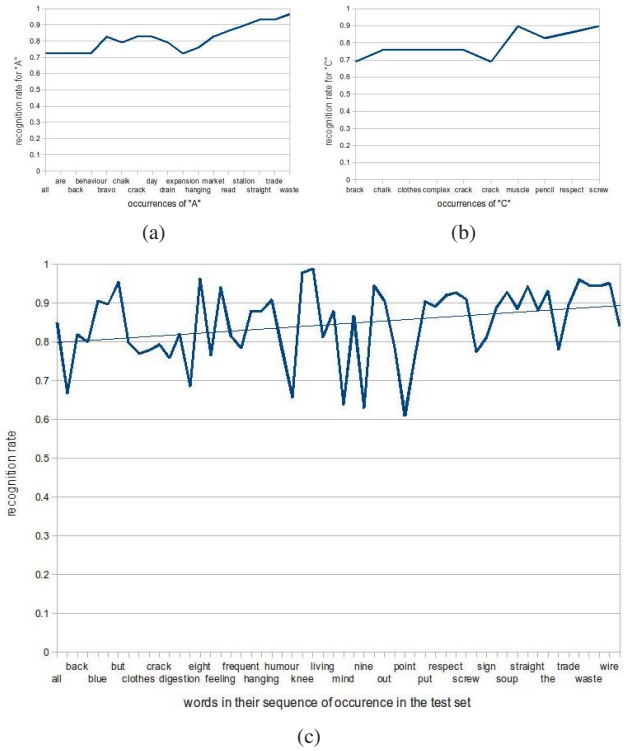


Figure 7. Due to the codebook updates, the letter and the word recognition improves over time. (a) Recognition rates for letter 'A'. (b) Recognition rates for letter 'C'. (c) Average letter recognition rates per word. The linear trend line increases over time.

results given in Table 1, one observes that for the setting $\lambda_{\text{ANMM}} = 0.33$, $\lambda_{\text{DD}} = 0.33$, $\lambda_{\text{ROT}} = 0.33$ the letter recognition accuracy tends towards the ideal performance of the mixed setup. Hence, the update procedure combines the generalizability of a multi-user system with the accuracy of a single-user system.

4.4. Computation time

On a notebook with an Intel Core Duo T2400 1.83 GHz CPU, hand segmentation and letter recognition for a single frame require in average 62.2ms (16fps) where 70% of the computation time are required for the hand segmentation. While the ANMM and DD classifier require around 10% each, the computation for the ROT classifier can be neglected. The remaining 10% are needed for capturing the images and storing the output of the system. We tested our system with two depth sensors. While the MESA SR4000 camera has been used for the evaluation, a video that shows a real-time demonstration with the Kinect camera is part of the supplemental material.

5. Conclusion

In this work, a real-time sign language letter and finger-spelled word recognition system has been presented. In contrast to previous work on gesture recognition from depth data, the system has been evaluated on a very challenging data set, namely the ASL finger alphabet. Although accurate detection results have been previously reported for sign language recognition, the hand segmentation task has been often simplified by the use of markers like a wristband and/or for human-computer interaction impractical recording setups. In this work, we have focused on a practical setup where the segmentation and recognition relies only on depth data acquired with a low-resolution depth sensor positioned in front of the user close to the monitor. During the development of the system, we observed that the hand segmentation is the most critical and most time consuming step for recognition. While the developed heuristic for these tasks performs well for different sensors (Kinect/SR4000), a more efficient non-iterative approach could improve the overall system performance. The evaluation of the classifiers has shown that average neighborhood margin maximization is very efficient and accurate for recognizing sign language letters and that the recognition accuracy can be further improved at negligible cost by taking the orientation into account for classification.

For the word recognition, we have used a lexicon where new words can be added or removed without requiring an additional off-line training step. The results have shown that an accurate recognition of all letters is not necessary for reliable word recognition. Finally, we conclude that the output of the word recognizer can be used as feedback to improve the letter recognition system on-line. In contrast to an on-line update procedure that is performed for each letter classifier independently, this strategy makes use of the dependencies of letters within words, *i.e.*, a letter with low confidence can be updated based on the high confidences of the other letters within the same word. Our experiments have shown that this strategy combines the generalizability of a multi-user system, where the user is unknown, with the accuracy of a single-user system, where the system is trained on the user. We believe that this type of feedback loop is also useful for other human-computer interaction systems.

Acknowledgments. This work is carried out in the context of the Seventh Framework Programme of the European Commission, EU Project FP7 ICT 248314 Interactive Urban Robot (IURO), and the SNF project Vision-supported Speech-based Human Machine Interaction (200021-130224).

Method	# of Gest.	Setup	Depth	Resolution	Markers	Real-time	ARR
[7]	12	multi-user	yes	160x120		yes	0.95
[19]	6	multi-user	yes	176x144		yes	0.94
[20]	5	multi-user	yes	176x144		yes	0.93
[22]	41	multi-user	yes	320x240, 1280x960(rgb)	wristband	no	0.97
[6]	23	multi-user	no		color glove	yes	0.97
[1]	26	multi-user	no		bounding box given	no	0.93
[15]	26	multi-user	no		bounding box given	no	0.92
Proposed	26	multi-user	yes	176x144		yes	0.76
[12]	46	single-user	no	320x240	wristband	yes	0.99
[9]	11	single-user	yes	160x120		yes	0.98
[13]	23	single-user	yes	320x240, 768x576(gray)	black long sleeve	no	0.81
[5]	19	single-user	no		colored glove	yes	0.91
[25]	6	single-user	yes	176x144, 640x480(rgb)		yes	0.99
Proposed	26	single-user	yes	176x144		yes	0.88

Table 3. Overview of related methods and comparison of average gesture recognition rates.

