Enhanced gesture-based human-computer interaction through a compressive sensing reduction scheme of very large and efficient depth feature descriptors

Tomás Mantecón, Ana Mantecón, Carlos R. del-Blanco, Fernando Jaureguizar, Narciso García

Abstract

In this paper, a hand gesture-based recognition system is presented with the aim of recognizing finger-spelling using the American Sign Language. The solution makes use of the depth imagery acquired by the new Kinect 2 sensor that provides more depth resolution. The main novelty is the introduction of a Compressive Sensing step to reduce the dimension of a depth-based feature descriptor, called Depth Spatiograms of Quantized Patterns, which is very discriminative, but also too large for its practical application. The system is composed by three steps: 1) depth-based feature descriptor computation that robustly characterizes the hand gesture; 2) Compressive Sensing based dimensionality reduction that shortens the previous highly discriminative but also large feature vector with almost no information lost; and 3) Support Vector Machine based classification that recognizes the performed hand gestures. Promising recognition results have been obtained in an American Sign Language based database.

1. Introduction

Thanks to the advent of new low-cost depth sensors, like the Kinect 2 sensor [1], the research activity in the field of gesture recognition is undergoing a significant increase, making possible the development of robust and efficient applications for human-Computer Interaction (HCI) [22], touchless interaction [23], human activity recognition, and sign language recognition [11]. The information provided by depth sensors is more relevant and convenient than the one provided by color cameras for applications of gesture recognition. The reason of this fact is that depth information provides more structural information about the human body parts, which allows to develop more discriminative gesture recognition algorithms. Moreover, the detection and segmentation tasks of human body parts in potentially complex backgrounds are also easier since depth imagery is almost immune to changes in illumination in indoor scenarios.

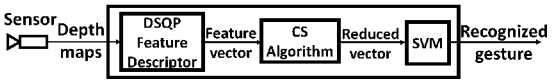

Figure 1. General scheme of the proposed solution.

Focusing on hand gesture recognition, some works have proposed to use the American Sign Language (ASL) as the dictionary of gesture to be recognized. Most of them are based only on the use of color imagery [8], but recently have appeared new contributions that use depth imagery [10], or a combination of both [15]. Regarding the recognition techniques, some solutions describe the hand gesture information using the output of a bank of Gabor filters, which are then deliver to a random forests classifier [15]. In other cases, intensity values are used directly as input to a classifier implemented via a Deep Belief Network [17]. The shape of the hand along with the relationship among the different parts of the hand have been also used as features to perform the gesture recognition using a Support Vector Machine (SVM) classifier [21]. In color imagery, the use of feature descriptors coming for the area of visual object recognition have been also commonly used, such as the Scale Invariant Feature Transform (SIFT) [19], the Histogram of Oriented Gradients (HOG) [9], and the Local Binary Patterns (LBP) [7].

The feature-based characterization of the hand gesture usually has a limited dimension for satisfying two purposes: first, improving the computational cost or memory requirements, and secondly to avoid the so-called 'curse of dimensionality' that can lead to a deterioration of the recognition performance. However, the limitation in length of the feature descriptors has also the side effect of a lower discriminative capability. To solve this problem, dimensionality reduction techniques along with higher dimensional feature descriptors can be used. Compressive Sensing (CS) theory is becoming highly popular for vision-based gesture recognition applications. It has been applied to color information [13], depth information [5],and both color and depth

information [20]. In many cases, they propose to generate a dictionary of gestures using a bag of features representation [3], and then to perform the recognition task by posing a l1-minimization problem. In other cases, they use CS to directly recognize the gesture using also a bag of features scheme [20]. Alternatively to CS for the dimensionality reduction task, other works have used the Singular Value Decomposition (SVD) technique [14], or even the combination of CS with Principal Component Analysis (PCA) [11]. The main advantage of CS techniques with respect to other dimensionality reduction techniques is that CS is not dependant of data distribution.

