Hand Gesture Recognition Using Infrared Imagery Provided by Leap Motion Controller

Tomás Mantecón(✉), Carlos R. del-Blanco, Fernando Jaureguizar, and Narciso García

Grupo de Tratamiento de Imágenes, E.T.S.I. de Telecomunicación,
Universidad Politécnica de Madrid, Madrid, Spain
{tmv,cda,fjn,narciso}@gti.ssr.upm.es

Abstract. Hand gestures are one of the main alternatives for Human-Computer Interaction. For this reason, a hand gesture recognition system using near-infrared imagery acquired by a Leap Motion sensor is proposed. The recognition system directly characterizes the hand gesture by computing a global image descriptor, called Depth Spatiograms of Quantized Patterns, without any hand segmentation stage. To deal with the high dimensionality of the image descriptor, a Compressive Sensing framework is applied, obtaining a manageable image feature vector that almost preserves the original information. Finally, the resulting reduced image descriptors are analyzed by a set of Support Vectors Machines to identify the performed gesture independently of the precise hand location in the image. Promising results have been achieved using a new hand-based near-infrared database.

Keywords: Feature extraction · Gesture recognition · Random projections · Image classification · Near-infrared imaging

1 Introduction

The number of works in the field of gesture recognition have increased considerably during the last years, boosting the Human-Machine Interaction (HMI). This is due to the advent of low-cost sensors that are able to obtain multimodal information from the scene. This is the case of Kinect 2 that provides depth, color, and skeletal information; Senz3D that can acquire depth and color information; Intel Realsense that provides depth, color, and skeletal information; and Leap Motion that can capture near-infrared and skeletal information. Some works have employed this kind of sensors to improve the interaction between a human and a robot. For example, the work presented in [7] controls the motion of a robot by vision. In [22] the payload is managed, while in [19] a robot is operated by using a remote connection for rescue situations. In other cases, the recognition system can be used to allow a car driver to interact with the radio [10], or to be able to answer phone calls while driving [11]. It can be also used for rehabilitation purposes by reproducing the arm movement [6].

J. Blanc-Talon et al. (Eds.): ACIVS 2016, LNCS 10016, pp. 47–57, 2016.
DOI: 10.1007/978-3-319-48680-2_5

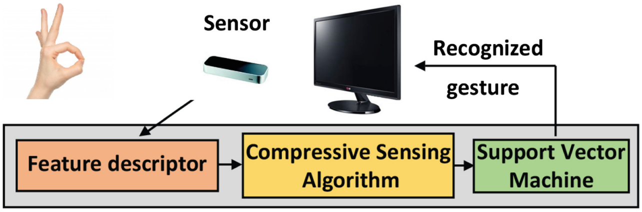

Fig. 1. General system of the proposed solution.

In comparison with other hand-based HMI sensors, Leap Motion controller has been scarcely exploited for HMI. Possibly, the two main reasons are: (1) it is a close range sensor that can only sense the hands, but not the human body; (2) it is strongly oriented to process hand-skeletal information, discarding raw imagery data. This last issue can be further attributed to two factors. The first one is that the Leap Motion Software Development Kit is oriented to provide high level hand information, such as fingertip position and velocity, palm orientation, etc. But, there is a lack of functionalities to process the near-infrared raw imagery, such as hand detection or segmentation. Moreover, the second factor is related to the two cameras embedded in the Leap Motion sensor which have a very wide field of view that introduces strong image distortions, which complicates even more their processing.

Due to the aforementioned factors, the researcher activity with the Leap Motion sensor has been mainly limited to the use of the hand skeleton information. In [23], they propose to use the palm trajectory (information given by the skeleton) to recognize on-air written numbers by means of a Support Vector Machine (SVM) classifier. In [17], the signature of a person is proposed to check his identity by processing skeleton-derived trajectories. For this purpose, a combination of 3D Histogram of Oriented Optical Flow (HOOF) and Histogram of Oriented Trajectories (HOT) are used as input feature vectors to an SVM classification stage. Multiple Leap Motion sensors have also been proposed in [9] to process the acquired hand skeletons using a framework based on Hidden Markov Models (HMM).

