A Dynamic Gesture Recognition System based on CIPBR Algorithm

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

***Abstract*—Dynamic gesture recognition has been studied actually for it big application in several areas such as virtual reality, games and sign language. But some problems have to be solved in computer applications, such as response time and classification rate, which directly affect the real-time usage. This paper proposes a novel algorithm called Convex Invariant Position Based on Ransac which improved the good results in dynamic gesture recognition problem. The proposed method is combined with a adapted PSO variation to reduce features and a HMM and three DTW variations as classifiers.**

***Index Terms*—Gesture recognition, computer vision, CIPBR, dynamic time wrapping, hidden Markov model.**

I. Introduction

HAND recognition gestures systems brought to several areas a more natural way for Human-Computer Interaction (HCI). Applications in sign language recognition [1], virtual reality [2], and computer games [3] have increasingly used hand gesture recognition approaches to facilitate the learning based on intuition to remember and performer a gesture.

There are three different categories for systems based in human gesture recognition: systems based on the hand gesture captured by gloves or external sensors [4], system which make the device tracking to generate a gesture path [5]. The last category use a camera to capture the gesture in images and extract features from it using computer vision techniques to interpret the gesture [6], [7].

The first class, hand gesture captured by gloves or external sensors, uses some devices connected in the user hand what turn difficult the natural articulation of hand gesture. This category has the advantage to being invariant to light conditions and complex background, achieving a better result than others applications but is more expensive, due do the high price of the devices to develop a system based on it.

The second category, is the more limited class, by the limited number of gestures which can be recognized. Most of this systems are presented in portable devices where predefined gesture execute some action. This gestures complexity is very low in compared with others categories.

The last category, system based on computer vision techniques, uses a camera to capture the hand gesture in images or videos and extract attributes and features, such as position, velocity, color, among others to identify the gesture.

Manuscript received on July 28, 2014, accepted for publication on September 22, 2014, published on November 15, 2014.

The authors are with Universidade de Pernambuco, Brazil. Corresponding author: Diego G.S. Santos (e-mail: dgs2@ecomp.poli.br).

This class has been grown by the present facility to obtain a camera, usually coupled in a smart-phone. Another reason for this growth, is the non-invasive techniques which do not require any devices connected and nothing which hinders the natural hand gesture movement.

Systems based on vision computer techniques usually involve two steps: feature extraction and pattern classification. This paper presents a novel technique called Convexity Invariant Position Based on Ransac (CIPBR) for hand gesture image feature extraction as first module in a dynamic hand gesture system. In addition we used two classifiers to evaluate the proposed method: Dynamic Time Wrapper (DTW) [8] and Hidden Markov Model (HMM) [9].

This paper is structured as follows: Section 2 presents some related works. Section 3 describes the CIPBR algorithm. Section 4 presents the Classifier module based on HMM. Section 5 presents the Classifier module based on DTW. Section 6 presents the Experiments Results. Finally, in Section 7, the conclusions and some future work are given.

II. Related Works

Several techniques have been used in hand gesture recognition systems. Meena [10] and El-Salwah [11] use the Local Contour Sequence (LCS) algorithm to reduce the hand posture into a distances vector with the better contour points.

Calinon et al. [12] use probabilistic techniques as Principal Components Analysis (PCA) and Independent Components (ICA) to recognize and reproduce gestures.

The Speed Up Robust Features (SURF) used for Bao et al. [13] extract points of interest sets over pyramidal images using a Gaussian Laplacian [14] and recognize dynamic gestures making a track path with SURF features.

Some of those algorithms previously cited lose their accuracy when a new test image has a different rotation from the ones observed in training step. To solve this problem Wysoski et al. [15] use boundary histograms and neural networks to correct the hand posture angle. Grzeszcuk et al. [16] use an approach based on stereo vision to realize in 3D the angle correction.

Keogh et al. [17] propose a simple method to correct the angle variation based on object form, where do not matter the angle distortion the same contour pixel is extracted of the image as the first one and the rest of the contour pixels were always organized in the same order. Keogh et al. achieve good results with this technique in skulls classification, extracting

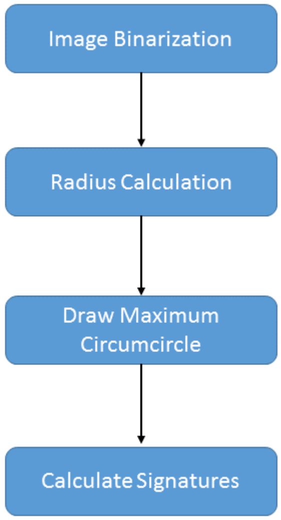

Fig. 1. CIPBR's flow

two signatures of the images, distances and angles, and turning them into a single features vector.

