HAGR-D: A Gesture Recognition System based on CIPBR Algorithm

Diego Santos, Bruno Fernandes, Byron Leite
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Recife, Brazil
http://www.ecomp.poli.br/

***Abstract*—Gesture recognition has been an area of great interest and study in recent years due to the evolution of technology and computers processing power, generating a higher degree in the Interaction Human Computer (IHC). These advances now allow communication between man and machine through hand gestures or entire body, especially in games, after the advent of Microsoft Kinect and other depth sensors. This paper proposes a dynamic gesture recognition system for user hand. The system is evaluated in two bases of dynamic hand gestures from the literature. The experiments show that the proposed model overcomes other algorithms presented in the literature in hand gesture recognition tasks, achieving a classification rate of 97.49% in the MSRGesture3D dataset and 98.43% in the RPPDI dynamic gesture dataset.**

***Keywords*-CIPBR, HMM, DTW, Gesture Recognition**

I. Introduction

Gestures are the main way of deaf-mute people communication and they are present in most of the communication between all kind of people. Gestures can express several and different emotions, expressions among other things [1]. Sign Languages use manual communication and body language instead of sound patterns.

Nine millions of people in the world cannot communicate through sound patterns due some hearing or speaking disability [2]. This group of people makes use of sign language to communicate. The sign language is a set of gestures with a meaning that can represent the language signs of a community.

As every language, sign language needs to be understood by people who surround the deaf-mute ones such as family, friends, sellers, customers and others. Systems based on computer vision can use a camera to capture the gestures performed by people and translate them using computer vision techniques so that anyone can understand.

Several efforts have been applied to perform a real-time translation between different sound pattern languages around the world [3], [4]. Likewise, many works have developed techniques to translate sign language in real time. Oreifej and Liu [5] proposed a technique called Histogram of Oriented 4D Normals (Hon4d) that uses a 4D histogram approach for feature extraction, while Yang [6] proposed an algorithm for 2D and 3D spaces that extracts some features from the executed gestures: the location of the left hand with respect to the signer's face in 3D space; the angle from the face to the left hand; the position of the left hand with respect to the shoulder center; the occlusion of both hands. Doliotis *et al.* [7] proposed a feature extraction method using images generated by a Microsoft Kinect, retrieving a 3D pose orientation and full hand configuration parameters.

Contributions: In this paper, we propose a novel approach for dynamic gesture recognition with depth maps called hybrid approach for gesture recognition with depth maps (HAGR-D). Our system uses a version of CIPBR (convex invariant position based on RANSAC) algorithm [8] for feature extraction, a combination of the binary particle swarm optimization [9] and a selector algorithm to make the feature selection and a hybridization between DTW and HMM classifiers for recognition. DTW is used to find the most probable gestures, while HMM refines DTW output. This system is the result of a research developed during a master course resulting in a dissertation entitled "Um sistema para reconhecimento de gestos baseado no Algoritmo Ransac Convexo" and the articles presented in Section V.

This paper is organized as follows. Section II describes the proposed model. In Section III, experiments with gesture images captured by the Microsoft Kinect are shown. Finally, in Section IV, we present some concluding remarks[1] .

II. Hybrid Approach for Gesture Recognition

Our hybrid approach is composed by some techniques described in the following subsections.

A. CIPBR

The Depth CIPBR algorithm is an approach composed of a sequence of tasks to reduce a depth map of a hand posture into two signature sets proposed by Keogh et al. [10]. To complete these tasks, there are four modules connected in cascade as presented in Figure 1.

The first module, "Radius Calculation," uses a hand posture image that is segmented from the depth map generated by the Microsoft Kinect, Figure 2.(a). The hand posture contour is

[1]This work is a master dissertation entitled "Um sistema para reconhecimento de gestos baseado no Algoritmo Ransac Convexo", presented on Universidade de Pernambuco. This master dissertation is part of a research project called: "Um Sistema de Reconhecimento de Gestos para Dispositivos Móveis", with aims to build a tool that is capable of recognizing gestures and synthesize them into sounds and being able to identify sounds converting them into text or signals. Thus, communication between deaf people with normal hearing and may occur with flow in both directions.