Alternatively, some works use the skeletal information of the Leap Motion with the color or depth imagery provided by other HMI sensors, mainly to avoid to process the highly geometrically distorted infrared images. For example, [15] that uses color imagery acquired by Kinect sensor to complement the hand skeleton provided by the Leap Motion. In [24], a Leap Motion skeleton is combined with the information provided by the accelerometers of a wearable watch, which is also used to give some feedback via vibration signals.

Although there are no works addressing the hand gesture recognition task using directly the geometrically distorted infrared images of the Leap Motion (as far as the authors' knowledge), the use of color and depth information for hand gesture recognition has been quite investigated. Based on machine learning approaches, different descriptors have been used with different classification techniques and probabilistic/Bayesian frameworks. For example, in [2], they propose a recognition solution based on the hand shape and its motion combined with a Hidden Markov Model (HMM) to control a robot. Other solutions make use of Fourier descriptors with Recurrent Neural Networks (RNN) [16]. In [4], they use a combination of Gabor features with SVM classifiers to perform hand gesture recognition under varying illumination scenes. Many other solutions have adopted popular object feature descriptors such as the Scale Invariant Feature Transform (SIFT) [21], the Histogram of Oriented Gradients (HOG) [12], the Local Binary Patterns (LBP) [8], and different LBP variations [13].

In this paper, a novel algorithm for hand gesture recognition using just the near-infrared imagery of the Leap Motion controller is presented. This paper considers that the aforementioned drawbacks of the Leap Motion sensor can be considered as strong advantages from other point of view. More precisely, the close range of the Leap Motion is ideal for reducing the background clutter, allowing to directly focus on the hand patterns. On the other hand, a specific machine learning framework is proposed, whose performance is not deteriorated by the geometrical distortions introduced by the Leap Motion cameras. Within this framework, there is no detection process as the feature descriptor is computed over the whole image, so no preprocessing is needed and the final computational cost is not incremented. Figure 1 shows the stages of the entire gesture recognition system: (1) computation of a DSQP based feature vector, (2) dimensionality reduction based on Compressive Sensing, and (3) classification based on a bank of SVMs.

The organization of the paper is as follows. Section 2 describes the feature extraction algorithm. Section 3 describes the Compressive Sensing (CS) algorithm. In Sect. 4, the classification process is presented. Section 5 describes the Leap Motion controller. Section 6 summarizes the obtained results with the proposed algorithm. Finally, conclusions are drawn in Sect. 7.

2 DSQP-Based Feature Description

To achieve the characterization of the hand gesture information in near-infrared imagery, some modifications have been made over the Depth Spatiograms of Quantized Patterns (DSQP) feature descriptor presented in [14], that is an evolution of the LBP algorithm [18]. There are two main differences between the original LBP and the DSQP descriptor. One is related to the relationship among pixels in a neighborhood. Instead of computing differences between the central pixel and each pixel in the neighborhood (LBP), the DSQP descriptor computes differences between each pair of pixels within the neighborhood, as it can be seen in Fig. 2, where $N_{neig} = 8$ pixels is chosen. Only one of the two possible difference values between two pixels is considered, obtaining the following total number of differences:

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

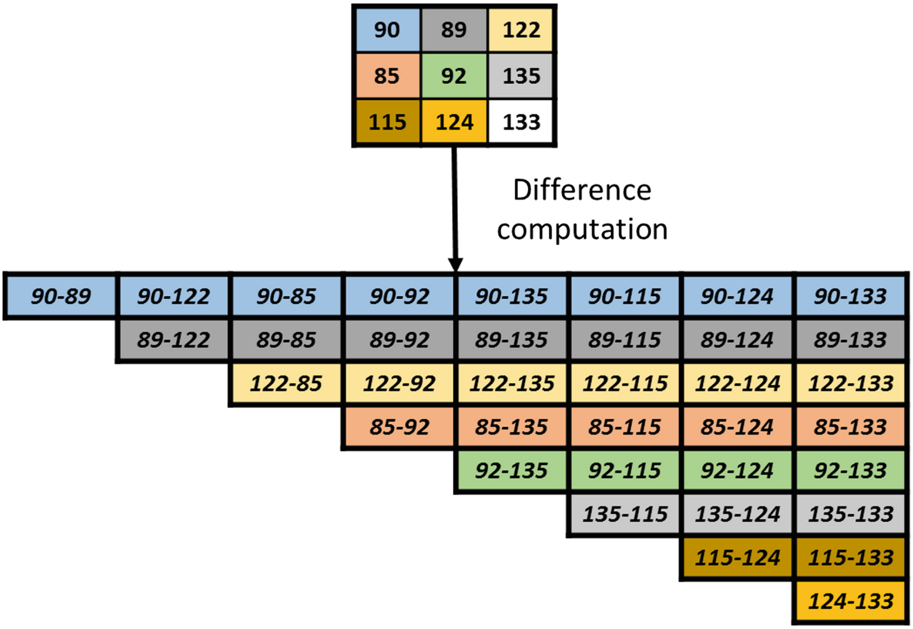

Fig. 2. Difference computation for the DSQP descriptor, specifying the differences that are taken into account.

The second difference affects the quantization process of each difference value. Instead of using a thresholding process in which the difference is represented by just 1 bit (encoding the sign), the DSQP algorithm uses $N_b = 3$ bits for the quantization of the difference following a non-uniform scheme, which results in 8 intervals of quantization. Since the DSQP descriptor is typically used with depth imagery, an adaptation is performed for infrared images, where the decision values are tuned to acquire the range of infrared difference values. The reason is that the nature and depth resolution of the infrared imagery is fundamentally different from the depth imagery (it is not directly related to the 3D hand structure).

Likewise LBP, DSQP also uses a bag of features approach. Each set of neighborhood differences are encoded into a decimal number, which contributes to a histogram that compactly represents all the decimal codes of an image region. Taking into account the previous configuration, the resulting dimension of the histogram is $2^{N_{diff} \times N_b} = 2^{108}$, which is clearly prohibitive for both, computational cost and memory requirements. To obtain a more tractable histogram, an alternative approach has been considered. It consists in dividing the long binary word into binary sub-words of $N_{div} = 6$ bits, computing then a histogram for every set of sub-words. As can be seen in Fig. 3, this result have the following number of histograms:

$$N_h = \frac{N_{diff} \times N_b}{N_{div}} = \frac{36 \times 3}{6} = 18 \tag{2}$$

Which are composed of the following number of bins:

$$2^{N_{div}} = 2^6 = 64 \tag{3}$$

The last step consists in dividing the image into $N_s \times N_s$ non-overlapping blocks, and computing a DSQP descriptor for each image block. This strategy adds more spatial information to the resulting descriptor. For the considered resolution images, a value of $N_s = 6$ has been fixed, which experimentally has achieved good recognition results. The performed block division is higher than the original DSQP configuration because only one descriptor is used to encode the whole image, independently of the exact hand location within the image, and therefore additional spatial information is desirable.

The final DSQP-based image descriptor is obtained as a concatenation of all DSQP descriptors coming from the different image blocks. Considering this configuration, the final length of the image descriptor is:

$$N_{T\_DSQP} = N_s \times N_s \times N_h \times 2^{N_{div}} = 41472 \tag{4}$$

Although the length of the final vector has been considerably reduced by spitting the binary words into sub-words, it is still too long to be efficiently used as the input of an SVM classifier. For this reason, a CS-based dimensionality reduction algorithm is considered in the next section.