Barros et al. [18] propose a method to solve another problem in computer vision approaches: the feature vector size. The traditional approaches return a big vector which complicate the real-time classification task. The Convexity Approach solve this reducing the hand posture into a polygonal shape and extracting only the more external and internal contour points. For last, it is calculated the distance between each pair of points. Then, a set of points is selected in order to obtain the minimal contour representation and generate a smaller feature vector. Barros et. al. evaluates their work using three pattern classification: Elman Recurrent Neural Network (Elman RNN) [19], HMM and DTW achieving better results with the last two.

Based on Keogh et al. and Barros et al. concepts, CIPBR algorithm is proposed and evaluated in this paper using two classifiers: HMM and DTW. The last one is used in three version, the traditional one, an adaptation to CIPBR's two signatures and a variation presented by Salvador [20].

III. CIPBR Algorithm

The CIPBR algorithm is composed by simple tasks to reduce the hand posture images into two signature sets. There is four modules in cascade.

Figure 1 presents CIPBR's flow starting with the input image and finishing with the feature vector. The first module, "Image Binarization", receives a hand posture image as input and outputs two images: the first one in RGB, same as the input, and the second a binary image (based on Otsu Threshold Algorithm [21]).

The second module, "Radius Calculation", has three simple tasks to complete. The first one is find the hand pulse line in RGB image using a simple linear regression. The second one is extract the contour where each pixel is marked as border, if it has one of its eight neighbors black in the binary image and calculate the mass center point by Hu Invariant moments. This module use same principle as Keogh et al. [17] to guarantees the rotation invariance starting the contour organization with the center pulse line pixel. Lastly the distance between the center of mass of the contour point and pulse center point is calculate as show Figure 2 (a).

The third module, "Draw Maximum Circumcircle", uses the distance previously calculated as radius to draw a circle inside the hand contour. In order to solve the problem of the circle exceeding the hand contour, a triangle is calculated using the three more distant contour points from mass center point and the biggest circle inside it is used.

Then, in "Calculate Signatures" module, convex Hull is calculate from hand posture contour using Andrew's monotone chain convex hull algorithm [22], thus reducing the number of contour points substantially.

As result a set $P = \{p_1, p_2, \ldots, p_n\}$ from Convex hull points are used to generate CIPBR signature sets. For each point $p_k = \rho$ is traced a line PC starting with ρ and ending with the central point C of the maximum circumcicle hand shape exported from Module 3. Then, it is measured the Euclidean distance from ρ to the point Q_k given by the intersection between PC and the maximum circumcicle. The set of all distances computed as this procedure is the descriptors for the first signature set. The distance ρQ_k can be calculate with the follow equation:

$$Distance_{\rho Q_k} = \sqrt{(\rho_x - C_x)^2 + (\rho_y - C_y)^2} - radius,$$

where C is the hand posture contour mass center point, ρ_x, C_x are the x coordinates from points ρ and C respective, ρ_y, C_y are the y coordinates from points ρ and C respective and $radius$ the radius calculated in the second module.

The second signature set consists of angles (A) obtained by calculating the angle (A) between a line composed for each point P of the convex hull hand shape point and radius (PC). All two signature sets are obtained in a clockwise direction always starting with point P as Figure 2 shows.

Finally, to create the feature vector the signature sets are normalized, distances by the radius calculated in the second module as follows:

$$Distance_i = \frac{Distance_i}{radius}$$

and angles set by 360°. The final vector is created concatenating distances and angles in a single vector as follows:

$$Angle_i = \frac{Angle_i}{360°}.$$

IV. Classifier Module based on HMM

For some classification techniques, such as Hidden Markov Model and Artificial Neural Networks, among others require a fixed length feature vector as input, but most of techniques

