Computer Vision and Image Understanding 155 (2017) 139–149

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

A dynamic gesture recognition and prediction system using the convexity approach

Pablo Barros[a,*], Nestor T. Maciel-Junior[b], Bruno J.T. Fernandes[b], Byron L.D. Bezerra[b], Sergio M.M. Fernandes[b]

[a] Department Informatics, University of Hamburg, Knowledge Technology, Hamburg, Germany
[b] Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife-PE, Brazil

ARTICLE INFO

Article history:
Received 14 September 2015
Revised 14 September 2016
Accepted 17 October 2016
Available online 21 October 2016

Keywords:
Gesture recognition
Computer vision
Features extraction
Gesture prediction

ABSTRACT

Several researchers around the world have studied gesture recognition, but most of the recent techniques fall in the curse of dimensionality and are not useful in real time environment. This study proposes a system for dynamic gesture recognition and prediction using an innovative feature extraction technique, called the Convexity Approach. The proposed method generates a smaller feature vector to describe the hand shape with a minimal amount of data. For dynamic gesture recognition and prediction, the system implements two independent modules based on Hidden Markov Models and Dynamic Time Warping. Two experiments, one for gesture recognition and another for prediction, are executed in two different datasets, the RPPDI Dynamic Gestures Dataset and the Cambridge Hand Data, and the results are showed and discussed.

1. Introduction

Gesture recognition systems can be used as a natural and welcoming interface of interaction between computational systems and humans. These systems use human movement patterns to identify, learn, and generalize gestures executed by a user. There are several applications in the area of gesture recognition, such as games (Lee and Hong, 2010; Rautaray and Agrawal, 2011), human-robot interaction (Bodiroza et al., 2013; Lee, 2006), interaction with televisions (Jeong et al., 2012), and sign language recognition (Ciaramello and Hemami, 2011; Liu and Xiao, 2015; Silanon and Suvonvorn, 2014; Zhou et al., 2008).

Nowadays, it is possible to capture gestures executed by humans using only a video camera on a smartphone, tablet, or notebook. Since most people have at least one of these devices in their possession, using gestures for communication with computational systems could be employed more often than in the previous years. The evolution of computers, in hardware and software, also increases the usage of gestures as an important tool for human-computer communication.

Gesture recognition systems can be clustered into three different categories (Ibraheem and Khan, 2012): systems based on gloves or external sensors attached to a user for gesture capture (Cheng et al., 2015; Dekate et al., 2014; Han, 2010; Huang et al., 2011; Jeong et al., 2011; Xinyu et al., 2010), systems that recognize a gesture through a tracking device, such as a mouse (Bhattacharjee et al., 2015; Chivers and Rodgers, 2011; Cho et al., 2004; Jeong et al., 2012; Schlecht et al., 2011), and systems that capture gestures using a video camera and process them with computer vision techniques (Bernardes et al., 2009; Koceski and Koceska, 2010; Leubner et al., 2001; Sen et al., 2005; Wu et al., 2015; Zhang and Zhang, 2008). The first category captures the gesture more precisely, but the process is invasive, as the user wears the sensors around him. The application of gloves for capturing happens in controlled environments, where the gloves are connected to computers, but the capturing process becomes complicated to be used in an external environment and real world applications.

The second category of gesture recognition systems uses a tracking technique to follow a hardware device in the screen. The tracked path is the gesture to be recognized. This kind of recognition uses a simple gesture definition, reducing the computational cost. However, the possible gestures represented are less significant and less precise than the other categories.

The last category uses a video camera to capture and identify the gesture. The recognition process uses the images to extract some features, such as movement, position, velocity, color,

* Corresponding author.
E-mail addresses: barros@informatik.uni-hamburg.de (P. Barros), ntmj@ecomp.poli.br (N.T. Maciel-Junior), bjtf@ecomp.poli.br (B.J.T. Fernandes), byronleite@ecomp.poli.br (B.L.D. Bezerra), smmf@ecomp.poli.br (S.M.M. Fernandes).

http://dx.doi.org/10.1016/j.cviu.2016.10.006
1077-3142/

among others. This approach can describe a more complex gesture than the second category, and it does not need to be used in a controlled environment. This category uses computer vision techniques that can be embedded in a smart-phone, for example, and used for gesture recognition in any place.

One of the main problems with dynamic gesture classification is to deal with sequence recognition. Gestures are naturally dynamic, as a sequence of movements and postures, and each step has influence in further ones. These works deal usually with temporal dependency (Alon et al., 2009; Chang, 2016; Santos et al., 2015), and usually demand very long time and computational effort for training. Besides that, a robust feature extractor is necessary, which increases the computational cost of the process (Frolova et al., 2013; Wu et al., 2016), and thus decreases the possibility of the model to work in real time.

A desirable characteristic in a gesture recognition system is the ability to recognize gestures in real time. As pointed out by Mori et al. (2006), a real time gesture recognition system has to be able to predict the gesture that is being executed before it ends. Prediction allows the recognizer to work in real time and makes the classifier more accurate, as it can use the prediction results to improve recognition. The prediction attempts to identify a pattern that has yet to be completed. Some prediction techniques have been used with success to improve speech recognition (Helander and Nurminen, 2007; Husssain et al., 2009; Javed and Ahmad, 2014; Satya et al., 2011; Stavrakoudis and Theocharis, 2007). A few studies use the prediction concept gesture prediction, as described by Ahmad et al. (2015); Kohlsdorf et al. (2011); Liu and Xiao (2015); Silanon and Suvonvorn (2014).

Although there are many studies using computer vision for gesture recognition, some problems remain, like significant computational and time costs for the algorithms. To use gesture recognition techniques in a real time environment, it is necessary to reduce their computational and time costs. Such reduction can be achieved using a smaller feature vector to describe a gesture or a prediction technique (Hasan and Kareem, 2012). Our study shows a dynamic gesture recognition system that uses an innovative technique, called the Convexity Approach, for feature extraction. The system is evaluated for dynamic gesture recognition and for gesture prediction. It implements classification and prediction modules based on Hidden Markov Models and Dynamic Time Warping. The proposed method is able to classify dynamic hand gestures by recognizing individual hand postures and modeling a sequence with them. This way, our approach can classify gestures with different speed and different users. Our approach is strongly based on an innovative feature extraction technique, and it uses the capability of dynamic time warpers to model gesture sequences.

This paper is organized as follows. Section 2 describes the Convexity Approach technique. Section 3 shows the recognition and prediction system architecture. Section 4 presents the experimental setup and results, as well as a discussion about the obtained results. Finally, in Section 5, we present the concluding remarks.

2. Convexity Approach

Several techniques are used to describe gestures. Most of these feature extraction techniques are affected by the curse of dimensionality (Bilal et al., 2011). The curse of dimensionality states that an approximation of a numerical function will have a higher computational cost if its variables increase (Kouiroukidis and Evangelidis, 2011). There are proposed solutions for the curse of dimensionality, such as the reduction of the feature vector dimension (Pagel et al., 2000; Zhao et al., 2010), classification algorithm optimization (Qaiyumi and Mirikitani, 2006; Qin and Tang, 2009), and the use of feature selection strategies for the problem (Teoh and Sheble, 2007).