Fig. 1. Depth CIPBR architecture.

extracted from this image generating the contour, Figure 2.(b), and the center of mass (C) of the hand posture is calculated from the image contour using the center moments [11]. Then, the point that has the lower Y coordinate is found, P, Figure 2.(c). Finally, this module calculates the distance between the center of mass and point P. Figure 2.(d) presents an output example of the "Radius Calculation" module. The dark gray point is the center of mass of the contour given by C, the red point is the highest point of the contour given by P, and the line connecting these points is given by $\overline{PC}$.

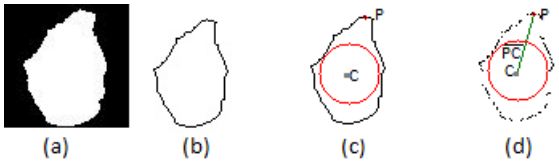

Fig. 2. (a) The hand segmented using the Microsoft Kinect; (b) The hand posture contour; (c) The mass center point drawn in the hand posture contour (dark gray point); (d) The convex hull points with the maximum circumcircle Θ (red circle), the center mass point(dark gray point), the highest point (red point) and the segment of line $\overline{PC}$ (green line)

The second module of Depth CIPBR, "Draw Maximum Circumcircle," uses the line segment $\overline{PC}$ as radius to draw a circle inside the hand contour. If this circle exceeds the hand contour boundary, a triangle is calculated using the three most distant contour points from the point C, being two of them in opposite sides of the contour. The biggest circle inside this triangle is the maximum circumcircle Θ of the contour with the center in point C.

The third module of Depth CIPBR, "Calculate Signatures," receives the maximum circumcircle Θ and points P and C as input. The hand contour points are substantially reduced using the Andrew's monotone chain convex hull algorithm [12].

Andrew's algorithm outputs a set $\Psi = \{p_1, p_2, \ldots, p_n\}$ from Convex hull points, which is used to generate two signature sets. The first signature set is composed of distances (D) calculated as follows:

- For each point $\omega \in \Psi$, the length of the line segment $\overline{\omega C}$ is calculated based on the Euclidean distance from ω to the point C;
- Then, this length is subtracted from the circumcircle radius, in order to obtain the $\overline{\omega Q}$ length, where point Q is the intersection between segment $\overline{\omega C}$ and Θ.

Therefore, the first signature set is composed of each distance $D_{\overline{\omega Q}}$, $\forall \omega \in \Psi$, calculated using the following equation (1):

where:

- C is the center of mass of the hand posture contour;
- ω_x, C_x are the x coordinates for points ω and C respectively;
- ω_y, C_y are the y coordinates for points ω and C, respectively;
- $radius$ is the radius of Θ calculated by the "Draw Maximum Circumcircle" module.

The second signature set consists of a vector of angles obtained by calculating the angle between a line composed of each point $w \in \Psi$ of the convex hull hand shape point and the line segment $\overline{PC}$. Both signature sets are obtained in a clockwise direction always starting with point P.

Finally, in the last module, "Feature Vector Normalization," the signature sets are normalized. The first signature set is normalized dividing each distance by the radius calculated in the "Draw Maximum Circumcircle" module. The normalized distance vector is represented by $D' = \{d'_1, d'_2, \ldots, d'_n\}$:

$$d'_i = \frac{d_i}{radius}. \quad (1)$$

The set of angles is normalized by dividing each angle by 360°:

$$a'_i = \frac{a_i}{360°}. \quad (2)$$

Angle and distance sets are concatenated in the following order: angles first and distances at the end of the signature vector. Therefore, the final feature vector is $F = \{a'_1, a'_2, \ldots, a'_n, d'_1, d'_2, \ldots, d'_n\}$.