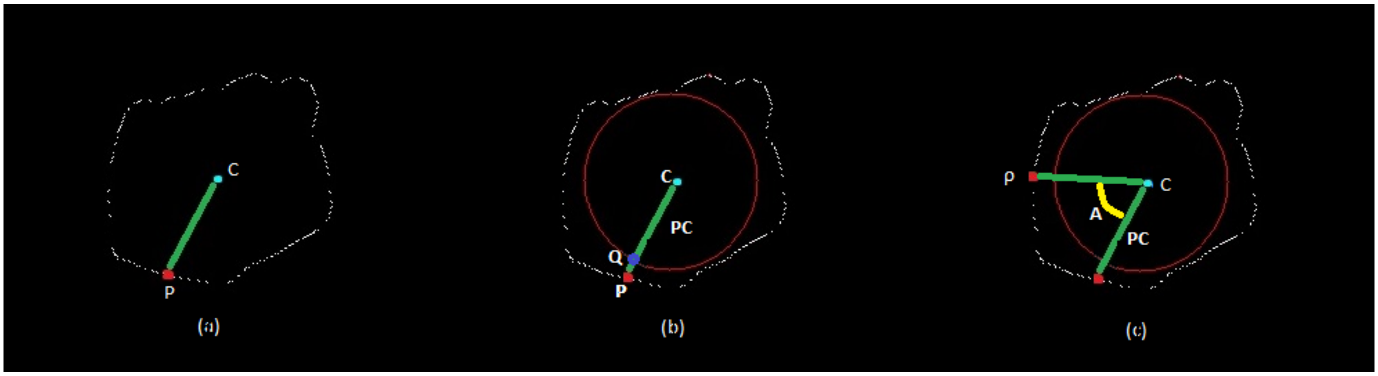

Fig. 2. (a) CIPBR second step output. The hand shape contours and central point (cyan point). The hand base middle point (red point), the radius (green line) and angle A (yellow arc). (b) and (c) CIPBR fourth step signature calculate.

return as output vectors with different sizes. CIPBR algorithm returns a set of vectors, each one containing around 200 features. To use this vectors as input in a HMM, and achieve a convergence state, it is necessary a heavy reduction of features in each vector. To solve this, two approaches is adapted in this work. At first one, Particle Swarm Optimization [23] chose the better feature vectors size and the selector algorithm proposed by Barros et al. [6] receives PSO size output and choose the candidate features for this vectors.

A. Particle Swarm Optimization

When a PSO is used to solve an optimization problem, a swarm of simple computational elements, called particles, is used to explore a solution space to find an optimum solution. The position from each particle represent a candidate solution in dimensional search space (D) represented by $X = \{x_1, x_2, x_3, \ldots, x_n\}$, where each x_n is a position in the n-th dimension. The same way, the velocity is represented by $V = \{v_1, v_2, v_3, \ldots, v_n\}$.

The fitness function evaluates how better each particle present itself in each iteration. Better position where the particle was, is saved in a variable called p_{Best} and has the value exchanged whenever particle finds a better value. 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}. This way, to update the particle velocity and position the following equations are used:

$$\begin{aligned} v_i(t+1) &= \omega * v_i(t) + c_1 r_1 [p_i Best - x_i(t)] \\ &\quad + c_2 r_2 [g_{Best} - x_i(t)] \\ x_i(t+1) &= x_i(t) + v_i(t+1), \end{aligned}$$

where $i = (1, 2, 3, \ldots, N)$ and N is the size of the swarm. c_1 which represent the private experience or "cognitive experience" and c_2 represents the "social experience" interaction usually used with value 2.05. r_1 and r_2 a random number between 0 and 1 which represents how much p_{Best} and g_{Best} will influence the particle movement. The inertia factor ω, is a variable used to control the balance of the search algorithm between exploration and exploitation. x_i represent particle position in $i - th$ dimension. The recursive algorithm will go on until a break condition happen, maximum number of iteration in this paper case.

B. Binary Particle Swarm Optimization

A binary PSO is a variation from traditional version where the particle position imitates the human genetic code. In the binary version, the position particle is represented only by $\{0\}$s and $\{1\}$s. The other change in binary PSO algorithm is how position and velocity are calculated, by the following equation:

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

For binarization process is chosen the threshold which delimit what vector cell will represent a feature. For our experiments is chosen 1.2 as threshold to be the average of all the values presented.

C. Fitness Function

Several fitness function are used to features selection in image recognition systems [24] based on binary PSO approach. After some study a fitness function based on euclidean distance is used in this paper.