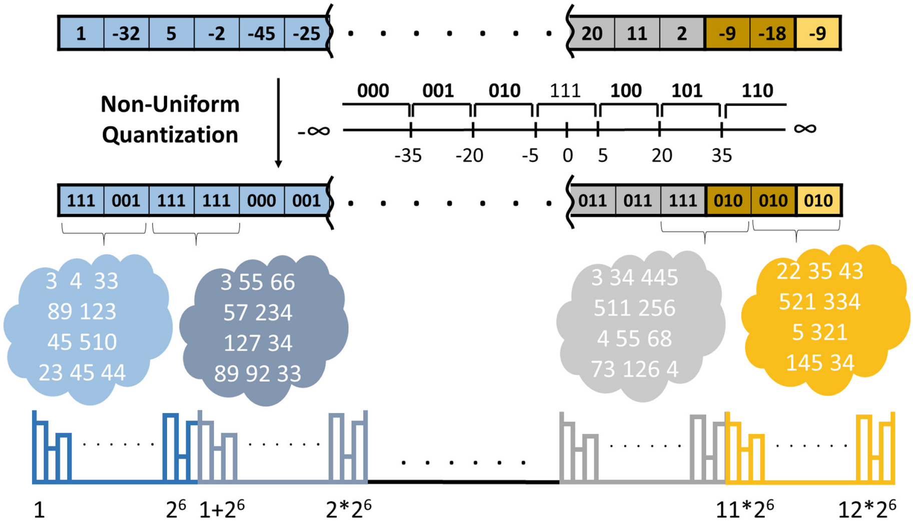

Fig. 3. Computation of the DSQP descriptor for an image region.

3 Compressive Sensing

As the dimension of the DSQP feature vector is quite high, a solution based on the CS framework is used to reduce its dimensionality, but preserving almost all the intrinsic information. Based on the CS paradigm, as the original vector of dimension N is sparese in some domain, it is possible to obtain a much lower dimensionality vector of dimension M. The vector of reduced dimensionality y is obtained by projecting the original vector x in a subspace by means of a matrix ϕ of dimensions $M \times N$, usually called measurement matrix. One of the keys of the CS algorithm is the design of the ϕ matrix as the distance between the original vectors and the reduced ones needs to be preserved. This is the reason why that matrix must satisfy the Restricted Isometry Property (RIP) [1], which can be expressed as follows:

$$(1-\delta_k)\left\|x\right\|_2^2 \leq \left\|\phi x\right\|_2^2 \leq (1+\delta_k)\left\|x\right\|_2^2 \tag{5}$$

where $\left\|x\right\|_2$ is the euclidean norm of the original vector x, $\left\|\phi x\right\|_2$ is the euclidean norm of the reduced vector ϕx, and $\delta_k \in (0,1)$ is the error in the vector distances after the projection ϕ.

Although the RIP theorem does not say how to obtain such a ϕ matrix, it has been proven in the literature that random matrices satisfy the RIP condition. More specifically, a matrix which elements are random realizations of a Bernoulli distribution is used in the present implementation, where the RIP is satisfied if:

$$M \geq c \times K \times \log(N/K) \tag{6}$$

where c is a constant with a value close to 0.3 [5], and K is the number of non-zero elements of x.

The decision of using the CS algorithm, instead of other dimensionality reduction algorithms (Principal Component Analysis (PCA) or Singular Value Decomposition (SVD)) was based on the computational cost. Both PCA and SVD algorithms needs to make expensive operations using all features of all gestures. In the case of CS algorithm, just a multiplication between each feature vector and the measurement matrix ϕ, whose number of rows is considerable less than the total number of features.

After applying the CS framework to the DSQP vectors the computational cost and memory requirements are decreased.

4 Classification Process

For the classification process, an SVM solution based on the SVM Pegasos algorithm [20] is proposed. For the purpose of multiple gesture recognition, a one-vs.-all strategy is used, where one SVM classifier is trained for each gesture. The main objective is to be able to distinguish between different hand gestures in the dataset, and therefore each SVM is trained using as positive samples those containing the considered gesture, and as negative samples the other gestures samples. A non-linear Hellinger kernel, more commonly known as Battacharyya coefficient [3], is used to improve the recognition score

$$k(h, h') = \sum_i \sqrt{h(i)h'(i)} \tag{7}$$

where h and h' are the test and train feature descriptors respectively.