B. Feature Selection Method

Some classifiers used for gesture recognition are more sensitive to the curse of dimensionality [13], such as the HMM [14], [15]. In order to overcome this obstacle, Feature Selection method finds the smallest size possible for the feature vector and assigns the same size for the feature vectors of all gestures. This is also an important task since many classifiers use inputs with the same predefined size.

In this study, Binary Particle Swarm Optimization [9] (BPSO) finds the target size of the reduced feature vector, while the Selector algorithm is used to resize the feature vectors. The objective that BPSO seeks to optimize is the minimum distance between the particle composed of 0's and 1's and the gestures sequences. The number of 1's in the particle denotes the size of the new feature vector.

The next subsections explain in detail how these algorithms work.

1) Particle Swarm Optimization: Particle swarm optimization [16] solves an optimization problem with a swarm of simple computational elements, called particles, exploring a solution space to find an optimal solution. The position from each particle represents a candidate solution in n-dimensional search space (D) defined as $X = \{x_1, x_2, x_3, \ldots, x_n\}$, where each x_n is a position in the n-dimension, and the particle velocity is represented by $V = \{v_1, v_2, v_3, \ldots, v_n\}$.

The fitness function evaluates how well each particle presents itself in each iteration. When a particle moves and

its new position has a better fitness value than the previous one, this value is saved in a variable called p_{best}. To guide the swarm to the best solution, the position, where a single particle found the best solution until the current execution, is stored in a variable called g_{best}. Therefore, to update the particle velocity and position, the following equations are used:

$$v_i(t+1) = \kappa v_i(t) + c_1 r_1[p_{i,best} - x_i(t)] + c_2 r_2[g_{best} - x_i(t)] \quad (3)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

where $i = (1, 2, 3, \ldots, N)$, N is the size of the swarm, c_1 represents the private experience or "cognitive experience" and c_2 represents the "social experience" interaction, usually used with a value of 2.05 [16]. Variables r_1 and r_2 are random numbers between zero and one and represent how much p_{best} and g_{best} will influence the particle movement. The inertia factor κ is used to control the balance of the search algorithm between exploration and exploitation. The x_i represents the particle position in the i-th dimension. The recursive algorithm runs until the maximum number of iterations is reached.

2) Binary PSO: The binary PSO is a variation of the traditional PSO in discrete spaces. The major difference between this algorithm and its canonical version is the interpretation of velocity and position. In the binary version, the particle's position and velocity are represented by zeros and ones only. This change requires a reformulation in how velocity is calculated, according to the following equation:

$$\begin{aligned} &If\ rand\ <\ \frac{1}{1+e^{-v_i(t+1)}}\ then \\ &x_i(t+1) = 1;\ else\ x_i(t+1) = 0 \end{aligned} \quad (5)$$

where $rand$ is a random number between zero and one.

Finally, to binarize all of the feature vectors, a threshold calculated through the mean of all of the feature vectors is used. BPSO calculates a distance from each x_{ij} binary particle's position to the same j position in all binary vectors for the same gesture. After each iteration, all distances are added up to generate the fitness function output. Particles are improved as soon as the fitness values become smaller in comparison with the fitness obtained by the previous iteration. The particle fitness function is:

$$fitness_i = \sum_{j=1}^{m}[\sqrt{\sum_{k=1}^{n}(x_{ik} - F_{jk})^2}] \quad (6)$$

where $(x_{i1},\ x_{i2},\ \ldots,\ x_{in})$ is the particle's i-th position and $(F_{j1},\ F_{j2},\ \ldots,\ F_{jn})$ is the j-th features in all vectors.

3) Selector Algorithm: BPSO chooses the target size for the reduced feature vector S'. Then, the selector algorithm [17] reduces the CIPBR feature vector S to S', producing the final vectors of the proposed approach. In this process, some rules must be respected. First, if any vector has fewer points than the target size of S', zeros are added to the feature vector until it matches the desired length. Second, feature vectors larger than the target size of S' are redefined using a selection algorithm. This algorithm consists of calculating a window W through the division of the current vector length by the target size of S'. The current vector S is parsed, and each value in the W position is included in the new feature vector. If the new output vector S' is even smaller than the desired length, the remaining positions are randomly visited in S and used to compose the new output vector S' until the desired length is reached.