First of all, the algorithm receives a binarized vector with n dimensions. Each vector reflect a image gesture and each dimension a feature from it. In this case, the fitness function is calculated with the Euclidean distance from the vector particle position to each vector of the gesture set correspondent and makes the sum of all the distances. In execution, the particles become more able as it fitness value is the smallest possible compared with the fitness obtained by other particles. The particle fitness function is:

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

D. Selector Algorithm

After better feature vector size has been chose for Binary PSO, another normalization is used to choice the features which will compose the vectors. To this task the Barros et al. Selector Algorithm [18] is applied.

First, choose S, the PSO chosen size. Then, if any vector has fewer points than S, are added in the feature vector until matches the desired length (S). The feature vectors with more points than S are redefined using a selection algorithm. This algorithm consists in calculation of a window W through the division of the current vector length by S (desired length). The current vector is parsed, and each value in W position is included to the new feature vector. If the new output vector is even smaller than the desired length, the remaining positions are randomly visited and used to compose the new output vector until the desired length is achieved.

E. Binary PSO Configuration

After some tests, the following values is used to configure Binary PSO in our experiments and find the better feature vector size:

- 15 particles;
- 20 dimensions (*features*);
- 30 simulations;
- 200 iterations;
- $c_1 = c_2 = 2.05$;
- Inertia factor $\omega = 0,9 \rightarrow 0,4$;
- $r_1 = r_2 = rand[0;1]$.

For each gesture the PSO returns a number which better describe each gesture set, but HMM only accepts vectors of the same size. To decide the final size is made a tournament where the number with more incidence wins.

F. HMM Training Configuration

The Hidden Markov Model technique uses a K-Means Clustering [25] to find the best initial approximation. The Baum-Welch algorithm [26] is used to train the HMM, resulting in a fast training process.

V. Classifier Module Based on DTW

The Dynamic Time Warping (DTW) was introduced to overcome the limitation in measure the distance between two time series in specific case: when there is distortion in one of then shifting some slice. To solve this a simple approach based on Euclidean distance is proposed as follows:

Given two time series X, and Y, of lengths $|X|$ and $|Y|$, construct a warp path $W = \{1, w_2, w_3, \ldots, w_k\}$ where $max(|X|, |Y|) \leq K < |X| + |Y|$ and the k^{th} element is $w_k = (i, j)$ where i is an index from time series X and j an index from time series Y.

The warp path must start at $w_1 = (1, 1)$ and finish at $w_k = (i, j)$ in order find the cost matrix

To find the minimum-distance warp path, every cell of the cost matrix must be filled. The value of a cell in the cost matrix is:

$$Dist(i,j) = Dist(i,j) + min[D(i-1,j), D(i,j-1) \\ , D(i-1,j-1)],$$

The warp path to $D(i, j)$ must pass through one of those three grid cells, and since the minimum possible warp path distance is already known for them, all that is needed is to simply add the distance of the current two points to the smallest one. Since this equation determines the value of a cell in the cost matrix by using the values in other cells, the order that they are evaluated in is very important.

A. DTW for CIPBR

The DTW algorithm has a complexity problem of $O(N^2)$ level. This has a direct effect on time rating increasing it exponentially higher order time serie which compromises a real time system. To solve this problem two approaches are used in DTW. The first one in the way that DTW calculates the cost matrix and second on in the presentation series to classifier.

The first change is made in the classifier in the follow way:

$$Dist(i,j) = abs[(D_i - D_j) + (A_i - A_j)] + \\ min[D(i-1,j), D(i,j-1), D(i-1,j-1)].$$

Where D_i, D_j are distances and A_i, A_j angles, from CIPBR feature vectors. All the rest of the DTW algorithm follows the same traditional way.

B. Coarsening

Another way to reduce the time rating in DTW execution is reduce the gesture feature vector size. *Coarsening* [20] is a technique to reduce the size (or resolution) of a time series by averaging adjacent number of points decided by a *radius* which say how many neighbors will be used. The resulting time series is a factor smaller than the original time series. In our experiments the gesture can be reduced several times what reduce the time rate for several times. If a $radius = 2$ is given to reduce a time-series $X = x_1, x_2, \ldots, x_n$, each element of the time-series X' will consist of:

$$x'_n = \frac{x_{i-2} + x_{i-1} + x_i + x_{i+1} + x_{i+2}}{5}.$$

The resulting time series is a factor of two smaller than the original time serie. *Coarsening* is run several times to produce many different resolutions of the time series

VI. Experiments

The effectiveness of the proposed method is shown in comparison with the results presented by Barros et al. [18], using the same dynamic gesture dataset, RPPDI Dynamic Gestures Dataset[1].