In this paper, a novel algorithm for hand gesture recognition using only depth information is presented. There are two main contributions of this paper. The first one is the design of a new and discriminative feature vector, called Depth Spatiograms of Quantized Patterns (DSQP), which is a major modification of the Local Binary Pattern (LBP) descriptor but is very logn. And the second one is the introduction of the Compressive Sensing (CS) technique to reduce the dimensionality of the DSQP vectors that are very long.The CS framework allows to reduce the length of the highly discriminative and long DSQP descriptors with almost no loss of information. The whole algorithm is composed by three steps: 1) DSQP based feature vector computation, 2) CS based dimensionality reduction, and 3) SVM based classification, as can be seen in Fig. 1.

The rest of the paper is organized as follows: in Section 2 a description of different features description techniques is presented, in Section 3 the compressive sensing technique is introduced, in Section 4 the classification process is outlined, next, the results are presented in Section 5, and the conclusions are drawn in Section 6.

2. DSQP feature descriptor

The DSQP feature descriptor is used to characterize the hand gesture information in depth imagery. This is a major LBP based modification [12], specially designed for depth imagery, which is more discriminative.

The LBP [12] is a feature descriptor technique widely used in many visual based systems for the task of hand gesture recognition. The main idea of this technique is compute the differences in intensity between a central pixel and each pixel in a neighborhood. This differences are then encoded as binary codes that represent the structure of local image regions. It is specially efficient in color imagery because of its robustness to illumination changes, and also due to its relatively low computational cost.

There are two major differences between the original LBP descriptor and the DSQP one. The first one affects to the relationships among the pixels in the neighborhood to be characterized. In LBP, the differences in intensity between a central pixel and each pixel in a neighborhood are computed, whereas in DSQP all the differences between each pair of pixels are used (see Fig. 2 for a neighborhood of $N_{neig} = 8$ pixels). Notice that the order in which the difference between a pair of pixels is computed is not relevant. For this reason, only one difference is considered, resulting in a total number of computed differences N_{diff}:

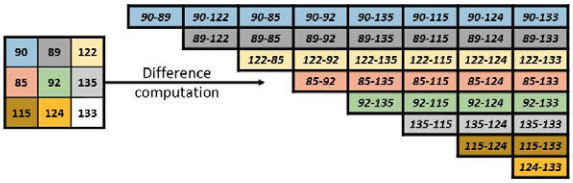

Figure 2. Difference computation for the DSQP descriptor specifying the differences that are taking into account.

$$N_{diff} = \frac{(N_{neig}+1)^2 - (N_{neig}+1)}{2} \quad (1)$$

The second difference is related with the way in which each difference value is encoded. In the case of the original LBP method, a thresholding process is used in such a way that each difference is represented by only 1 bit (similar to the sign function). However, the DSQP algorithm quantifies each difference using 3 bits. A non-uniform quantification scheme is used with $N_b = 3$ bits, resulting in 8 quantification intervals. Intervals are specially adapted to hand structures in depth imagery to take into account different hand poses (open, close, or with some fingers making certain gesture). These intervals cover the depth ranges between $(-35, -5)$ mm and $(5, 35)$ mm, which experimentally are the ones that contain the most representative information. Four intervals are used to uniformly cover each of these two depth ranges, making the descriptor highly discriminative to this key depth information. The intermediate depth range $(-5, 5)$ mm, which usually contains noisy depth variations, is quantified into a unique interval. Two more intervals are used to cover the last two depth ranges, $(-\infty, -35)$ and $(35, \infty)$. These two ranges represent large depth values that usually cover the border between the hand and the background. This is the reason why the precise depth value is not so relevant, but only the detection of such discontinuity.