Regarding the training and evaluation procedure, the database images have been divided into training and testing sets, 50 % for training and 50 % for testing process.

5 Leap Motion Controller

The Leap Motion controller is a small USB peripheral device designed for HMI that supports hand and finger tracking without physical contact. As it can be seen in Fig. 4, the sensor has its own LEDs that emit infrared light, and two wide-angle monochromatic infrared cameras that receive the light reflected by the objects close to the Leap Motion. The Leap Motion SDK computes the 3D position of some hand key elements by a proprietary algorithm that uses a stereo-matching technique over the infrared images and a predefined hand skeleton model. Thus, the Leap Motion SDK is able to describe hands, fingers, palm, wrist, arms, and tools (similar to fingers but thinner). Regarding the characteristics of the infrared images, they have a resolution of 640 × 240 and a frame rate of 60 frames per second and per camera. Figure 5 shows some examples of captured infrared images.

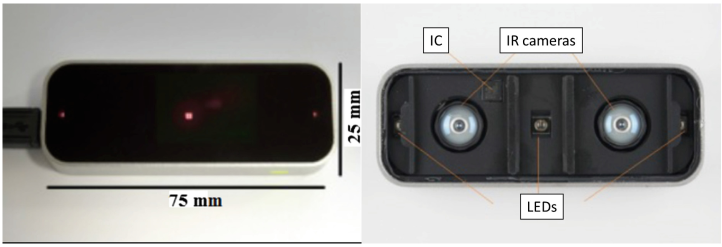

Fig. 4. Leap Motion controller showing where the LEDs and infrared cameras are placed.

6 Results

As far as the authors' knowledge, there is no hand database using the infrared information provided by the Leap Motion. This is the reason for which some samples of different hand poses have been acquired using a Leap Motion sensor placed over a table, the subjects sat close to it, and moved their right hand over the sensor at a distance between 10 and 15 cm in front of it. A group of 10

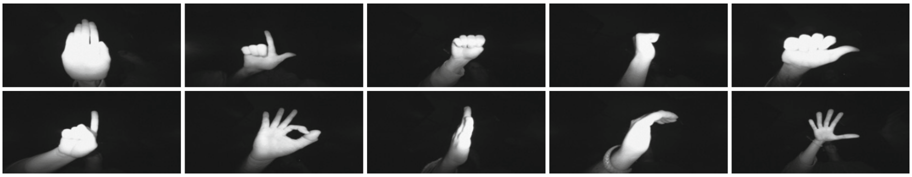

Fig. 5. Dataset samples.

different gestures that were performed by 10 different subjects (5 women and 5 men) have been acquired. A total number of 200 frames have been recorded for each gesture and subject. In Fig. 5, a sample of each hand gestures performed by different subjects is shown. In the first row from left to right, there are an open palm parallel to the sensor (Palm), a closed palm with the thumb and index fingers extended (L), a palm closed (Fist), a fist perpendicular to the sensor (Fist_m), and a palm closed with the thumb extended (Thumb). In the second row from left to right, there are a palm closed with the index extended (Index), an open palm with the index and thumb making a circle (OK), an open palm perpendicular to the sensor (Palm_m), a semi close palm in a shape like a 'C' (C), and an open palm with all its fingers separate (Palm_d).

To test those recorded images, the proposed hand gesture recognition system has been compared with the one proposed by Huang in [4]. This solution has three main stages: the first one is a segmentation stage to determine where the hand is placed in the image; a second stage computes image descriptors using a bank of Gabor filters; and the final stage also uses SVM classification algorithm for the recognition. For the comparison, the same parameters proposed by the original work has been used.