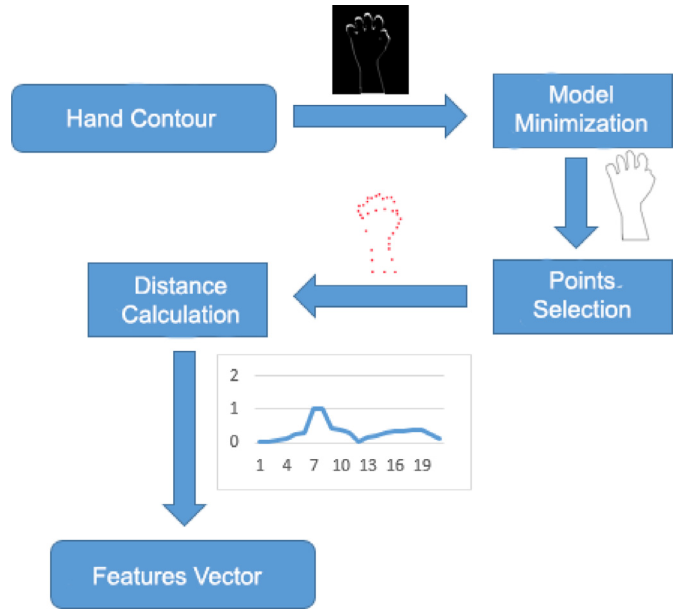


Fig. 1. Illustration of Convexity Approach execution.

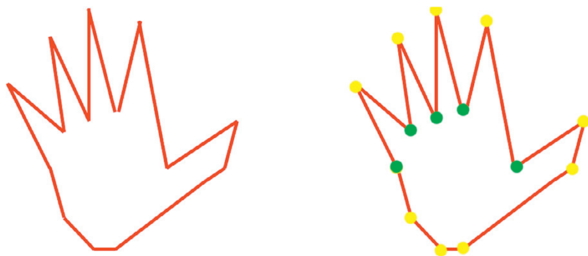
The Convexity Approach, the feature extraction technique, can describe a hand gesture using only dynamically selected points in the hand contour. The selection of points is minimized for each hand posture, so the feature vector contains the minimal features necessary to describe the hand.

The Convexity Approach extracts features of one image at a time. This image must contain only the hand contour. The first step of the algorithm is to reduce the geometrical model of the hand, eliminating curves. The second step is to find a minimal set of points that can represent the minimized hand model. The last step is to extract the distance of these points and create a feature vector that will describe the hand. Fig. 1 shows the illustration of the Convexity Approach execution.

2.1. Model minimization

The first step of the Convexity Approach assures that any extra information will not be extracted. In the hand gesture context, extra information is a part of the hand that can be excluded without losing the shape of the geometrical model of the hand. For example, a curve in the hand can be represented by three points. The Douglas-Peucker (Douglas and Peucker, 1973) algorithm is used to create the minimized hand model and it is successfully applied for traffic sign recognition (Soendoro and Supriana, 2011), model minimization using compression (Nandakumar et al., 2005), and geographical applications (Youfu and Guoan, 2010).

An ordered set of $n+1$ points in a plane forms a polygonal chain. Given the chain C with n segments, the Douglas-Peucker algorithm will generate a chain C' with fewer segments than C. The two endpoints of a set of points are connected by a straight line $\overline{AB}$ as the first rough approximation of the polygon. Iterating over all the vertexes, v, of all segments n in C, the distance between the vertex v_i and the center of $\overline{AB}$ is calculated. If the distance of all vertexes is shorter than a threshold t, then the approximation is good, the endpoints are retained, and the segment $\overline{AB}$ will be added to C' and represent the polygon. However, if any of these distances exceeds t, the approximation is not good. In this case, it chooses the furthest point P, and subdivides the initial set points into two new segments $\overline{AP}$ and $\overline{PB}$. The same procedure is repeated recursively on these two new segments, and the new segments are



(a) Polygon Minimization (b) Convex Hull and inner points selection

Fig. 2. Outputs of step (a) one and (b) two of Convexity Approach.

added to C'. The routine continues until all possible points have been eliminated. Fig. 2(a) shows the output of this step.

2.2. Selecting the points

The second step of the Convexity Approach is the selection of a minimal set of points that can represent the hand. This selection is performed in two parts: the first part consists of selecting the external polygon edges that remain in the model; and the second part consists of selecting the internal edges that detail the model, based on the previous selection. The first part uses a convex hull around the model to identify the selected edges in the model. The edges that touch the convex hull, are chosen for the second step. The Sklansky (Sklansky, 1982) algorithm, later corrected by Melkman (1987), is used to create a convex hull from the model. The Sklansky Algorithm is implemented as follows:

- The convex vertex of the polygon is found.
- The remaining n-1 vertexes are named in clockwise order starting at P0.
- Select P0, P1, and P2 vertices and call then “Back,” “Center,” and “Front,” respectively
- Execute the follow algorithm:
  –

while “Front” is not on vertex P0 and “Back,” “Center,” and “Front” form a right turn **do**
  if “Back,” “Center,” and “Front” form a left turn or are collinear vertex **then**
    change “Back” to the vertex ahead of “Front”. Relabel “Back” to “Front,” “Front” to “Center,” and “Center” to “Back”.
  else if “Back,” “Center,” and “Front” turn left **then**
    change “Center” to the vertex behind “Back”, remove the vertex and associated edges that “Center” was on and relabel “Center” to “Back” and “Back” to “Center”
  end if
end while

For the second part a new algorithm was developed using the previously selected external points to find the internal points of the contour. The external points represent the general shape of the hand and the internal points are responsible for distinguishing the small changes in the shape of the hand. For each pair of external points, a line segment that crosses these two points is created, creating the segment $\overline{AB}$. The points in the original polygon located between the vertical or horizontal coordinates of the points A and B are chosen as internal point candidates. A distance t is calculated between each candidate and the center of $\overline{AB}$. The point that has the greatest distance is selected and marked as an internal point, and the others are removed. This operation can be described as follows:

$$internalPoint = max(t_i), \tag{1}$$

where i iterates over all the points in the subset. In case that there is no point or the distance is 0, no point is considered. After the internal and external points are selected, the minimized hand model is generated and is ready for the last step, feature extraction. Fig. 2(b) shows the output of this step.

2.3. Distance calculation

The last step of the Convexity Approach is feature extraction based on distance calculation. A line segment $\overline{AB}$ is crossed between each pair of the external points, in a clockwise order. This process makes the Convexity Approach robust to rotation invariance. The distance between this line and the closer inner point is calculated and added to the output vector. The distances are normalized, dividing by the maximum value obtained, to make the distances robust for scale changes. This operation is described as

$$\begin{aligned} &distances = distance(\overline{AB}, internalPoint_{AB}), \\ &NormalizedDistances = distances/max(distances). \end{aligned} \tag{2}$$

2.4. Convexity Approach remarks

Since the Convexity Approach uses a hand contour as input, it can be used with other feature extraction techniques. The Local Contour Sequence (LCS) technique (Gupta and Ma, 2001) uses a set of imaging processing techniques to find the hand contour that can be used as input for the Convexity Approach. The use of LCS with Convexity Approach is called CLCS (Barros et al., 2013b). The Speed Up Robust Features (SURF) (Bay et al., 2008) is a technique that nds and describes interest points in an image. When using these interest points extracted by SURF as input for the Convexity Approach, we have a technique called CSURF. Fig. 3 shows the result obtained by LCS and SURF and the application of the Convexity Approach, respectively CLCS and CSURF.

For some classification techniques, such as neural networks (Chen et al., 2007), the feature vector must be normalized. To solve this problem, we propose a method. First, the number of normalized distances is defined. Then, if the image has fewer points than the determined one, “0” is added in the feature vector, until it matches the desired length. The outputs with more points than the desired length are normalized using a selection algorithm. This algorithm consists of calculating a window W through the division of the output length for the desired length. The output vector is traversed, and each W position is added to the new vector of outputs. If the new output vector is 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. This process does not change the normalized vector; it only collects some samples that already belong to the pool of calculated values. This operation keeps the same data structure of the values that are not normalized. Therefore, it does not result in a drastic change in the feature distribution. Fig. 4 illustrates this process.