[1]Available at http://rppdi.ecomp.poli.br/gesture/database

RPPDI Gesture dataset contains 7 different dynamic gestures each one composed by 14 frames with an uniform background and a size of 640×480 pixels in RGB format. In order to make the hand detection easier, the user wears a black belt to define the pulse in each hand posture. The Figure 3 shows an example of each gesture in the dataset.

In order to evaluate our approach, a classification system composed of two modules connected in cascade is used. The first module contains the CIPRB algorithm to extract features from the hand gesture images, while the second module classify the gesture using HMM or DTW.

To evaluate the proposed system as presented by Figure 4, four scenarios are executed. For all scenarios a system based on computer vision is tested with CIPBR algorithm as features extractor module. The scenarios are presented as follows:

1) The first one scenario consists in use Classifier Module based on HMM as system classifier, normalizing the feature vectors with PSO and Barros et al Selector Algorithm;
2) The second scenario consists in traditional DTW as classifier;
3) Third scenario uses the modified DTW for CIPBR (accord section 5.1);
4) The last scenario reduce feature vectors using *coarsening* method with $radius = 4$ and uses the traditional DTW to classify.

All scenarios are repeated 30 times in a dataset randomly chosen containing 2/3 of the sequences in each gesture class for training and 1/3 for test. The results present the average among all repetitions.

A. Results and discussion

Table I presents the resume results in classification for this work. The table shows that all classifiers have achieved rates nearing complete resolution of the RPPDI dataset.

HMM recognize the lower number of gesture, achieving only 94% of classification rate, making more mistakes than all DTW versions. But the average time rated is several times smaller than the faster version of DTW. Due to the low number of features generated by the combination of PSO with the selector algorithm, generating vectors with size 10.

TABLE I
CLASSIFICATION RATE RESUME FOR ALL EXPERIMENTS.

Scenario	Classification rate	Standard Deviation	Time Average
1.	94.11%	6.72%	1.78ms
2.	99.79%	0.54%	4516.74 ms
3.	98.85%	1.41%	16337.94ms
4.	96.82%	1.85%	956.74ms

Table II presents a comparison between the classification rate obtained by Barros et al. [6] using the Convexity Approach applied to Local Contour Sequence (LCS) and Speed Up Robust Features (SURF). The combination of the techniques LSC and SURF with Convexity Approach are called CLCS and CSURF, respectively.

TABLE II
CLASSIFICATION RATE RESUME COMPARING BARROS ET AL. [18] RESULTS WITH OURS.

Method	Classification rate (%)
CLCS + HMM	91.00
CSURF + HMM	91.00
CIPBR + HMM	94.11
CLCS + RNN	90.00
CSURF + RNN	92.00
CSURF + DTW	93.00
CLCS + DTW	97.00
CIPBR + Coaserning + DTW	96.82
CIPBR + DTW for CIPBR	98.85
CIPBR + DTW	99.79

CIPBR and DTW combination increases the classification rate in 2.79 percentage points in the better Barros' result. For several executions the proposed system do not commit any classification mistake, solving the gesture recognition problem in the RPPDI database.

The results comparative have a smaller difference among them, been necessary apply a Shapiro-Wilk statistic test between the result using the CLCS and CIPBR for HMM and DTW as classifiers. The test with HMM presents a p-value equals to 3.89336E-26 what discard the possibility of equality between scenarios for 95% significance. To DTW versions the p-values are smaller than 0.5 for scenario 2 and 3. For scenario 4, p-value has 0.508361 as value demonstrating the equality among the CLCS and CIPBR scenarios. This demonstrates the effectiveness of CIPBR approach achieving better results than CLCS in RPPDI dataset, where in the worse case our proposed system has an effectiveness equal an system based on CA.

VII. CONCLUSION

This paper presents a new approach for feature extraction and classification of hand gestures called CIPBR. CIPBR + DTW combination improve the previously results for RPPDI dataset.

Also is presented a new method to use the binary version of PSO to find the optimal number of features for CIPBR vector being selected by a simple selector algorithm and classified using the Hidden Markov Model as classifier. This selector algorithm is necessary because PSO fitness function generated inconclusive results for classification, being necessary another method to choice the feature to final vectors.