Setup	λ_{ANMM}	λ_{DD}	λ_{ROT}	ARR
single-user	0.0	0.4	0.6	0.883 ± 0.212
multi-user	0.5	0.0	0.5	0.761 ± 0.344
multi-user	0.0	0.4	0.6	0.567 ± 0.405
multi-user	0.333	0.333	0.333	0.731 ± 0.377
mixed[1]	0.333	0.333	0.333	0.896 ± 0.214

Table 1. Overview of letter recognition rates for different classifier combinations and setups *(ARR: average recognition rate and standard deviation)*. The error per letter for two setups is given in Fig. 6.

Setup	λ_{ANMM}	λ_{DD}	λ_{ROT}	ARR
single-user	0.0	0.4	0.6	0.936 ± 0.245
multi-user	0.5	0.0	0.5	0.878 ± 0.328
mixed[1]	0.333	0.333	0.333	0.964 ± 0.187

Table 2. Overview of word recognition rates for different classifier combinations and setups *(ARR: average recognition rate and standard deviation)*.

References

[1] M. Amin and H. Yan. Sign language finger alphabet recognition from gabor-pca representation of hand gestures. In *Machine Learning and Cybernetics*, 2007.

[2] P. Breuer, C. Eckes, and S. Müller. Hand gesture recognition with a novel ir time-of-flight range camera: a pilot study. In *MIRAGE*, pages 247–260. 2007.

[3] K. Derpanis, R. Wildes, and J. Tsotsos. Hand gesture recognition within a linguistics-based framework. In *European Conference on Computer Vision*, pages 282–296, 2004.

[4] P. Dreuw, H. Ney, G. Martinez, O. Crasborn, J. Piater, J. M. Moya, and M. Wheatley. The signspeak project - bridging the gap between signers and speakers. In *International Conference on Language Resources and Evaluation*, 2010.

[5] I. Incertis, J. Garcia-Bermejo, and E. Casanova. Hand gesture recognition for deaf people interfacing. In *International Conference on Pattern Recognition*, pages 100–103, 2006.

[6] D. Kelly, J. Mc Donald, and C. Markham. A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters*, 31:1359–1368, 2010.

[7] E. Kollorz, J. Penne, J. Hornegger, and A. Barke. Gesture recognition with a time-of-flight camera. *International Journal of Intelligent Systems Technologies and Applications*, 5:334–343, 2008.

[8] H. Lahamy and D. Litchi. Real-time hand gesture recognition using range cameras. In *Canadian Geomatics Conference*, 2010.

[9] Z. Li and R. Jarvis. Real time hand gesture recognition using a range camera. In *Australasian Conference on Robotics and Automation*, 2009.

[10] X. Liu and K. Fujimura. Hand gesture recognition using depth data. In *International Conference on Automatic Face and Gesture Recognition*, 2004.

[11] S. Liwicki and M. Everingham. Automatic recognition of fingerspelled words in british sign language. In *IEEE Workshop on CVPR for Human Communicative Behavior Analysis*, 2009.

[12] R. Lockton and A. W. Fitzgibbon. Real-time gesture recognition using deterministic boosting. In *British Machine Vision Conference*, 2002.

[13] J. Marnik. The polish finger alphabet hand postures recognition using elastic graph matching. In *Computer Recognition Systems 2*, volume 45 of *Advances in Soft Computing*, pages 454–461. 2007.

[14] Z. Mo and U. Neumann. Real-time hand pose recognition using low-resolution depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1499–1505, 2006.

[15] Q. Munib, M. Habeeb, B. Takruri, and H. Al-Malik. American sign language (asl) recognition based on hough transform and neural networks. *Expert Systems with Applications*, 32(1):24–37, 2007.

[16] S. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 27(6):873–891, 2005.

[17] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, pages 555–562, 1998.

[18] T. Pei, T. Starner, H. Hamilton, I. Essa, and J. Rehg. Learnung the basic units in american sign language using discriminative segmental feature selection. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4757–4760, 2009.

[19] J. Penne, S. Soutschek, L. Fedorowicz, and J. Hornegger. Robust real-time 3d time-of-flight based gesture navigation. In *International Conference on Automatic Face and Gesture Recognition*, 2008.

[20] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber. 3-d gesture-based scene navigation in medical imaging applications using time-of-flight cameras. In *Workshop On Time of Flight Camera based Computer Vision*, 2008.

[21] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[22] H. Takimoto, S. Yoshimori, Y. Mitsukura, and M. Fukumi. Classification of hand postures based on 3d vision model for human-robot interaction. In *International Symposium on Robot and Human Interactive Communication*, pages 292–297, 2010.

[23] S. Theodorakis, V. Pitsikalis, and P. Maragos. Model-level data-driven sub-units for signs in videos of continuous sign language. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2262–2265, 2010.

[24] M. Van den Bergh, E. Koller-Meier, and L. Van Gool. Real-time body pose recognition using 2d or 3d haarlets. *International Journal of Computer Vision*, 83:72–84, 2009.

[25] M. Van den Bergh and L. Van Gool. Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *IEEE Workshop on Applications of Computer Vision*, 2011.

[26] F. Wang and C. Zhang. Feature extraction by maximizing the average neighborhood margin. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[27] H.-D. Yang, S. Sclaroff, and S.-W. Lee. Sign language spotting with a threshold model based on conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1264–1277, 2009.

[28] Z. Zafrulla, H. Brashear, H. Hamilton, and T. Starner. A novel approach to american sign language (asl) phrase verification using reversed signing. In *IEEE Workshop on CVPR for Human Communicative Behavior Analysis*, pages 48–55, 2010.