The following parameters have been considered for the configuration of the proposed solution. For the DSQP descriptor, $N_{neigh} = 8$, $N_b = 3$, $N_{div} = 6$, and $N_s = 6$ have been selected. For the CS algorithm, the parameter M has been selected according to Eq. 6. Two additional parameters related with the input data are needed to obtain M, the length of the DSQP vector that is $N = 41472$ according to Eq. 4, and the sparsity of input vectors that in mean value is $K = 15352$; with these parameters a final value of $M = 4579$ has been obtained.

To obtain quantitative results, the confusion matrix (CM) has been used. As the aim of this work is to recognize different gestures independently of the subject, each column of the CM represents the percentage of gestures that belongs to each class, and each row represents the number of gestures, positive and negative, recognized to each class. From CM matrix two measurements can be directly observed, the precision and the recall, which can be expressed as follows:

$$\text{Precision} = 100 \times \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{8}$$

Table 1. Confusion Matrix with result for the proposed system.

	Palm	L	Fist	Fist_m	Thumb	Index	OK	Palm_m	C	Palm_d
Palm	100	0	0	0	0	0	0	0	0	0
L	0	99,9	0	0	0	0	0	0	0	0
Fist	0	0	99,8	0	0	0	0,3	0	0	0
Fist_m	0	0	0	100	0	0	0	0	0	0
Thumb	0	0,1	0,2	0	100	0	0	0	0	0
Index	0	0	0	0	0	100	0	0	0	0
OK	0	0	0	0	0	0	99,7	0	0	0
Palm_m	0	0	0	0	0	0	0	100	0	0
C	0	0	0	0	0	0	0	0	100	0
Palm_d	0	0	0	0	0	0	0	0	0	100

$$\text{Recall} = 100 \times \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{9}$$

In Table 1, the CM results obtained by using the proposed algorithm. As it can be seen, for all the gestures the precision, elements of the main diagonal, are over 99 % and just a small percentage of samples are detected as belonging to other gestures, but an amount less than 0.5 %. This indicates that our algorithm is quite accurate in both measurements, precision and recall.

Along with the CM values, the F-Score measure is also used to compare the results of both solutions. This measure is obtained as follows:

Table 2. Accuracy results for the proposed system and [4].

Gest.	Alg.	
	Proposed	[4]
Palm	**1**	0,95
L	0,99	**1**
Fist	**0,99**	0,87
Fist_m	**1**	**1**
Thumb	**0,99**	0,95
Index	**1**	**1**
OK	**0,99**	0,87
Palm_m	**1**	0,95
C	**1**	0,87
Palm_d	**1**	0,87
Mean	**0,99**	0,94

$$\text{F-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{10}$$

Table 2 shows the accuracy results of the proposed solution and the one proposed by Huang in [4]. The proposed solution is the one that achieves better results. In addition, the presented solution does not need a segmentation stage as the one used by Huang, decreasing significantly the complexity. This comparison also allows us to notice that a segmentation stage is not so important within this database.

7 Conclusions

A hand gesture recognition system for near-infrared images acquired by the Leap Motion has been presented. The system computes a DSQP-based image descriptor directly, without any hand segmentation stage. The resulting image descriptor is reduced in dimension by applying a CS framework. Finally, the obtained reduced vectors are delivered to a bank of SVMs that perform the gesture recognition. The promising obtained recognition scores prove the efficiency of the presented recognition framework, and claim a higher prominence of the Leap Motion sensor for future HMI applications.