III. Experiments

In order to evaluate our proposed system, two experiments are performed with public benchmarks: the MSRGesture3D dataset [18] and the RPPDI dynamic gesture dataset [19]. The next subsections explain these experiments in detail.

A. MSRGesture3D

The MSRGesture3D is a dynamic hand gesture dataset captured by the Kinect RGB-D camera. There are 12 dynamic hand gestures defined by American Sign Language (ASL) in MSRGesture3D, and each dynamic gesture was performed two or three times by each one of 10 subjects. The gestures presented in the dataset are: "bathroom", "blue", "finish", "green", "hungry", "milk", "past", "pig", "store", "where", "J" and "Z'.' The dataset contains only depth data images and is considered challenging mainly because of self-occlusion issues. We used the leave-one-subject-out cross-validation to evaluate the dataset as proposed in [18].

The following parameters are used for training the HAGR-D: noitemsep

- DTW: $k = 5$;
- BPSO: noitemsep
  - 15 particles;
  - 20 dimensions;
  - 30 simulations;
  - 200 iterations;
  - $c1 = c2 = 2.05$;
  - inertia factor of $w = 0.9 \rightarrow 0.4$;
  - $r1 = r2 = rand(0...1)$;
- HMM: noitemsep
  - three states;
  - 100 iterations.

To find the initial states of HMM, we use a k-means clustering [20] technique, avoiding the random initial matrices, while the BPSO uses few dimensions in its particles to guarantee small final vectors. The works in [21], [22] use similar approaches to determine the final size of the vectors in their studies.

We compared our system with a model using the same feature extraction approach, but with DTW or HMM alone as classifiers, "called depth CIPBR + DTW" and "depth CIPBR + HMM". Therefore, we are able to evaluate the performance of the proposed model with DTW before HMM and solely with HMM. The inputs for DTW are the raw sequences generated by depth CIPBR, and the inputs for the HMM are the output sequences of the feature selection method. Figure 3

presents the boxplot for each method. It is easy to see that the hybridization between DTW and HMM significantly improved the proposed model. Furthermore, in several iterations, our hybrid approach classification rate was very close to 100%, and only two sequences were misclassified, corresponding to the gesture "green" being classified as "store" and the gesture "blue" being classified as "where". Another point to be made is that outliers presented using solely HMM as a classifier no longer exist with our approach. The size of the boxplot generated by the results of each classifier also elucidate a low variation between the results of each classifier, providing a certainty about the consistency of the results.

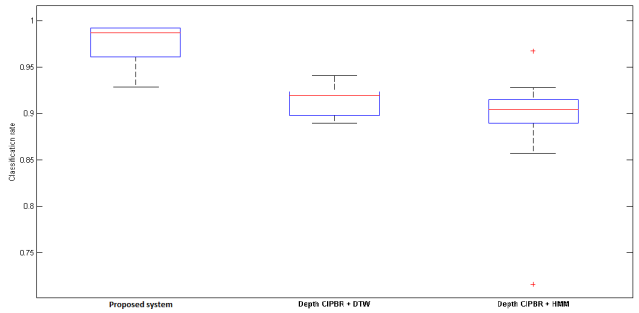

Fig. 3. Comparison between our proposed system, depth CIPBR with DTW and depth CIPBR with HMM for hand gesture recognition.

Our proposed system and "depth CIPBR + DTW", respectively, applied to the MSRGesture3D dataset present low number of mistakes. The hybridization of the "depth CIPBR + DTW" with HMM generating an improvement of the classification rates, and most of mistakes happened between the gestures "green" and "store", with 7% of the "green" being classified as "store".

Figure 4 shows a representation of the two gesture vectors most confused by our system and how close they are. To represent this gesture in a 2D space, they are normalized in size using the "selector algorithm" with only two features as the final length. It is easy to see that sometimes, a few examples cross the division between classes, making classification more difficult.