Following the bag of features approach adopted in LBP, each set of neighborhood differences should be encoded in a decimal code, which then contributes to a histogram computed from all the decimal codes of a given image region. However, the resulting dimension of the histogram would be $2^{N_{diff} * N_b} = 2^{108}$, which is clearly prohibitive in memory requirements and computational cost. As a histogram with such a length is not tractable, a more feasible approach has been adopted to reduce its length. The adopted solution consists in dividing each 108-binary word into binary words of $N_{div} = 9$ bits. Using this solution, the new histogram is

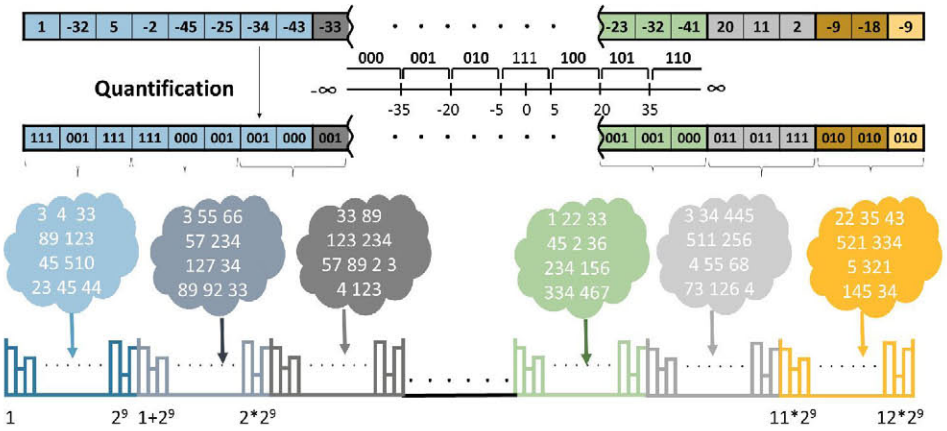

Figure 3. Computation of the DSQP histogram for one region of the depth map. It is composed by two stages: a non-uniform depth difference quantification using 3 bits per difference value, and a multiple histogram computation. In the second stage, multiple histograms are computed from a set of binary words with a length of 9 bits to characterize each depth region.

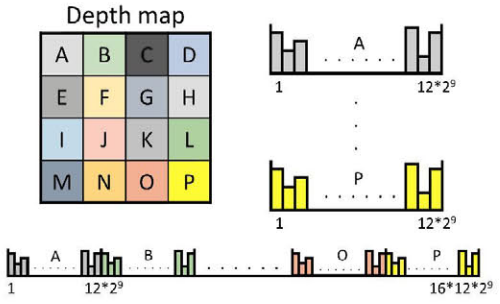

Figure 4. Block division of the depth map. One histogram is computed for each block, and the final feature descriptor is a stack of all block based histograms.

composed by $N_h = \frac{N_{diff} \times N_b}{N_{div}} = \frac{36 \times 3}{9} = 12$ histograms of $2^{N_{div}} = 2^9$ bins per histogram, and therefore the length of this new histogram is 12×2^9 bins (see Fig. 3).

A final step is carried out to add more spatial information to the resulting descriptor: the depth map is divided into $N_s \times N_s$ non-overlapping blocks. Considering the set of reasonable sizes of a the hand inside a depth map, a value around $N_s = 4$ divisions should be enough discriminative (see Fig. 4). The final DSQP feature descriptor is then computed as a concatenation of each histogram obtained for every block of the depth map. With this configuration, the total length of each descriptor can be expressed as follows:

$$N_{T-DSQP} = N_s \times N_s \times N_h \times 2^{N_{div}} = 98304 \quad (2)$$

Although the length of this feature vector is more tractable than one with $4 \times 4 \times 2^{108}$ bins (obtained without dividing the code words into smaller ones), it is also too long to be practically used as the input of an SVM classifier. For this reason a CS algorithm is applied in the next step to reduce its length.

3. Compressive sensing

With the purpose of reduce the high dimensionality of the DSQP feature descriptor, but keeping almost all the relevant information, a Compressive Sensing framework is proposed. Compressive sensing technique allows the reconstruction of sparse signals with fewer measurements than the ones requires by the Shannon-Nyquist criterion [6]. Those solutions are able to highly compress the information while preserving enough one to recover the original signal (or at least a good approximation). This process makes possible to reduce the computational cost and the memory requirements of the recognition step, while keeping all the discriminative power of the DSQP descriptors.