2.5. Convexity Approach algorithm

The Convexity Approach algorithm, described below, has a complexity of O (N^2). The algorithm starts with the minimization of the model, as shown in line 1. After this, the search for the most significant points is performed in line 2. The search separates the points into external, the ones that define the model, and internal, the ones that specialize the model. The characteristic vector is composed of the distance between the external points and internal points, as shown in lines 3–7. The normalization of the feature vector is performed in line 8.

Fig. 3. (a) Applying the Convexity Approach on LCS (b) produces a new output vector. (c) Applying the Convexity Approach on SURF (d) produces a new output vector.

Initial Array

W=10 / 4 = 2,5.

Position:	1	2	3	4	5	6	7	8	9	10
Value:	0.3	0.4	0.2	0.1	0.8	0.9	0.7	0.6	0.5	1
	W_1			W_2			W_3			W_4

Normalized Array

Position:	1	2	3	4
Value:	0.3	0.1	0.7	1

Fig. 4. Example of the normalization process. In this example the original vector has 10 elements and the normalized one has 4. The window W is calculated and each W position is visited.

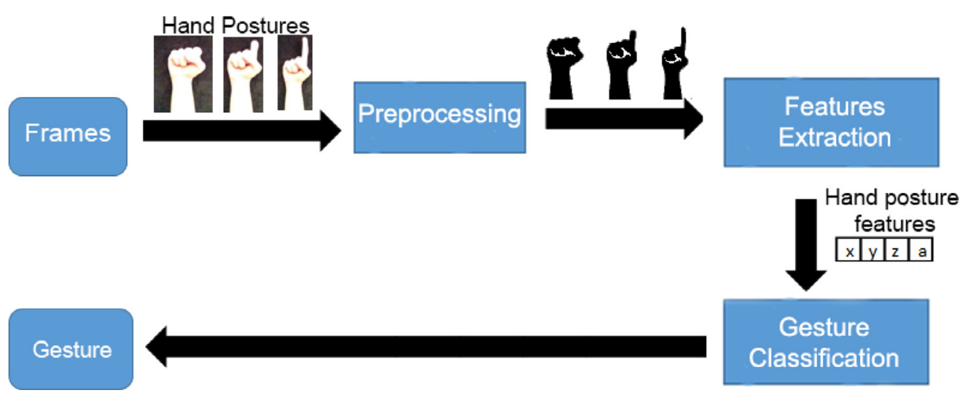


Fig. 5. Proposed gesture recognition system.

Algorithm 1 Convexity Approach algorithm.

```
List points← minimizeModel(image)
points← searchSignificantPoints(points)
List distances
for externalPoints em points do
    if has internalPoint then distances.add(externalPoint,
internalPoint)
    end if
end for
distances = normalization(distances,normalizationLenght)
return distances
```

3. Gesture recognition system architecture

Gesture recognition systems usually are separated into three components: image preprocessing, features extraction, and pattern recognition. Computer vision based systems use data obtained from video cameras as input, which is preprocessed, eliminating noise and highlighting interest regions. This data is used as input to a feature extraction algorithm that generates domain-based feature vectors to be employed in the classification module. A classification module receives these features and estimates to which class they belong. The recognition system used in this study is not different from the previous description. Fig. 5 presents the proposed architecture.

The system is designed so that each part could be implemented with different techniques and the feature extraction algorithm used outputs the hand contour for the Convexity Approach. When using the CLCS, it implements the LCS hand contour algorithms. When using the CSURF, it implements the SURF algorithm to find the interest points. Each frame extracted from the video camera is individually preprocessed and used as input to the feature extraction technique. Each frame has its features extracted independently as well. Finally, the gesture is classified based on its frames by the classification module and the system can recognize dynamic gestures.

3.1. Prediction model

A prediction system can classify an incomplete pattern. Such a system is suitable for real world applications, because a gesture can be recognized before the full input sequence has been captured. It uses a partially captured pattern and classifies it as one of the previously learned patterns. This study uses gesture prediction architecture based on an incomplete pattern capture. For each captured frame, a feature vector is extracted using the Feature Extraction module. Each new feature vector is added into a buffer to be

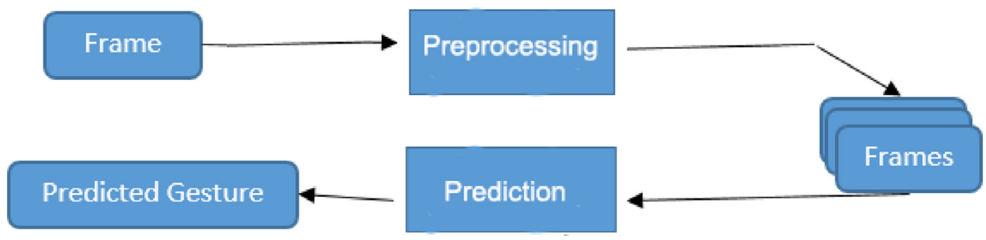


Fig. 6. Gesture prediction system general architecture.

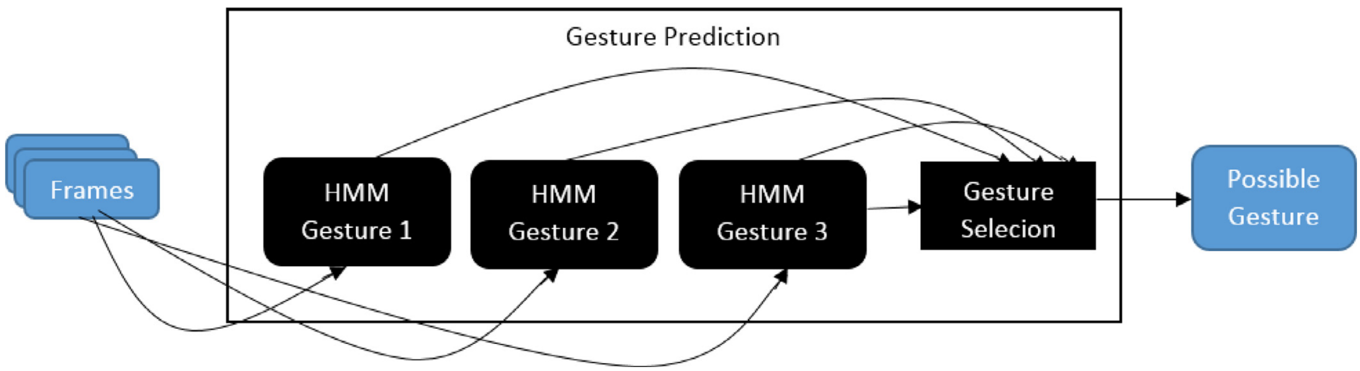


Fig. 7. HMM prediction system architecture.

used as system input, and is submitted to the Gesture Prediction module. The gesture is predicted for the partial input, and each new feature vector added produces a new prediction. Fig. 6 shows the general architecture of the system. To evaluate the architecture, two techniques are used for the gesture prediction task. The first uses a Hidden Markov Model (HMM) (Rabiner, 1990) to learn the full gestures and recognize the partial ones. The second prediction system uses a Dynamic Time Warping (DTW) (Sakoe and Chiba, 1990) method to calculate the distances between the gestures and find the predicted one.

3.2. HMM Prediction system

The HMM Prediction System uses one HMM to describe each gesture. Each HMM is composed by three states, which proved to be sufficient for the prediction task. The system uses a K-means Clustering (Hartigan and Wong, 1979) to find the best initial approximation that showed an improvement in the nal prediction rate in previous experiments. The Baum-Welch algorithm (L. Baum et al., 1995) is used to train the HMM resulting in a fast training process. As shown in Fig. 7, each new system input undergoes all of the HMMs and the output probability is calculated. This probability indicates how close the input is to each HMM model. These probabilities are sent to Gesture Selection, and the model with the highest probability is selected as the predicted gesture.