The results showed promise in that first moment of finding the best number necessary to describe every gesture extracted by CIPBR. Using dynamic gestures RPPDI dataset, our approach achieved 100% of classification rate in almost all executions solving the classification problem of RPPPDI dataset on both assertiveness as at runtime with both classifiers. As can be seen in the results session, DTW is much more assertive than HMM but the last got a lot shorter rating

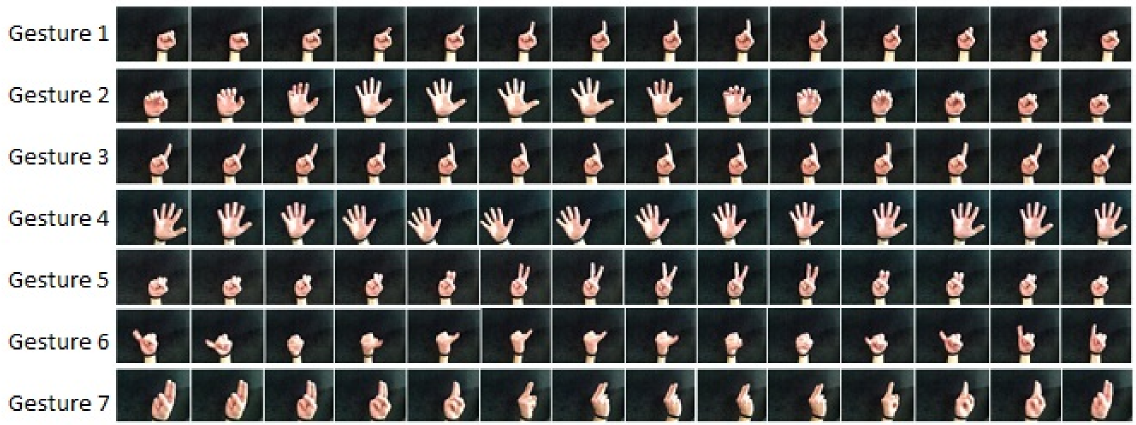

Fig. 3. RPPDI Dynamic Gesture Dataset [18]

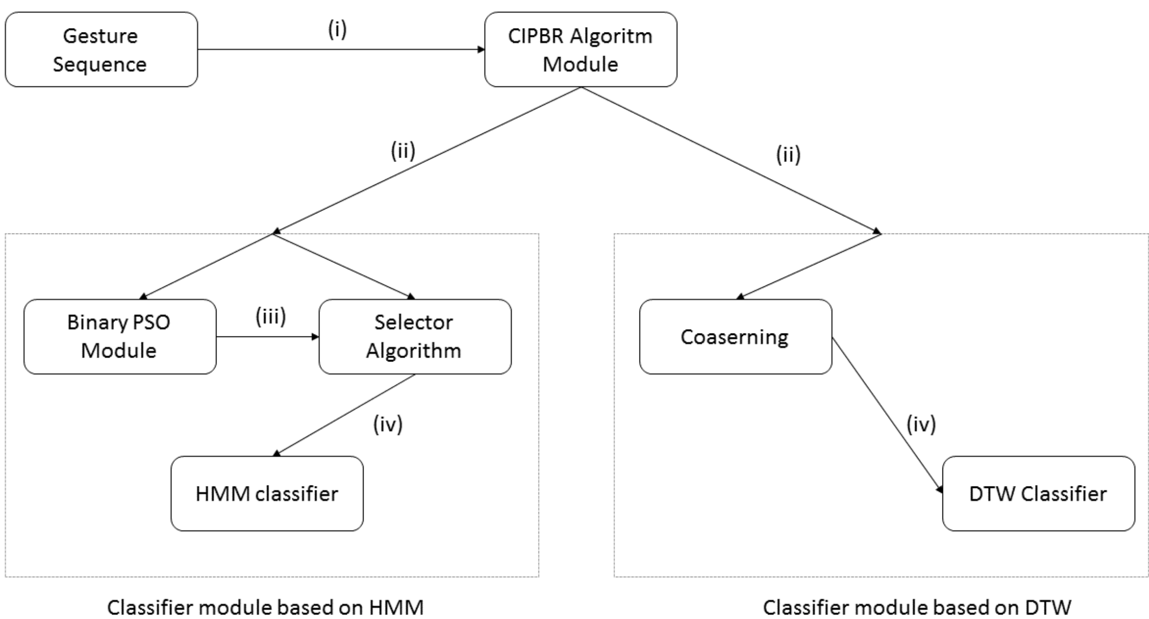

Fig. 4. General architecture of the system. (i) Gesture images sequence as input for CIPBR module. (ii) Gesture features vector. (iii) The best size for gesture features vector. (iv) Reduced vector for respective classifier.

time, being the best for a system to classify gestures in real time.