The CS algorithm uses a measurements matrix ϕ with dimensions $(M \times N)$ to obtain a lower dimensionality vector y of length M from a sparse vector x with dimension $N >> M$ (see figure 5):

$$y = \phi \cdot x. \quad (3)$$

The design of the matrix ϕ is one of the keys of the CS framework. To ensure that distance among the original vectors and the reduced ones is preserved, the matrix must satisfy the Restricted Isometry Property (RIP) [2]:

$$(1 - \delta_k) \|x\|_2^2 \leq \|\phi x\|_2^2 \leq (1 + \delta_k) \|x\|_2^2 \quad (4)$$

where $\|x\|_2$ is the Euclidean norm of x, and δ_k is the error in the vector distances after the projection ϕ.

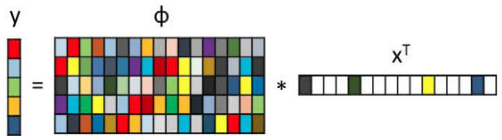

Figure 5. Example of a CS process in which x is the input and sparse vector, ϕ is the transform matrix and y the output vector.

In the literature, it is shown that different kind of matrices satisfy the RIP property. In this paper, for the design of ϕ, the Gaussian distribution matrix has been used, which is one of the most commonly used. Thus, a matrix of size $(M \times N)$ is computed, whose elements are i.i.d. samples of a normal Gaussian distribution $N(0, 1)$. This matrix satisfies the RIP property (Eq. 4) if the following relationship is achieved:

$$M \geq c \cdot K \cdot \log(N/K), \tag{5}$$

where M is the length of the output vector y, N is the length of the input vector x, K is the sparsity of the vector x (i.e. the number of elements that are not zero, or close to zero), and c is a constant of value $c = 1/2 \cdot \log(\sqrt{24}+1) \approx 0.38$.

The CS dimensionality reduction allows to obtain more accurate recognition rates since the classification task is carried out in a lower dimensionality space.

4. Classification Process

An SVM solution is proposed for the classification process, using the SVM Pegasos algorithm [18]. To achieve the goal of a multiple classification process, the algorithm has been configured with a one-vs-all strategy, using an SVM classifier per gesture. As the main goal is to be able to distinguish between all the gestures of the considered gesture dictionary, each SVM is trained with positive samples extracted from features vectors from one specific gesture, and with negative samples with features vectors from the other gestures. To improve the results obtained by the Pegasos solution, a Hellinger kernel, more commonly known as Battacharyya distance [4], has been applied:

$$k(h, h') = \sum_i \sqrt{h(i)h'(i)} \tag{6}$$

where h and h' are the CS-reduced feature descriptors of the test and training samples.

Regarding the dataset, it has been divided into training and test sets, where the 80% of the depth maps have been used for the training process, and the other 20% for the testing process.

5. Results

The propose hand-gesture recognition framework, called DSQP-CS-SVM, has been compared with a variation of itself consisting in discarding the CS step, called DSQP-SVM. Both solutions are also compared with the solution proposed in [16], from now on called PUGEAULT algorithm, which makes use of color and depth information to recognize different signs. Comparisons between those different methods have been made using the ASL hand gesture database [16] using only depth information. The database is composed by 24 gestures that represents 24 letters from the ASL. Each sign is recorded from 5 different subjects (non-native to sign language). For each gesture, over 500 samples are recorded, so the entire database is composed by around 48000 samples.

To be able to compare the PUGEAULT solution with the one proposed in this paper, some parameters have been selected. For the configuration of the DSQP-SVM algorithm, the following parameters have been set: $N_{neig} = 8$, $N_b = 3$, $N_{div} = 9$ and $N_s = 4$. The same parameters have been selected for the DSQP-CS-SVM algorithm, with the addition of the parameter M (that defines the length of the compressed vector) according to Eq. 5. For the selection of that parameter two parameters related to the input data, the first one is the length of the DSQP vectors, which is $N = 98304$ according to Eq. 2, the other parameter is the sparsity of those vectors, examining all DSQP vectors computed for different images, a value of $K = 3848$ has been used as it is the mean sparsity value of all DSQP vectors. Finally, the number of elements of the compressed vector has been determined as the minimun value of Eq. 5, so $K = 3848$ has been used to obtain the results.