3.3. DTW Prediction system

DTW is a technique that compares two distances that can be different in time and space, and thus can be used to compare two dynamic gestures. The DTW Prediction System uses a set of examples for each gesture to compose the full gesture representation. The distances between each input and the set of samples of each gesture are calculated. The average distance of all the sample distances is chosen as the distance of the input and the gesture. The gesture with the smallest average distance is selected as the predicted gesture. Fig. 8 shows this system architecture. The Simple DTW implementation (Sakoe and Chiba, 1990) was applied, and it presented good results in the prediction rate. This implementation uses a Euclidean distance calculation to find the smallest distance between two sequences, thus the computational costs increases drastically as the input vector size increases.

4. Experimental results

4.1. Experimental setup

4.1.1. Cambridge hand gesture data set

We execute several experiments, in two different datasets. The first set of experiments is performed on the Cambridge Hand Gesture Data set (Kim et al., 2007). This dataset is composed of 900 image sequences separated into nine hand gestures classes. Fig. 9 shows an example of gestures in this dataset. The dataset is divided in five different types of illumination, and contain ten sequences executed by two separated subjects, in a total of 100 sequences per class. Each sequence has a different number of images.

To be able to segment the Cambridge dataset, a different segmentation technique was used. Each image is preprocessed by convolving it with a Difference of Gaussian (DoG) filter. DoG filters are effectively similar to ZCA whitening (Bell and Sejnowski, 1997), without the need to learn the filter kernels first. The shape of DoG filters is a good approximation of ideal decorrelation filters for grayscale images (Brown et al., 2011), and thus smooth the illumination difference between the sets and allow the images to be pre-processed similarly to each other. After the convolution, a Laplace-Beltrami operator (Reuter et al., 2009) is applied. This operator highlights rapid intensity changes, working as a capable edge detector. Fig. 10 illustrates an example of a segmented sequence.

In these sets of experiments we used all sequences of one illumination for training, and the others for testing. We repeated the experiment for each illumination set. For this analysis, we only used the CLCS technique in the feature extraction system and the DTW in the classification and prediction systems. For the prediction experiments, we separated the number of frames in percentage. So, we evaluate the model using 10%, 25%, 75% and finally 100% of the frames.

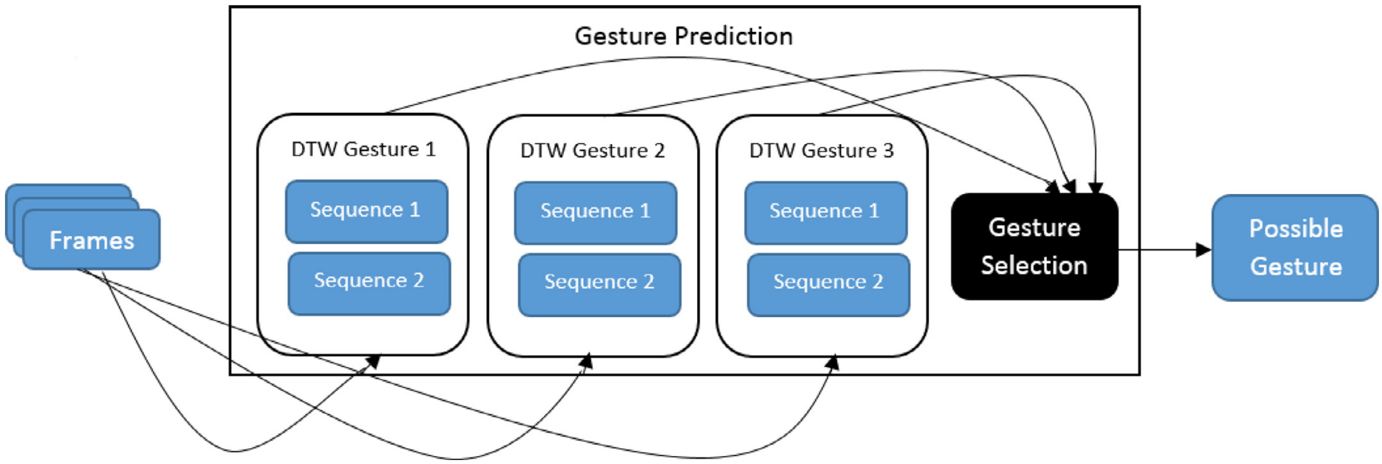


Fig. 8. DTW prediction system architecture.

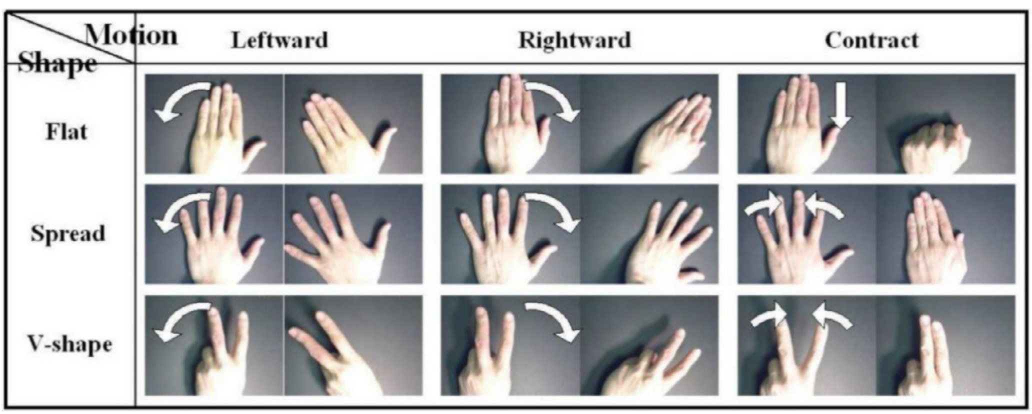


Fig. 9. Cambridge hand gesture dataset [9].

Fig. 10. Example of preprocessing applied to each image in the Cambridge dataset: First a convolution with a DoG filter (Brown et al., 2011) is applied, and a Laplace-Beltrami (Reuter et al., 2009) operator is applied to highlight the edges of the hand.

Table 1
Parameters for the experiments.

Technique	Parameters	CLCS	CSURF	LCS	SURF
HMM	States	3	3	3	3
	Baum-Welch iterations	100	100	10	10
	Features	10	10	5600	1400
DTW	Features	140	210	5600	1400

4.1.2. RPPDI dynamic gestures dataset

In our second sets of experiments the RPPDI Dynamic Gestures Dataset (Barros et al., 2013a)[1] is used for the evaluation. This dataset contains seven dynamic gestures with several examples per gesture. Each gesture is composed of 14 frames. Fig. 11 shows the dynamic hand gesture of this dataset.

Several tests are executed to evaluate the implemented gesture recognition architecture and the Convexity Approach technique. These tests are performed with 12 different configurations: using four different techniques in the feature extraction module and two in the classification module. CLCS, CSURF, LCS, and SURF are implemented for feature extraction, but we only present the results for the first three methods because the recognition rates obtained with SURF are too low and not conclusive enough to be shown in this paper. DTW and HMM were implemented for classification.

Each set of 14 frames is used for training and recognition. In the HMM, each gesture is represented by a different HMM. To classify a new gesture, all of the 14 feature vectors of one gesture are presented to each HMM. The output that generates the biggest probability is the chosen one. For the DTW, each set of frames that represents one gesture is separated into groups. Each gesture to be classied is compared to each group, and the one that shows the lowest distance is chosen.

