A Dynamic Gesture Prediction System Based on The CLCS Feature Extraction

Nestor T. M. Júnior, Pablo V. A. Barros,
Bruno J. T. Fernandes, Byron L. D. Bezerra and Sergio M. M. Fernandes
Escola Politécnica de Pernambuco
Universidade de Pernambuco, Recife, Brasil
Email: {pvab, ntmj, bjtf, byronleite, smmf}@ecomp.poli.br

***Abstract*—Real-time recognition of dynamic gestures is a problem for most of the applications nowadays. The prediction approach can be used as a solution for this. This approach uses an incomplete gesture input and it tries to predict which gesture the given input represents. This paper presents the application of the dynamic gesture feature extraction technique called Convexity Local Contour Sequence (CLCS) as the extractor for the prediction task. Two predictor systems are used to achieve this task and results are compared and discussed in this paper.**

I. Introduction

Nowadays, human-computer interaction based on gesture recognition systems is becoming more usual. There are a lot of applications for this technology, such as video games, televisions, systems for impaired people, augmented reality, medical applications, among others.

One way to build gesture recognition systems is using computer vision techniques. This field of study provides the necessary algorithms and tools to capture, identify, learn and classify a gesture using a video camera. The work of Mitra and Acharya [1] shows several approaches to achieve gesture recognition using computer vision. It shows two kind of gesture recognition systems: static gesture recognition systems and dynamic gesture recognition system. Hasan et al. [2] specifies the works that contain dynamic gesture recognition systems based on hand gestures. These works showed that there are many techniques and approaches that could be used to recognize gestures, but only few can be used to recognize dynamic gestures. Hasan et al. showed that for hand posture based gestures, approaches using geometrical techniques to represent the gesture achieved better results than others, like fuzzy decision trees, transition movements or common sense context.

All the works discussed in Mitra and Acharya [1] and in Hasan et al. [2] surveys can recognize complete gestures but are not prepared to recognize incomplete gestures or gestures being executed in real time. This problem was described by Mori et al. [3]. They proposed a method for early recognition and prediction using a single and static representation for each gesture. It achieved a good success rate for gesture prediction, but it is not applicable for multi-user execution or even changes in the camera angle and/or hand position. They use a rigid representation of a gesture, describing the ideal gesture and thus is not applicable for slight changes in the execution.

The concept of prediction is useful in real time and natural environment applications. The prediction improves the recognition by trying to identify the pattern before it has been completely executed. It has been used in speech recognition systems with success as showed in the works of Stavrakoudis et al. [4], Hussain et al. [5], Varoglu and Hacioglu [6], Satya et al. [7] and Helander and Nurminen [8]. The approach used in those works can be adapted for gesture prediction: a learning model for the complete gesture and a feature extraction technique to extract only one portion of the gesture.

In our previous work [9] we presented a novel method, called CLCS, for dynamic gesture recognition based on hand postures. It can represent the hand posture and it uses the transition of postures to represent a dynamic gesture. The CLCS extracts only the features of one hand posture at each time, thus is ideal for gesture prediction. CLCS was tested with different classification techniques proving that it can be used in different scenarios of classification.

This paper shows the use of CLCS applied in a gesture prediction system. To learn and classify the partial executed gestures, two techniques were tested: Hidden Markov Model and Dynamic Time Warping.

This paper is structured as follows: Section II describes the CLCS algorithm and the prediction system. Section III presents the experimental results. Finally, in Section IV, the conclusions and some future work are given.

II. Prediction Model

A prediction system is able to classify an incomplete pattern. It is the ideal system to be used in real world applications, because it can predict a gesture without the full input been captured. It uses a partial captured pattern and classify it as one of the previously learned patterns.

This work uses a 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 in the system input and passed trough the Gesture Prediction module. The gesture is predicted for the partial input, and each new feature vector added produces a new prediction. Figure 1 shows the general architecture of the system.

978-1-4799-3194-1/13 $31.00 © 2013 IEEE
DOI 10.1109/BRICS-CCI-CBIC.2013.89

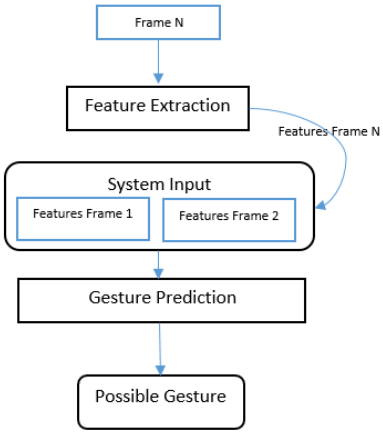

Fig. 1. Gesture Prediction System General Architecture

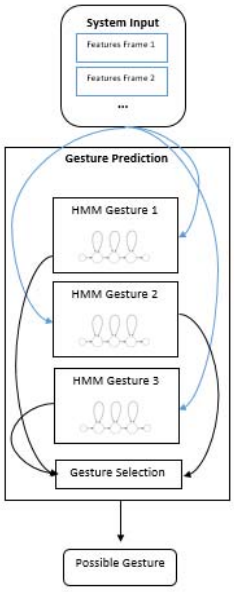

Fig. 2. HMM prediction system architecture.