For the comparison between different solutions, the confusion matrix (CM) is used. This metric is widely used in object recognition problems. The aim of the presented solutions is to recognize the sign that has been performed, independently of the subject. For this reason, each column of the CM represent the number of signs that belongs to each class, and each row is the number of signs recognized to each class (positive and negative ones). Two different measures are used to test different algorithms. The first one is the normailzed accuracy measure, that is the main diagonal of the CS, that is obtained using the following equation:

$$\text{Accuracy} = \frac{\text{Total number of correct signs}}{\text{Total number of signs per class}}. \tag{7}$$

The other one is the F-Score that represents the relationship between precision and recall and is obtained using the following equation:

$$\text{F-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Sign	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a	0,83	0,07	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0	0	0,01	0,01	0	0	0	0	0	0,02	0	0	0	0	0
b	0	0,9	0,14	0	0	0,01	0	0	0	0	0	0	0	0	0	0	0	0	0	0,01	0	0	0	0
c	0,01	0	0,81	0,03	0,02	0	0	0	0	0	0	0	0	0	0,01	0,02	0	0,01	0	0	0	0	0	0
d	0,01	0	0	0,94	0,02	0,01	0,01	0,01	0,01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0,01	0	0,92	0,03	0	0	0	0	0	0	0	0	0	0	0	0,01	0	0	0	0	0	0
f	0,01	0	0	0	0	0,91	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0	0	0	0	0	0	0	0	0	0	0
g	0,01	0	0	0	0	0	0,88	0,04	0,01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,02
h	0	0	0,01	0	0	0	0,02	0,92	0,02	0	0	0	0	0	0	0,01	0	0	0	0	0	0	0	0
i	0	0	0,01	0,01	0	0	0	0	0,93	0,13	0,01	0,01	0	0	0	0	0	0	0	0	0	0	0	0,01
k	0	0	0	0	0	0	0	0	0	0,85	0,1	0,07	0,06	0,06	0,06	0,06	0	0	0	0	0	0	0	0
l	0	0	0	0	0	0	0	0	0	0	0,85	0,01	0	0	0	0	0	0	0	0	0,02	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0,85	0,04	0,01	0	0	0	0,02	0,04	0,01	0,01	0,01	0,01	0
n	0,01	0	0	0	0	0	0	0	0	0	0	0,02	0,83	0,02	0	0	0,01	0	0,16	0	0	0	0	0
o	0,01	0	0,01	0	0	0	0	0	0	0	0	0	0	0,88	0,03	0,01	0	0,01	0,02	0	0	0	0	0
p	0	0	0	0	0,01	0	0,01	0	0	0	0	0	0	0,01	0,84	0,05	0,01	0	0	0	0	0	0	0,03
q	0,05	0	0	0	0	0	0	0,01	0	0	0	0	0	0	0,03	0,84	0,21	0,01	0,01	0	0	0	0	0
r	0	0	0	0,01	0	0,01	0	0	0	0,01	0	0	0	0	0	0	0,75	0,24	0,14	0,14	0,15	0,14	0,14	0,14
s	0,02	0	0	0	0	0	0,02	0	0	0	0,01	0,01	0	0	0	0	0	0,67	0,03	0	0	0	0	0
t	0,02	0	0	0	0	0	0	0	0	0,01	0	0,01	0,03	0,01	0	0	0	0,01	0,55	0,01	0,01	0	0	0
u	0,03	0,02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,81	0,04	0,03	0,02	0
v	0	0	0	0	0	0,01	0	0	0	0	0,01	0	0	0	0	0	0	0	0	0,02	0,77	0,12	0,02	0,01
w	0	0	0	0	0	0	0,01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,7	0	0,01
x	0	0	0	0	0	0	0	0	0	0	0,01	0	0,01	0	0,03	0,01	0	0	0,01	0	0	0	0,78	0,01
y	0	0	0,01	0	0	0	0,01	0	0	0	0	0	0	0	0	0	0	0	0,01	0	0	0	0	0,77

Table 1. Confusion matrix obtained using the DSQP-CS-SVM algorithm.