In the experiments, the dynamic gesture dataset is randomly divided into 2/3 for training and 1/3 for testing. This procedure is repeated 30 times and the average recognition rate of these executions is shown. The parameters for the experiments are presented in Table 1. A series of tests was executed and the parameters were chosen based on the results of these tests.

For gesture prediction, three feature extraction techniques are used, LCS, CLCS and CSURF, along with the two proposed prediction architectures using DTW and HMM. The prediction process occurs when an incomplete gesture is classified correctly. To be able to do that, the classification techniques are trained using the full definition of the gesture. To achieve the prediction, a feature vector containing fewer frames than the complete gesture is classified. These experiments were executed with the same configu-

[1] Available in http://rppdi.ecomp.poli.br/gesture/database/, together with our implementation of the Convexity Approach.

Fig. 11. RPPDI dynamic gestures dataset.

Table 2
Comparison between different techniques applied to the Cambridge Hand Gesture Dataset.

Techniques	Class. rate(%)
GPF (Liu and Shao, 2013)	85.00
HDN (Kovashka and Grauman, 2010)	85.60
AFMKL (Wu et al., 2011)	87.27
COV3D (Harandi et al., 2013)	93.91
DTW+CSURF	93.98 (+/- 2.1)

rations of the previous one. The difference is that 14 experiments are performed, starting with one frame and then adding one more frame to the feature vector, until the full gesture with 14 frames is achieved.

4.2. Results on the cambridge hand gesture data

4.2.1. Gesture classification

We compare our results with different approaches: Hierarchy of discriminative space-time neighborhood features (HDN) (Kovashka and Grauman, 2010), Augmented features in conjunction with multiple kernel learning (AFMKL) (Wu et al., 2011), Spatial-temporal covariance descriptors (Cov3D) (Harandi et al., 2013), and genetic programming generated features(GPF) (Liu and Shao, 2013). The HDN approach learns the shape of spatial-temporal features, based on the neighbors of each hand posture. The AFMKL method learns pixel intensity change distribution and classifies the gesture based on it. The Cov3D technique creates spatial-temporal covariance video descriptors, based on the integral video. The GPF approach uses genetic programming to generate a gestures feature set which is classified using common machine learning techniques. Table 2 shows the results for each technique. The standard deviations of the other methods are not reported. Our method achieves a result comparable with the Cov3D descriptor, and superior to the others. Our technique can describe the features based on each posture. In contrast to other methods, our approach first detects and extracts the hand shape, and composes the nal gesture with the transition between the shapes. Our model is more robust to the different hand gestures, and not only to pixel intensity change in a video. This capability allows our model to be used in tasks where the hand posture is necessary, such as sign language modeling, which would not be possible using any of the other methods.

Table 3
Prediction results using CLCS and DTW with 15%, 25%, 50%, 75% and 100% of each gesture sequence.

Percentage of frames	15	25	50	75	100
Prediction rate(%)	10.5	37.8	50.2	67.1	93.98
Standard deviation	3.2	7.5	3.1	2.7	2.1

Table 4
Results for the dynamic gesture recognition classification using HMM in the classification module.

Techniques	Class. Rate(%)	Standard deviation.	Class. time(ms)
LCS+HMM	52.55	5.70	3.07
CLCS+HMM	81.66	5.41	0.79
CSURF+HMM	77.44	4.81	1.19

4.2.2. Gesture prediction

We applied the CLCS and DTW techniques in a prediction task using the Cambridge dataset. Table 3 exhibits the results. It is possible to see that there is an increase in the prediction rate, with the increase of the number of frames in the sequence, which is expected. However, when less than half of the frames are used for the prediction, the model could not achieve prediction rates above 40%. That shows that the Cambridge set has a lot of variation in the shape of the hand, for the first frames, and only in the later stage of the gesture execution, the gesture is differentiated.

4.3. Results on the RPPDI dynamic gesture dataset

4.3.1. Gesture classification

In the gesture classification task, the methods should identify a given gesture after the movement is entirely completed, i.e. using all the 14 frames. Using the LCS as extraction technique and the HMM as classification technique, a total of 52.55% of the gestures are recognized correctly, with a standard deviation of 5.7 and a classification time of 3.07 ms. The use of CLCS with HMM showed a classification rate of 81.66% with a standard deviation of 5.41 and took 0.79 ms for each classification. The CSURF combined with HMM produced a classification rate of 77.44% and a standard deviation of 4.81, each gesture being classied in 1.19 ms. Table 4 shows the results for the combinations using HMM in the classification module.

The HMM presented good results when in combination with the CLCS and the CSURF. The HMM is ideal for modeling the dy-

Table 5
Results for the dynamic gesture recognition classification using DTW in the classification module.

Techniques	Class. rate(%)	Standard deviation.	Class. time(ms)
LCS+DTW	89.06	4.88	1237.47
CLCS+DTW	97.00	2.70	63.37
CSURF+DTW	94.08	3.20	79.75

Table 6
Prediction results using CLCS with 1, 5, 10 and 14 frames.

Technique	Results	F1	F5	F10	F14
HMM	Prediction rate(%)	37.90	66.00	80.00	81.66
HMM	Standard deviation	6.50	5.50	4.60	5.41
HMM	Prediction time(ms)	0.10	0.42	0.83	1.18
DTW	Prediction rate(ms)(%)	17.18	17.29	64.06	97.00
DTW	Standard deviation	0.00	0.30	5.30	2.70
DTW	Prediction time(ms)	9.7	48.3	98.6	123.04

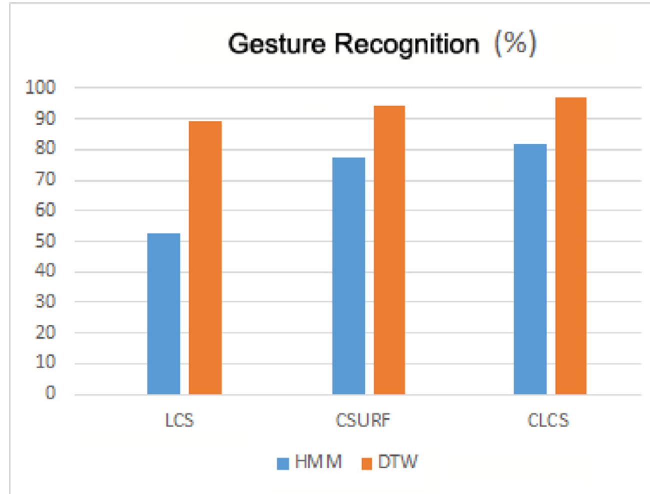


Fig. 12. Dynamic gesture recognition experiment results.

Table 7
Prediction results using CSURF with 1, 5, 10 and 14 frames.

Technique	Results	F1	F5	F10	F14
HMM	Prediction rate(%)	35.57	62.91	75.41	77.44
HMM	Standard deviation	4.60	5.56	4.60	4.81
HMM	Prediction time(ms)	0.09	0.44	0.86	1.23
DTW	Prediction rate(ms)(%)	16.53	18.43	85.31	94.08
DTW	Standard deviation	0.00	0.30	5.60	3.20
DTW	Prediction time(ms)	11.37	113.52	202.76	312.66

Table 8
Prediction results using LCS with 1, 5, 10 and 14 frames.

Technique	Results	F1	F5	F10	F14
HMM	Prediction rate(%)	36.14	44.27	52.65	52.55
HMM	Standard deviation	6.67	6.54	7.68	5.70
HMM	Prediction time(ms)	0.27	1.16	2.25	2.91
DTW	Prediction rate(ms)(%)	12.50	21.87	78.43	89.06
DTW	Standard deviation	2.47	2.20	3.00	4.88
DTW	Prediction time(ms)	89.39	444.30	876.76	1237.43

namics of the gestures, and could generate a better representation for each gesture when used with the smaller feature vector produced by the CLCS. When used along with techniques that produce a large amount of data, like the LCS, the results were worse. The classification time should be noted. When using the CLCS, the HMM had a classification time almost 35% lower than when using the CSURF. If compared with the LCS, it is almost 63% lower.