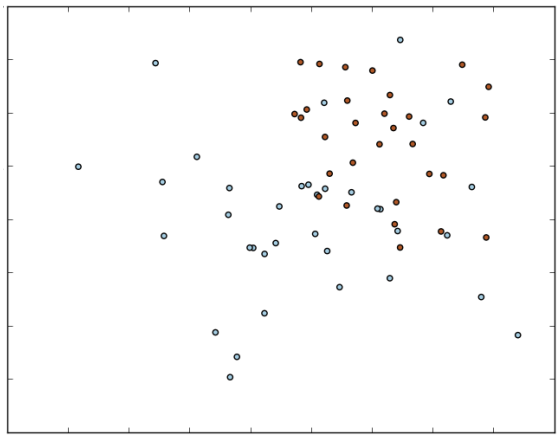

Fig. 4. Example of two generated classes by the CIPBR algorithm.

Finally, Table I presents the results obtained for the MSGesture3D dataset using the leave-one-subject-out cross-validation as a testing procedure in comparison with other methods in the literature. The proposed system achieved the best classification result of 97.49%.

Method	(%)
HAGR-D	97.49
Depth CIPBR + DTW	91.53
Depth CIPBR + HMM	88.98
Actionlet [23]	95.29
Histogram of Oriented 4D Normals (HON4D) + Discriminative Density (D_{disc}) [5]	92.45
HON4D [5]	87.29
Random Occupancy Patterns (ROP), Wang *et al.* [18]	88.50
Depth motion maps, Yang *et al.* [24]	89.20
Kurakin *et al.* [25]	87.70
Klaser *et al.* [26]	85.23

TABLE I
COMPARISON BETWEEN THE RESULTS FOR MSRGESTURE3D WITH THE LEAVE-ONE-SUBJECT-OUT CROSS-VALIDATION AS THE CLASSIFICATION PROCESS.

Venkateswara *et al.* [27] use the same dataset in their study, but modifying the experiment using five subjects for training and five for testing their methods, achieving 94.6% as their best result, which is still above ours.

B. RPPDI Dynamic Gesture Dataset

The RPPDI dynamic gesture dataset is a set of images of seven dynamic hand gestures performed in front of a smartphone camera. Figure 8 illustrates one sequence example for each gesture in the dataset. Each gesture is performed several times, and Table II presents the number of sequences in each gesture. We used the same test configuration as proposed by Barros *et al.* [19], [17] with $2/3$ of the dataset for training the model and $1/3$ for testing.

Fig. 5. Example for each of the gestures performed on the RPPDI dynamic gesture dataset.

Gesture	**Number of Sequences**
Gesture 1	24
Gesture 2	24
Gesture 3	31
Gesture 4	18
Gesture 5	26
Gesture 6	33
Gesture 7	32

TABLE II
NUMBER OF SEQUENCES CAPTURED BY EACH GESTURE IN THE RPPDI DYNAMIC DATASET.

The experiments in the RPPDI dataset used the same configuration for BPSO and HMM as presented in Section III-A, and the results are compared to Barros *et al.*'s previous works. The CIPBR uses the Otsu threshold [28] as a binarization method in the first module to segment hand posture. Table 6 presents the results obtained, and Table IV presents the confusion matrix of the HAGR-D system. The proposed method committed a few mistakes, misclassifying only one gesture, while achieving 100% accuracy in some iterations.

Method	**Classification Rate (%)**
HAGR-D	98.43
SURF + HMM [17]	75.00
LCS + HMM [17]	77.00
Convex SURF (CSURF) + HMM [17]	91.00
Convex LCS (CLCS) + HMM [17]	91.00
SURF + DTW [17]	38.00
LCS + DTW [17]	78.00
CSURF + DTW [17]	93.00
CLCS + DTW [17]	97.00

TABLE III
COMPARISON BETWEEN THE RESULTS IN RPPDI DYNAMIC GESTURE DATASET.