Sign	DSQP-SVM		DSQP-CV-SVM		PUGEAULT	
	Acc.	F-S	Acc.	F-S	Acc.	F-S
a	**0,91**	0,73	0,83	**0,82**	0,75	0,38
b	0,88	0,78	**0,9**	**0,88**	0,83	0,47
c	0,75	0,82	**0,81**	**0,84**	0,57	0,47
d	**0,95**	0,89	0,94	**0,94**	0,37	0,33
e	0,69	0,77	**0,92**	**0,94**	0,63	0,61
f	0,87	0,90	**0,91**	**0,91**	0,35	0,36
g	**0,94**	**0,91**	0,88	0,90	0,6	0,61
h	0,85	0,84	**0,92**	**0,93**	0,8	0,83
i	**0,93**	0,80	**0,93**	**0,89**	0,73	0,61
k	0,73	0,66	**0,85**	**0,75**	0,43	0,55
l	**0,87**	0,86	0,85	**0,90**	**0,87**	0,77
m	0,77	0,58	**0,85**	**0,85**	0,17	0,25
n	0,52	0,60	**0,83**	**0,81**	0,23	0,29
o	0,72	0,73	**0,88**	**0,90**	0,13	0,18
p	0,81	0,77	**0,84**	**0,86**	0,57	0,56
q	0,47	0,58	**0,84**	0,78	0,77	**0,82**
r	0,58	0,42	**0,75**	0,53	0,63	**0,60**
s	0,56	0,68	**0,67**	**0,77**	0,17	0,17
t	0,37	0,46	**0,55**	**0,67**	0,07	0,09
u	0,77	0,76	**0,81**	**0,83**	0,67	0,73
v	0,64	0,72	0,77	**0,79**	**0,87**	0,70
w	**0,73**	0,79	0,7	**0,81**	0,53	0,68
x	0,6	0,72	**0,78**	**0,85**	0,2	0,27
y	0,74	0,83	**0,77**	**0,86**	**0,77**	0,82
Mean	0,74	0,73	**0,82**	**0,83**	0,53	0,51

Table 2. Comparision of accuracy results using different algorithms: DSQP-SVM, DSQP-CS-SVM and PUGEAULT [16].

Table 1 shows the confusion matrix for all the signs of the database using the DSQP-CS-SVM algorithm. These results shown small confusion values between among signs taking into account the amount of different signs. Also, it can be observed that the performance of some of them is very similar, i.e. stable. For the most of the signs, an accuracy over 0.7 (70%) is achieved.

Table 2 shows both accuracy and F-Score results for the three compared solutions. The proposed solutions, DSQP-SVM and DSQP-CS-SVM, achieve better results using both measurements than the PUGEAULT one, in both mean value and variance of both measurements results. Also, the proposal that uses the CS algorithm to reduce the dimension of the original DSQP descriptors obtains the best recognition score, indicating that most of the information is preserved in the compressed feature vector.

6. Conclusions

A hand-gesture based recognition system is presented that recognizes different finger-spelling gestures using the American Sign Language in depth imagery. The key contribution is the introduction of a CS framework to reduce the length of the high dimensional DSQP descriptor, which enables to reduce memory requirements, computational cost, and at the same time to increase the recognition accuracy. The promising recognition scores using CS strategy proves the efficiency of the presented recognition framework.

7. Acknoledgments

This work has been partially supported by the Ministerio de Economía y Competitividad of the Spanish Government under projects TEC2010-20412 (Enhanced 3DTV) and TEC2013-48453 (MR-UHDTV), and by AIRBUS Defense and Space under the project SAVIER.