The combination of the LCS and the DTW presented a recognition rate of 89.06% with a standard deviation of 4.88, and this method takes 1,237.47 ms to classify each gesture. The use of the CSURF with the DTW obtained a recognition rate of 94.08%, the second highest. The standard deviation for this technique is 3.20 and it takes 79.75 ms to recognize each gesture. The CLCS with the DTW achieved the highest recognition rate of all the methods, 97.00%, with a standard deviation of 2.70, taking 63.37 ms to classify each gesture. Table 5 shows the results obtained with the DTW.

The Convexity Approach presents better results in combination with the DTW than when used with the HMM. One of the properties of the DTW is the comparison between sequences with different dimensions and length and when it is used with an ecient hand posture representation, high classification rates are shown because, in a smaller feature vector that contains strong hand representation data, the differences between sequences are easily highlighted. On the contrary, the time spent for the DTW to classify a gesture is almost 100 times greater than when using the HMM, because the DTW has to compare a sequence with each other sequence in the training set.

The experimental results demonstrate that the application of the Convexity Approach improved the recognition rates and computational time of the gesture recognition methods. This happens because the feature vector is minimized with the Convexity Approach, containing fewer features that are enough to represent a given gesture significantly. Thus, the classification techniques require less computational effort for classification. The proper representation of the hand, using less data, results in a better recognition rate. To illustrate the advantages of the Convexity Approach, Fig. 12 shows a summary of the recognition experiment.

4.3.2. Gesture prediction

In the gesture prediction experiments, the methods try to identify a given gesture before it is completed. Tables 6–8 present the prediction results with both HMM and DTW classiers for the feature extraction methods CLCS, CSURF, and LCS, respectively.

The main difference of the prediction is the use of an incomplete set of data. When using the CLCS, the data is still representative enough to give high prediction rates even with half of the frames presented. The same behavior occurs for the prediction of the CSURF and LCS. The difference is that the techniques where the Convexity Approach was applied obtained better results, showing that this technique enhances hand posture representation.

HMM presented the best classification rates with less than half of the data, but its accuracy increased slowly when more frames were presented, as shown in Fig. 13. This indicates that even with an incomplete feature vector, the HMM model is still able to infer the gesture using the CLCS representation. This behavior is also presented with the DTW, but only starting with half of the data. Using less than seven frames, the DTW is not capable of obtaining enough difference for each gesture. After half the frames are added to the feature vector, the DTW showed the best improvement by frame added. This shows that the DTW works better with sequences of smaller length variation. It is also important to note that when using CLCS for feature extraction, the HMM performs better than the DTW until the tenth frame.

For the prediction experiment, the same observation of the gesture classification task can be noted. The use of the Convexity Approach minimizes the time for prediction of each gesture. An interesting behavior of the prediction is that with half of the frames, the gesture is predicted with high accuracy. Fig. 13 shows the summary of the prediction experiment with the recognition rate achieved by each method for all the possible 14 frames.

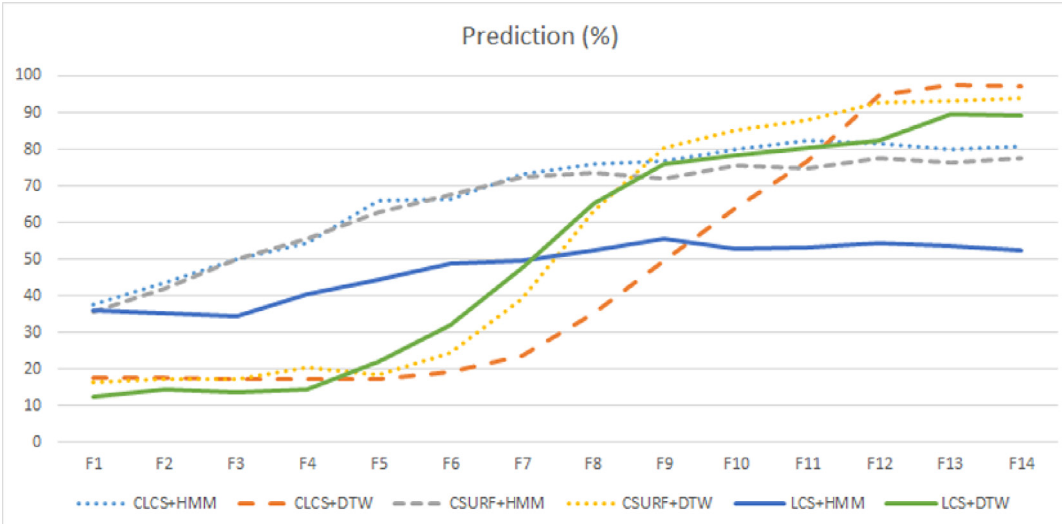


Fig. 13. Gesture prediction experiment results.

5. Discussion

We presented the Convexity Approach for dynamic gesture representation and evaluated its features using several techniques for recognition and prediction tasks. The Convexity Approach relies on the contour of the hand, and thus is strongly dependent on a proper segmentation method. In another hand, once the hand is segmented, the Convexity Approach showed to be very reliable to reduce the describe the hand shape, especially when compared with established techniques such as general image descriptors SURF and LCS.

The features extracted by the combination of Convexity Approach and LCS and SURF presented very promising results on recognition and prediction tasks, even overcoming state-of-the-art approaches on the complex Cambridge Hand Data. As we reduce the number of features, but still keep the descriptive aspect of the hand shape, our technique is shown to be a non-expensive option for hand gesture description. The use of techniques such as Hidden Markov Models and Dynamic Time Wrappers extends our descriptor to temporal domain, been able to deal with dynamic gestures. They also allow us to work with prediction tasks, where the gesture sequence is still not complete and the model should predict the correct output.

We also introduce the RPPDI Dynamic Gestures Dataset that consists of severals sequences of seven hand gestures. In this dataset, each gesture has a limited number of frames, which introduces a standard execution in the gestures. However, each gesture is executed at different speeds, which allow us to evaluate our models for very different gesture executions. We also evaluate our model using the Cambridge Hand Gesture Data set, which contains nine different hand gestures, but several executions in different illuminations.

When evaluating the model using the RPPDI Dynamic Gestures Dataset, we could see that our recognition system worked very well. We used this dataset to evaluate different algorithms for feature representation and classification and found out that the application of the Convexity Approach in the well known SURF algorithm produced the best results. That is explained by the nature of the Convexity Approach, in simplifying the shape representation but keeping the topological form of the hand. The SURF algorithm can represent the hand based on very complex gradient operations, and the Convexity Approach specify this representation for hand shapes.

The application of Hidden Markov Models and Dynamic Time Wrappers shows to be successful for classification tasks and very promising for prediction tasks. The fact that the Cambridge dataset has very different sequence lengths, varying from 10 frames to 180 frames, made the prediction task tough when very feel frames were present. That can be seen as a limitation of our model, because for the HMM and DTW work, we need normalized samples, which was not the case. In the case of the RPPDI dataset, the sequences have the same length, varying only on the speed of the gesture execution. Our architecture was able to deal with speed variances, and even with half of the frames present, it was able to have a prediction rate higher than 60%. That means that, our model can deal with speed variance, but not with time delay. The use of recurrent neural networks which can deal with the time lag, as the Long-Short Memory Recurrent Neural Networks (LSTMs) could help to solve this problem.