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

References

1. Achlioptas, D.: Database-friendly random projections: Johnson-Lindenstrauss with binary coins. J. Comput. Syst. Sci. **66**(4), 671–687 (2003)
2. Aditya, R., Namrata, V., Santanu, C., Subhashis, B.: Recognition of dynamic hand gestures. Pattern Recogn. **36**(9), 2069–2081 (2003)
3. Choi, E., Lee, C.: Feature extraction based on the Bhattacharyya distance. Pattern Recogn. **36**(8), 1703–1709 (2003)
4. Deng-Yuan, H., Wu-Chih, H., Sung-Hsiang, C.: Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. Expert Syst. Appl. **38**(5), 6031–6042 (2011)
5. Eldar, Y.C., Kutyniok, G.: Compressed Sensing: Theory and Applications. Cambridge University Press, Cambridge (2012)
6. Gieser, S.N., Boisselle, A., Makedon, F.: Real-time static gesture recognition for upper extremity rehabilitation using the leap motion. In: Duffy, V.G. (ed.) DHM 2015. LNCS, vol. 9185, pp. 144–154. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21070-4_15
7. Tran, T.T.H.: How can human communicate with robot by hand gesture? In: International Conference on Computing, Management and Telecommunications, pp. 235–240, January 2013
8. Jiang, F., Wang, C., Gao, Y., Wu, S., Zhao, D.: Discriminating features learning in hand gesture classification. IET Comput. Vis. **9**(5), 673–680 (2015)

9. Kai-Yin, F., Ganganath, N., Chi-Tsun, C., Tse, C.: A real-time ASL recognition system using Leap Motion sensors. In: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 411–414, September 2015
10. Kim, J., Ryu, J., Han, T.: Multimodal interface based on novel HMI UI/UX for in-vehicle infotainment system. ETRI J. **37**(4), 793–803 (2015)
11. Kopinski, T., Geisler, S., Handmann, U.: Gesture-based human-machine interaction for assistance systems. In: IEEE International Conference on Information and Automation, pp. 510–517, August 2015
12. Kuizhi, M., Lu, X., Boliang, L., Bin, L., Fang, W.: A real-time hand detection system based on multi-feature. Neurocomputing **158**, 184–193 (2015)
13. Mantecon, T., del Blanco, C.R., Jaureguizar, F., Garcia, N.: New generation of human machine interfaces for controlling UAV through depth-based gesture recognition. In: Proceedings of the SPIE, vol. 9084, May 2014
14. Mantecon, T., Mantecon, A., del Blanco, C., Jaureguizar, F., Garcia, N.: Enhanced gesture-based human-computer interaction through a compressive sensing reduction scheme of very large and efficient depth feature descriptors. In: IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 1–6, August 2015
15. Marin, G., Dominio, F., Zanuttigh, P.: Hand gesture recognition with Leap Motion and Kinect devices. In: IEEE International Conference on Image Processing, pp. 1565–1569, October 2014
16. Ng, C.W., Ranganath, S.: Real-time gesture recognition system and application. Image Vis. Comput. **20**(1314), 993–1007 (2002)
17. Nigam, I., Vatsa, M., Singh, R.: Leap signature recognition using HOOF and HOT features. In: IEEE International Conference on Image Processing, pp. 5012–5016, October 2014
18. Ojala, T., Pietikinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recogn. **29**(1), 51–59 (1996)
19. Shang, W., Cao, X., Ma, H., Zang, H., Wei, P.: Kinect-based vision system of mine rescue robot for low illuminous environment. J. Sens. **2016**, 1–9 (2016)
20. Singer, Y., Srebro, N.: Pegasos: primal estimated sub-gradient solver for SVM. In: ICML, pp. 807–814, October 2007
21. Sykora, P., Kamencay, P., Hudec, R.: Comparison of SIFT and SURF methods for use on hand gesture recognition based on depth map. In: AASRI Conference on Circuit and Signal Processing, vol. 9, pp. 19–24, September 2014
22. Tornow, M., Al-Hamadi, A., Borrmann, V.: Gestic-based human machine interface for robot control. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 2706–2711, October 2013
23. Yuanrong, X., Qianqian, W., Xiao, B., Yen-Lun, C., Xinyu, W.: A novel feature extracting method for dynamic gesture recognition based on support vector machine. In: IEEE International Conference on Information and Automation, pp. 437–441, Jul 2014
24. Zhang, P., Li, B., Du, G., Liu, X.: A wearable-based and markerless human-manipulator interface with feedback mechanism and kalman filters. Int. J. Adv. Robot Syst. **12**, 164–170 (2015)