As future work, a priori is find a fitness function that makes assertive binary PSO both in reducing characteristics as for classification. In consequence, the use of more complex dataset and using other cameras also, such as the Microsoft Kinect.

Acknowledgment

The author would like to thank the support of the Brazilian agencies: CNPQ, Facepe and Capes.

References

[1] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, 2011.

[2] G. Lu, L.-K. Shark, G. Hall, and U. Zeshan, "Immersive manipulation of virtual objects through glove-based hand gesture interaction," *Virtual Reality*, vol. 16, no. 3, pp. 243–252, 2012.

[3] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[4] G. Dewaele, F. Devernay, and R. Horaud, "Hand motion from 3d point trajectories and a smooth surface model," in *Computer Vision-ECCV*

2004. Springer, 2004, pp. 495–507.

[5] J. Schlecht, B. Carqué, and B. Ommer, "Detecting gestures in medieval images," in *2011 18th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2011, pp. 1285–1288.

[6] P. V. A. Barros, N. T. M. Júnior, J. M. M. Bisneto, B. J. T. Fernandes, B. L. D. Bezerra, and S. M. 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.

[7] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[8] H. Sakoe, S. Chiba, A. Waibel, and K. F. Lee, "Dynamic programming algorithm optimization for spoken word recognition," *Readings in speech recognition*, vol. 159, 1990.

[9] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Readings in speech recognition*, A. Waibel and K.-F. Lee, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, pp. 267–296. [Online]. Available: http://dl.acm.org/citation.cfm?id=108235.108253

[10] S. Meena, "A study on hand gesture recognition technique," Master's thesis, National Institute of Technology, Rourkela, 2011.

[11] A. El-Sawah, C. Joslin, N. D. Georganas, and E. M. Petriu, "A framework for 3D hand tracking and gesture recognition using elements of genetic programming," in *Fourth Canadian Conference on Computer and Robot Vision, 2007. CRV'07*. IEEE, 2007, pp. 495–502.

[12] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 105–112.

[13] J. Bao, A. Song, Y. Guo, and H. Tang, "Dynamic hand gesture recognition based on SURF tracking," in *2011 International Conference on Electric Information and Control Engineering (ICEICE)*. IEEE, 2011, pp. 338–341.

[14] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[15] S. G. Wysoski, M. V. Lamar, S. Kuroyanagi, and A. Iwata, "A rotation invariant approach on static-gesture recognition using boundary histograms and neural networks," in *Proceedings of the 9th International Conference on Neural Information Processing, ICONIP'02*, vol. 4. IEEE, 2002, pp. 2137–2141.

[16] R. Grzeszczuk, G. Bradski, M. Chu, and J.-Y. Bouguet, "Stereo based gesture recognition invariant to 3D pose and lighting," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 826–833 vol.1.

[17] 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.

[18] P. V. A. Barros, N. Junior, J. M. M. Bisneto, B. J. T. Fernandes, B. L. D. Bezerra, and S. M. 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.

[19] L. Medsker and L. C. Jain, *Recurrent neural networks: design and applications*. CRC press, 1999.

[20] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

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

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

[23] J. Kennedy, R. Eberhart *et al.*, "Particle swarm optimization," in *Proceedings of IEEE international conference on neural networks*, vol. 4, no. 2. Perth, Australia, 1995, pp. 1942–1948.

[24] X. Zhang, W. Wang, Y. Li, and L. Jiao, "PSO-based automatic relevance determination and feature selection system for hyperspectral image classification," *Electronics Letters*, vol. 48, no. 20, pp. 1263–1265, September 2012.

[25] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[26] T. L. Baum, G. Peterie, and N. W. Souled, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," in *Proceedings of Annals of Mathematical Statistics*, vol. 41, 1995, pp. 164 –171.