6. Conclusion

This study proposes a system for dynamic gesture recognition and prediction using the Convexity Approach technique for feature extraction. This method selects a minimal amount of points in the hand contour that can represent the hand shape. This selection is performed dynamically and nearly in real time, selecting different points per hand shape. Using the Convexity Approach in the hand contour obtained by other techniques, such as Local Contour Sequence (LCS) or Speed Up Robust Features (SURF), the feature vector is smaller which directly reflects in a faster and more accurate recognition of dynamic gestures.

The proposed system uses three modules, one for image preprocessing, one for feature extraction and one for classification. In prediction tasks, the system implements two different architecture structures and is used to recognize incomplete gestures. To evaluate the system, four feature extraction techniques were applied: LCS and SURF and the application of the Convexity Approach with both of them, creating the CLCS and CSURF. In the classification module, two techniques were implemented. Hidden Markov Models and Dynamic Time Warping were used. The combination of all these methods was evaluated and the results showed that the use of CLCS and CUSRF surpassed LCS and SURF in recognition and prediction tasks. The use of the Convexity Approach minimized the classification time and improved the recognition rate.

We compare our gesture classification with state-of-the-art approaches. We conclude that our gesture has similar results, but one very crucial characteristic that is not present in the other approaches: our method is capable of classifying gestures based on hand posture alone. This capability allows us to model any gesture, only by constructing the gesture sequence with different hand postures. One of the advantages of our approach is that the gestures can be executed at different speeds, or with different hand positions.

In future research, changes to make the model work in real time will be studied. The application of the Convexity Approach for the recognition of objects and human activity will be analyzed as well. The expansion of the system to integrate the prediction and recognition in one system will be implemented.

References

Ahmad, B.I., Murphy, J.K., Langdon, P.M., Godsill, S.J., Hardy, R., Skrypchuk, L., 2015. Intent inference for hand pointing gesture-based interactions in vehicles. IEEE Trans. Cybern. PP (99). 1–1.

Alon, J., Athitsos, V., Yuan, Q., Sclaroff, S., 2009. A unified framework for gesture recognition and spatiotemporal gesture segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 31 (9), 1685–1699.

Barros, P.V.A., Junior, N.T.M., Bisneto, J.M.M., Fernandes, B.J.T., Bezerra, B.L.D., Fernandes, S.M.M., 2013. Convexity local contour sequences for gesture recognition. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 34–39.

Barros, P.V.A., Junior, N.T.M., Bisneto, J.M.M., Fernandes, B.J.T., Bezerra, B.L.D., Fernandes, S.M.M., 2013. An effective dynamic gesture recognition system based on the feature vector reduction for surf and lcs. In: Proceedings of the 23rd International Conference on Artificial Neural Networks. Elsevier, New York, NY, USA, pp. 412–419.

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (surf). Comput. Vis. Image Underst. 110 (3), 346–359.

Bell, A.J., Sejnowski, T.J., 1997. The independent components of natural scenes are edge filters. Vision Res. 37 (23), 3327–3338. URL http://www.sciencedirect.com/science/article/pii/S0042698997001211.

Bernardes, J., Nakamura, R., Tori, R., 2009. Design and implementation of a flexible hand gesture command interface for games based on computer vision. In: Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on, pp. 64–73.

Bhattacharjee, A., Jana, I., Das, A., Kundu, D., Ghosh, S., Gupta, S.D., 2015. A novel probabilistic approach of colored object detection and design of a gesture based real-time mouse tracking along with virtual teaching intended for color-blind people. In: Signal Processing and Integrated Networks (SPIN), 2015 2nd International Conference on, pp. 512–519.

Bilal, S., Akmeliawati, R., El Salami, M., Shafie, A., 2011. Vision-based hand posture detection and recognition for sign language x2014; a study. In: Mechatronics (ICOM), 2011 4th International Conference On, pp. 1–6.

Bodiroza, S., Doisy, G., Hafner, V., 2013. Position-invariant, real-time gesture recognition based on dynamic time warping. In: Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on, pp. 87–88.

Brown, M., Susstrunk, S., Fua, P., 2011. Spatio-chromatic decorrelation by shift-invariant filtering. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on, pp. 27–34.

Chang, J. Y., 2016. Nonparametric feature matching based conditional random fields for gesture recognition from multi-modal video.

Chen, Z., Haykin, S., Eggermont, J.J., Becker, S., 2007. Correlative Learning: A Basis for Brain and Adaptive Systems. Wiley.

Cheng, J., Chen, X., Liu, A., Peng, H., 2015. A novel phonology-and radical-coded chinese sign language recognition framework using accelerometer and surface electromyography sensors. Sensors 15 (9), 23303–23324.

Chivers, D., Rodgers, P., 2011. Gesture-based input for drawing schematics on a mobile device. In: Information Visualisation (IV), 2011 15th International Conference on, pp. 127–134.

Cho, S.-J., Oh, J.-K., Bang, W.-C., Chang, W., Choi, E., Jing, Y., Cho, J., Kim, D.-Y., 2004. Magic wand: a hand-drawn gesture input device in 3-d space with inertial sensors. In: Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on, pp. 106–111.

Ciaramello, F., Hemami, S., 2011. A computational intelligibility model for assessment and compression of american sign language video. Image Process. IEEE Trans. 20 (11), 3014–3027.

Dekate, A., Kamal, A., Surekha, K.S., 2014. Magic glove - wireless hand gesture hardware controller. In: Electronics and Communication Systems (ICECS), 2014 International Conference on, pp. 1–4.

Douglas, D.H., Peucker, T.K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica 10 (2), 112–122.

Frolova, D., Stern, H., Berman, S., 2013. Most probable longest common subsequence for recognition of gesture character input. IEEE Trans. Cybern. 43 (3), 871–880.

Gupta, L., Ma, S., 2001. Gesture-based interaction and communication: automated classification of hand gesture contours. Syst. Man Cybern. Part C: Appl.Rev. IEEE Trans. 31 (1), 114–120.

Han, Y., 2010. A low-cost visual motion data glove as an input device to interpret human hand gestures. Consum. Electron. IEEE Trans. 56 (2), 501–509.

Harandi, M.T., Sanderson, C., Sanin, A., Lovell, B.C., 2013. Spatio-temporal covariance descriptors for action and gesture recognition. In: Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision (WACV). IEEE Computer Society, Washington, DC, USA, pp. 103–110. doi:10.1109/WACV.2013.6475006.

Hartigan, J.A., Wong, M.A., 1979. A k-means clustering algorithm. In: Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, pp. 100–108.

Hasan, H.S., Kareem, S., 2012. Human computer interaction for vision based hand gesture recognition: A survey. In: Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on, pp. 55–60.

Helander, E., Nurminen, J., 2007. A novel method for prosody prediction in voice conversion. In: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, vol. 4, pp. IV–509–IV–512.

Huang, Y., Monekosso, D., Wang, H., Augusto, J., 2011. A concept grounding approach for glove-based gesture recognition. In: Intelligent Environments (IE), 2011 7th International Conference on, pp. 358–361.

Husssain, A., Jameel, A., Al-Jumeily, D., Ghazali, R., 2009. Speech prediction using higher order neural networks. In: Innovations in Information Technology, 2009. IIT '09. International Conference on, pp. 294–298.

Ibraheem, N.A., Khan, R.Z., 2012. Article: survey on various gesture recognition technologies and techniques. Int. J. Comput. Appl. 50 (7), 38–44. Published by Foundation of Computer Science, New York, USA.

Javed, S., Ahmad, N.A., 2014. Adaptive total least squares based speech prediction. In: Information Technology and Multimedia (ICIMU), 2014 International Conference on, pp. 78–82.