To evaluate the architecture, two techniques are used for the gesture prediction task. The first uses a Hidden Markov Model (HMM) [10] to learn the full gestures and recognize the partial ones. The second prediction system uses a Dynamic Time Warping (DTW) [11] to calculate the distances between the gestures and find the predicted one.

A. 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 enough to the prediction task. It uses a K-means Clustering [12] to find the best initial approximation, this showed an improve in the final prediction rate. The Baum-Welch algorithm [13] is used to train the HMM resulting in a fast training process.

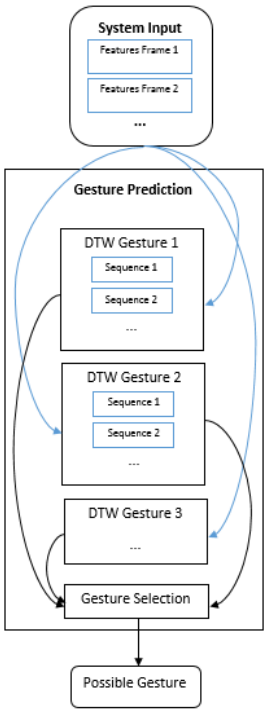

Fig. 3. DTW Prediction System architecture.

As shown in Figure 2, each new system input is passed trough all the HMMs and the output probability is calculated. This probability shows how close the input is to each HMM model. This probabilities are send to Gesture Selection and the model that has higher probability is selected as the predicted gesture.

B. 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 composes the full gesture representation. The distance 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 smaller average distance is selected as the predicted gesture. Figure 3 shows this system architecture.

The Simple DTW implementation [11] is used, and it presented good results in the prediction rate. This implementation uses a euclidean distance calculation to find the smaller distance between two sequences, thus the computational costs increases drastically as the input vector size increases.

C. CLCS

Convexity Local Contour Sequence (CLCS) is a method for hand feature extraction that can be applied in dynamic and

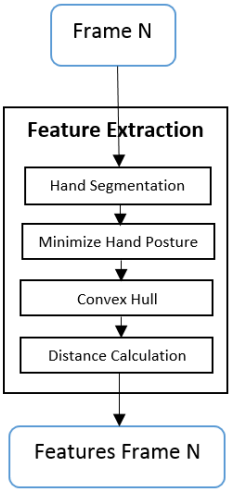

Fig. 4. CLCS execution illustration.

static gesture recognition. It uses the hand shape variations in the gesture movement to generate a descriptor of the gesture that can be used in classifications techniques. This is possible through the dynamic selection of the minimal amount of points able to represent the hand shape in each movement. This generates a representative model that is used to calculate the LCS and results in a gesture representation output vector.

The first step of CLCS is the hand segmentation. It removes the background of the image and find only the contour of the hand. The next step is to minimize the hand posture. The third step is to find the convex hull of the previously selected points. The last step uses the points that composes the convex hull for a feature calculation based on point distance. Figure 4 shows the execution illustration of the CLCS.

The input of the CLCS is composed by an image set of frames representing the hand gesture. Each frame is used as input and has its own feature vector. The vector descriptor of all image set is used to represent the entire gesture. It is showed in Figure 5(a).

The input is composed by raw images containing the entire hand and the background; the CLCS algorithm needs to identify and remove the latter. To accomplish this, the Otsu Threshold Technique [14] is used, resulting in a separated image containing only the hand shape in a binary representation. Even after applying Otsu, the image presents noise, especially in the edges. A Median Filter [15] is used, resulting in a smoothed image. The result is showed in Figure 5(b).

The binary image containing only the hand shape information is used to find the hand contour. An erode image is created and then subtracted of the original binary image. This creates a reduced image data that is more representative than the entire hand for the CLCS. This completes the first step and the result is showed in Figure 5(c).

The hand contour is used as input for the third step, hand minimization. First, the minimal number of points that can represent the gesture are selected. This is performed dynamically for each different hand position as the hand shape changes through gesture. This is accomplished using the Douglas-Peucker Algorithm [16] to create an approximation curve of the external points, forming a polygon that represents the gesture. This algorithm produces a minimized polygon for hand posture. In this algorithm, the two extreme endpoints of a set of points are connected with a straight line as the initial rough approximation of the polygon. Then, it approximates the whole polygon by computing the distance from all intermediate polygons vertexes to that line segment. If all these distances are less than the specified tolerance T, then the approximation is good, the endpoints are retained, and the other vertexes are eliminated. However, if any of these distances exceeds the T tolerance, then the approximation is not good enough. In this case, it chooses the point that is furthest away as a new vertex subdividing the original set points into two set points. This procedure is repeated recursively on these two shorter set points. If at any time, all of the intermediate distances are less than the T threshold, then all the intermediate points are eliminated. The routine continues until all possible points have been eliminated. Figure 5(d) shows the output of this step.