Gest.	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	100%	-	-	-	-	-	-
2	-	100%	-	-	-	-	-
3	-	-	89%	-	-	11%	-
4	-	-	-	100%	-	-	-
5	-	-	-	-	100%	-	-
6	-	-	-	-	-	100%	-
7	-	-	-	-	-	-	100%

TABLE IV
CONFUSION MATRIX OF RPPDI DYNAMIC GESTURE DATABASE CLASSIFIED BY THE HAGR-D SYSTEM.

C. Remarks

The proposed approach, achieved the best results in two different datasets due to the combination of depth CIBPR for feature extraction and the hybrid classifier with DTW and HMM. The classifiers compensated the failures of each other by reducing misclassifications between different gestures with DTW and refining the classification output through validation of the most similar sequences using the HMM. The hybrid classifier improved the results in comparison with DTW and HMM applied individually.

One of the limitations of our system is the definition of the number of sequences returned by DTW, k. In this study, such a parameter was empirically defined. Another limitation is the computation cost of DTW that might impair real-time application of the proposed model.

Another point to be made is that the few mistakes committed by the proposed system were due to very similar sequences. Nevertheless, many conditions must be fulfilled in order for HAGR-D to misclassify a given gesture: the number of similar postures, the distance from the hand to the sensor, the speed of gesture execution and occlusion.

IV. CONCLUSION

Hand gesture recognition have been a challenge for real time applications due the need of requirements of robustness, accuracy and efficiency. In this Master dissertation, we proposed both a variation of the CIPBR algorithm for depth maps and a hybrid classifier for gesture recognition using DTW and HMM. The HAGR-D, presents better results than the ones in the literature, achieving a classification rate of 97.49% in the MSRGesture3D dataset and 98.43% in the RPPDI dynamic gesture dataset. The application of depth CIPBR for feature extraction showed good results, while the hybridization between the DTW and HMM classifiers significantly improved classification accuracy.

Although the focus of classification in this paper relies on the task of hand gesture recognition, in future research, we intend to extend our proposed system to other types of gestures, such as human body movements. Furthermore, the DTW has a high computational cost, which makes the system execution slow; however, the FastDTW [29] is a variation of the traditional form of the DTW that promises to exponentially reduce the computational cost, and it will be addressed in our next experiments.

V. PUBLICATIONS

This work has yielded three publications, listed below:

Title: The Dynamic Gesture Recognition System based on CIPBR Algorithm

Authors: Diego D.S. Santos, Rodrigo C. Neto, J.T. Bruno Fernandes, and Byron L. D. Bezerra.

Published: MICAI 2014. Algorithm. Polibits, v. 50, p. 13-19, 2014.

Url: http://www.scielo.org.mx/pdf/poli/n50/n50a3.pdf

Title: FSSGR: Feature Selection System to Dynamic Gesture Recognition

Authors: Diego D.S. Santos, Rodrigo C. Neto, J.T. Bruno Fernandes, and Byron L. D. Bezerra.

Published: Image Based Smart City Applications (ISCA) 2014.

Url:http://link.springer.com/chapter/10.1007/978-3-319-23222-529

Title: HAGR-D: A Novel Approach for Gesture Recognition with Depth Maps

Authors: Diego D.S. Santos, Bruno Fernandes J.T., and Byron L. D. Bezerra.

Published: Sensors (Basel), v. 15, p. 28646-28664, 2015.

Uri:http://www.mdpi.com/1424-8220/15/11/28646

ACKNOWLEDGMENT

The authors would like to thank the support of the funding agency FACEPE for all support in this project.

REFERENCES

[1] E. Backer, "Cluster analysis by optimal decomposition of induced fuzzy sets," Ph.D. dissertation, TU Delft, Delft University of Technology, 1978.

[2] N. Praveen, N. Karanth, and M. Megha, "Sign language interpreter using a smart glove," in *Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on*. IEEE, 2014, pp. 1–5.