References

[1] Microsoft Corporation. Kinect for Xbox 360 http://dx.doi.org/10.1007/s10107-010-0420-4.

[2] D. Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671 – 687, 2003. Special Issue on PODS 2001.

[3] A. Boyali and M. Kavakli. A robust gesture recognition algorithm based on sparse representation, random projections and compressed sensing. In *7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 243–249, July 2012.

[4] E. Choi and C. Lee. Feature extraction based on the bhattacharyya distance. *Pattern Recognition*, 36(8):1703 – 1709, 2003.

[5] P. Cirujeda and X. Binefa. 4dcov: A nested covariance descriptor of spatio-temporal features for gesture recognition in depth sequences. In *2nd International Conference on 3D Vision (3DV)*, volume 1, pages 657–664, Dec 2014.

[6] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok. Introduction to compressed sensing. *Preprint*, 93, 2011.

[7] Y. Ding, H. Pang, X. Wu, and J. Lan. Recognition of hand-gestures using improved local binary pattern. In *International Conference on Multimedia Technology (ICMT)*, pages 3171–3174, July 2011.

[8] A. Kindiroglu, H. Yalcin, O. Aran, M. Hrz, P. Campr, L. Akarun, and A. Karpov. Automatic recognition fingerspelling gestures in multiple languages for a communication interface for the disabled. *Pattern Recognition and Image Analysis*, 22(4):527–536, 2012.

[9] J. Konečný and M. Hagara. One-shot-learning gesture recognition using HOG-HOF features. *Journal of Machine Learning Research*, 15(1):2513–2532, Jan. 2014.

[10] A. Kuznetsova, L. Leal-Taixe, and B. Rosenhahn. Real-time sign language recognition using a consumer depth camera. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 83–90, Dec 2013.

[11] S.-Z. Li, B. Yu, W. Wu, S.-Z. Su, and R.-R. Ji. Feature learning based on sae-pca network for human gesture recognition in RGBD images. *Neurocomputing*, 151, Part 2(0):565 – 573, 2015.

[12] T. Ojala, M. Pietikinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

[13] S. Poularakis, G. Tsagkatakis, P. Tsakalides, and I. Katsavounidis. Sparse representations for hand gesture recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3746–3750, May 2013.

[14] R. Ptucha and A. Savakis. LGE-KSVD: Flexible Dictionary Learning for Optimized Sparse Representation Classification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 854–861, June 2013.

[15] N. Pugeault and R. Bowden. Spelling it out: Real-time asl fingerspelling recognition. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1114–1119, Nov 2011.

[16] N. Pugeault and R. Bowden. Spelling it out: Real-time asl fingerspelling recognition. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1114–1119, Nov 2011.

[17] L. Rioux-Maldague and P. Giguere. Sign language fingerspelling classification from depth and color images using a deep belief network. In *Canadian Conference on Computer and Robot Vision (CRV)*, pages 92–97, May 2014.

[18] Y. Singer and N. Srebro. Pegasos: Primal estimated subgradient solver for svm. In *ICML*, pages 807–814, 2007.

[19] P. Sykora, P. Kamencay, and R. Hudec. Comparison of SIFT and SURF methods for use on hand gesture recognition based on depth map. *AASRI Procedia*, 9(0):19 – 24, 2014. AASRI Conference on Circuit and Signal Processing (CSP 2014).

[20] J. Wan, Q. Ruan, W. Li, and S. Deng. One-shot learning gesture recognition from rgb-d data using bag of features. *Journal of Machine Learning Research*, 14:2549–2582, 2013.

[21] Y. Wang and R. Yang. Real-time hand posture recognition based on hand dominant line using kinect. In *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–4, July 2013.

[22] H. Zhang, V. Patel, M. Fathy, and R. Chellappa. Touch gesture-based active user authentication using dictionaries. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 207–214, Jan 2015.

[23] L. Zhou and H. Leung. Human motion retrieval based on sparse coding and touchless interactions. In *International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)*, pages 417–418, Nov 2013.