Jeong, E., Lee, J., Kim, D., 2011. Finger-gesture recognition glove using velostat (iccas 2011). In: Control, Automation and Systems (ICCAS), 2011 11th International Conference on, pp. 206–210.

Jeong, S., Jin, J., Song, T., Kwon, K., Jeon, J.W., 2012. Single-camera dedicated television control system using gesture drawing. Consum. Electron. IEEE Trans. 58 (4), 1129–1137.

Kim, T.-K., Wong, K.-Y.K., Cipolla, R., 2007. Tensor canonical correlation analysis for action classification. In: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, pp. 1–8.

Koceski, S., Koceska, N., 2010. Vision-based gesture recognition for human-computer interaction and mobile robot's freight ramp control. In: Information Technology Interfaces (ITI), 2010 32nd International Conference on, pp. 289–294.

Kohlsdorf, D., Starner, T., Ashbrook, D., 2011. Magic 2.0: A web tool for false positive prediction and prevention for gesture recognition systems. In: Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, pp. 1–6.

Kouiroukidis, N., Evangelidis, G., 2011. The effects of dimensionality curse in high dimensional knn search. In: Informatics (PCI), 2011 15th Panhellenic Conference on, pp. 41–45.

Kovashka, A., Grauman, K., 2010. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2046–2053.

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

Lee, D.-H., Hong, K.-S., 2010. Game interface using hand gesture recognition. In: Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on, pp. 1092–1097.

Lee, S.-W., 2006. Automatic gesture recognition for intelligent human-robot interaction. In: Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on, pp. 645–650.

Leubner, C., Brockmann, C., Muller, H., 2001. Computer-vision-based human-computer interaction with a back projection wall using arm gestures. In: Euromicro Conference, 2001. Proceedings. 27th, pp. 308–314.

Liu, L., Shao, L., 2013. Synthesis of spatio-temporal descriptors for dynamic hand gesture recognition using genetic programming. In: Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on. IEEE, pp. 1–7.

Liu, S., Xiao, Q., 2015. A signer-independent sign language recognition system based on the weighted knn/hmm. In: Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015 7th International Conference on, vol. 2, pp. 186–189.

Melkman, A.A., 1987. On-line construction of the convex hull of a simple polyline. Inf. Process. Lett. 25 (1), 11–12.

Mori, A., Uchida, S., Kurazume, R., Taniguchi, R.-I., Hasegawa, T., Sakoe, H., 2006. Early recognition and prediction of gestures. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol. 3, pp. 560–563.

Nandakumar, G.N., Patel, N., Reddy, R., Kothandaraman, M., 2005. Application of douglas-peucker algorithm to generate compact but accurate ibis models. In: VLSI Design, 2005. 18th International Conference on, pp. 757–761.

Pagel, B.-U., Korn, F., Faloutsos, C., 2000. Deflating the dimensionality curse using multiple fractal dimensions. In: Data Engineering, 2000. Proceedings. 16th International Conference on, pp. 589–598.

Qaiyumi, S., Mirikitani, D., 2006. An extension of the k-mean algorithm for dealing with dimensionality curse. In: Engineering of Intelligent Systems, 2006 IEEE International Conference on, pp. 1–5.

Qin, H., Tang, S., 2009. A solution to dimensionality curse of bp network in pattern recognition based on rs theory. In: Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on, vol. 1, pp. 636–638.
Rabiner, L.R., 1990. In: Waibel, A., Lee, K.-F. (Eds.), Readings in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 267–296. chapter A tutorial on hidden Markov models and selected applications in speech recognition.
Rautaray, S., Agrawal, A., 2011. Interaction with virtual game through hand gesture recognition. In: Multimedia, Signal Processing and Communication Technologies (IMPACT), 2011 International Conference on, pp. 244–247.
Reuter, M., Biasotti, S., Giorgi, D., Patan, G., Spagnuolo, M., 2009. Discrete laplacebeltrami operators for shape analysis and segmentation. Comput. Graph. 33 (3), 381–390. {IEEE} International Conference on Shape Modelling and Applications 2009. URL http://www.sciencedirect.com/science/article/pii/S0097849309000272.
Sakoe, H., Chiba, S., 1990. In: Waibel, A., Lee, K.-F. (Eds.), Readings in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 159–165. chapter Dynamic programming algorithm optimization for spoken word recognition.
Santos, D.G., Fernandes, B.J., Bezerra, B.L., 2015. Hagr-d: a novel approach for gesture recognition with depth maps. Sensors 15 (11), 28646–28664.
Satya, K., Gogoi, A., Sahu, G., 2011. Regressive linear prediction with doublet for speech signals. In: Control System, Computing and Engineering (ICCSCE), 2011 IEEE International Conference on, pp. 33–36.
Schlecht, J., Carque, B., Ommer, B., 2011. Detecting gestures in medieval images. In: Image Processing (ICIP), 2011 18th IEEE International Conference on, pp. 1285–1288.
Sen, M., Corretjer, I., Haim, F., Saha, S., Bhattacharyya, S., Schlessman, J., Wolf, W., 2005. Computer vision on fpgas: Design methodology and its application to gesture recognition. In: Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on, p. 133.
Silanon, K., Suvonvorn, N., 2014. Finger-spelling recognition system using fuzzy finger shape and hand appearance features. In: Digital Information and Communication Technology and it's Applications (DICTAP), 2014 Fourth International Conference on, pp. 419–424.
Sklansky, J., 1982. Finding the convex hull of a simple polygon. Pattern Recogn. Lett. 1 (2), 79–83.
Soendoro, D., Supriana, I., 2011. Traffic sign recognition with color-based method, shape-arc estimation and svm. In: Electrical Engineering and Informatics (ICEEI), 2011 International Conference on, pp. 1–6.
Stavrakoudis, D., Theocharis, J., 2007. A recurrent fuzzy neural network for adaptive speech prediction. In: Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on, pp. 2056–2061.
Teoh, C.-C., Sheble, G., 2007. Lattice method of real option analysis - solving the curse of dimensionality and strategic planning. In: Power Tech, 2007 IEEE Lausanne, pp. 825–830.
Wu, D., Pigou, L., Kindermans, P.-J., Nam, L., Shao, L., Dambre, J., Odobez, J.-M., 2016. Deep dynamic neural networks for multimodal gesture segmentation and recognition.
Wu, X., Mao, X., Chen, L., Xue, Y., 2015. Trajectory-based view-invariant hand gesture recognition by fusing shape and orientation. IET Comput. Vision 9 (6), 797–805.
Wu, X., Xu, D., Duan, L., Luo, J., 2011. Action recognition using context and appearance distribution features. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 489–496.
Xinyu, W., Guan, S., Dong, H., Ting, Z., 2010. Data glove gesture recognition based on an improved neural network. In: Control Conference (CCC), 2010 29th Chinese, pp. 2434–2437.
Youfu, D., Guoan, T., 2010. Comparison of popular methods for simplifying terrain based on grid dems. In: Environmental Science and Information Application Technology (ESIAT), 2010 International Conference on, 3, pp. 227–230.
Zhang, G., Zhang, D., 2008. Research on vision-based multi-user gesture recognition human-computer interaction. In: System Simulation and Scientific Computing, 2008. ICSC 2008. Asia Simulation Conference - 7th International Conference on, pp. 1455–1458.
Zhao, L., Mo, Y., Pa, L., 2010. Research of characteristic vector based on hht in rotary machine failure diagnostics. In: Mechanic Automation and Control Engineering (MACE), 2010 International Conference on, pp. 2272–2275.
Zhou, Y., Chen, X., Zhao, D., Yao, H., Gao, W., 2008. Mahalanobis distance based polynomial segment model for chinese sign language recogniton. In: Multimedia and Expo, 2008 IEEE International Conference on, pp. 317–320.