[3] J.-P. van Arnhem, "Google translate app," *The Charleston Advisor*, vol. 17, no. 3, pp. 24–27, 2016.

[4] W. D. Lewis, "Skype translator: Breaking down language and hearing barriers," *Proceedings of Translating and the Computer (TC37)*, 2015.

[5] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 716–723.

[6] H.-D. Yang, "Sign language recognition with the kinect sensor based on conditional random fields," *Sensors*, vol. 15, no. 1, pp. 135–147, 2014.

[7] P. Doliotis, V. Athitsos, D. Kosmopoulos, and S. Perantonis, "Hand shape and 3d pose estimation using depth data from a single cluttered frame," in *International Symposium on Visual Computing*. Springer, 2012, pp. 148–158.

[8] D. G. Santos, R. C. Neto, B. J. Fernandes, and B. L. Bezerra, "A dynamic gesture recognition system based on cipbr algorithm," *Polibits*, no. 50, pp. 13–19, 2014.

[9] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 5. IEEE, 1997, pp. 4104–4108.

[10] E. Keogh, L. Wei, X. Xi, S.-H. Lee, and M. Vlachos, "Lb_keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures," in *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 882–893.

[11] C. E. Clauser, J. T. McConville, and J. W. Young, "Weight, volume, and center of mass of segments of the human body," DTIC Document, Tech. Rep., 1969.

[12] A. Day, "Planar convex hull algorithms in theory and practice," in *Computer graphics forum*, vol. 7, no. 3. Wiley Online Library, 1988, pp. 177–193.

[13] S. Bilal, R. Akmeliawati, M. J. El Salami, and A. A. Shafie, "Vision-based hand posture detection and recognition for sign languagea study," in *Mechatronics (ICOM), 2011 4th International Conference On*. IEEE, 2011, pp. 1–6.

[14] S. Chafik and D. Cherki, "Some algorithms for large hidden markov models," *World Journal Control Science and Engineering*, vol. 1, no. 1, pp. 9–14, 2013.

[15] S. Okada, O. Hasegawa, and T. Nishida, "Machine learning approaches for time-series data based on self-organizing incremental neural network," in *International Conference on Artificial Neural Networks*. Springer, 2010, pp. 541–550.

[16] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1. IEEE, 2000, pp. 84–88.

[17] P. V. Barros, N. Junior, J. M. Bisneto, B. J. Fernandes, B. L. Bezerra, and S. M. Fernandes, "Convexity local contour sequences for gesture recognition," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 34–39.

[18] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 872–885.

[19] P. V. Barros, N. T. Júnior, J. M. Bisneto, B. J. Fernandes, B. L. Bezerra, and S. M. Fernandes, "An effective dynamic gesture recognition system based on the feature vector reduction for surf and lcs," in *Artificial Neural Networks and Machine Learning–ICANN 2013*. Springer, 2013, pp. 412–419.

[20] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[21] R. Raghavendra, B. Dorizzi, A. Rao, and G. K. Hemantha, "Pso versus adaboost for feature selection in multimodal biometrics," in *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*. IEEE, 2009, pp. 1–7.

[22] G. Deepa, R. Keerthi, N. Meghana, and K. Manikantan, "Face recognition using spectrum-based feature extraction," *Applied Soft Computing*, vol. 12, no. 9, pp. 2913–2923, 2012.

[23] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, "Real time action recognition using histograms of depth gradients and random decision forests," in *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2014, pp. 626–633.

[24] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 1057–1060.

[25] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 1975–1979.

[26] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *BMVC 2008-19th British Machine Vision Conference*. British Machine Vision Association, 2008, pp. 275–1.

[27] H. Venkateswara, V. N. Balasubramanian, P. Lade, and S. Panchanathan, "Multiresolution match kernels for gesture video classification," in *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–4.

[28] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[29] S. Salvador and P. Chan, "Fastdtw: Toward accurate dynamic time warping in linear time and space," in *KDD workshop on mining temporal and sequential data*. Citeseer, 2004.