The next step starts selecting the most significant points for the specifically hand posture. We run the Sklansky's [17] algorithm in the last step output. The algorithm consists in the following sequence:

- The convex vertex of the polygon is found.
- The remaining n-1 vertexes are labeled in clockwise order starting at P0.
- Select P0, P1 and P2 vertexes and call then "Back", "Center" and "Front" respectively
- Execute the follow algorithm:
  - **while** "Front" is on vertix 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 each pair of selected points the algorithm traces a line. The farthest point of this line is selected as an inner point. Figure 5(e) shows the resultant points of the algorithm in an input image.

This step returns the feature extracted by a distance calculation. A line is formed by each pair of the external points chosen by the Sklansky's algorithm. The distance between this line and the closer inner point is calculated and added to the

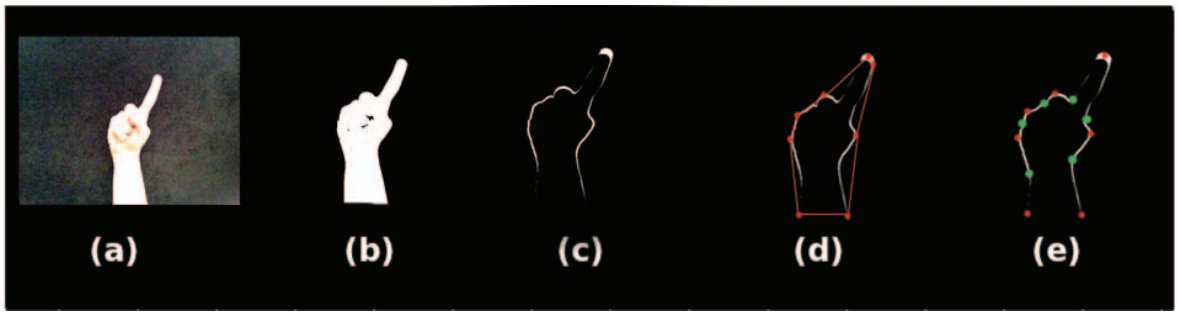

Fig. 5. Feature extraction algorithm sequence, starting at (a) as one of the images in the gesture image sequence. (b) Shows result image after the application of background removal. (c) Shows the result image at the end of second step, Find Contour. (d) Shows the result after the application of the third step, Feature Extraction, and it shows the minimized hand contour convex hull. (e) Shows the selected points to distance calculation.

output vector.

To get a reduction in the execution time of Dynamic Time Warping and Hidden Markov Model, a normalization technique is used. First, the number of normalized distances is defined. Then, for all the images that have fewer points than the previously determined length, a ”0” is added at the end of the 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 in calculate a window, W, as the division of the output length for the desired length. The vector of outputs is traversed and each position which is a multiple of W has it value 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.

To exemplify the normalization technique, imagine that a input containing one hundred elements is selected. The desired normalized distance is selected as eleven. As the input has more points than the normalized distance, the W window is calculated as 100/11 = 9. If the desired normalized distance were bigger than the input size, ”0” would be added to it until it reach the desired size. The input vector is visited and the positions 9, 18, 27, 36, 45, 54, 63, 72, 81, 100 are selected to compose the normalized vector. The normalized vector only has 10 elements, so a position is selected randomly, excluding the ones previously visited, and is added to the normalized vector.

III. Experimental Results

The efficiency and effectiveness of the CLCS applied to gesture prediction is presented in this section. We compare the results of the prediction of dynamic gestures using the two prediction models showed bellow.

A. Experimental Methodology

The database used for this test is the RPPDI Gesture Database[1]. It contains a set of seven different gestures. Figure 6 shows a gesture sequence. Each gesture is composed by 14 frames and the database has different sequences of each gesture.

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

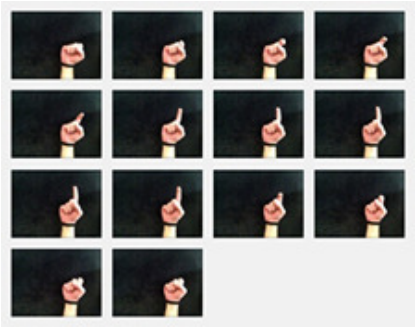

Fig. 6. Example of one gesture in the RPPDI database.

Each prediction system uses 66% of each gesture examples to learn the full gestures. The remaining 34% are used for tests. To evaluate the system each test gesture is used as input fourteen times, starting with only one frame and adding a new frame at each iteration, until every frame is considered. This produces fourteen results for each gesture test sequence and can show which the best frame/prediction rate ratio.

Each test is executed thirty times with randomly chosen sequences and the average recognition and execution time are showed.

B. Results

The collected results for each new frame added to the system are here presented, being possible to observe the evolution of the prediction rate and the evolution on the time needed for it. The first set of results is shown in Table I. It shows the results from frames 1 to 4 and, as can be seen, the HMM based prediction shows a continuous improvement in the prediction rate. Based on the HMM structural nature, each new frame contains additional data for recognition and it will increase the calculated probability. For the first four frames, the correct prediction rate advances in 20 %.

For the DTW based prediction system it shows that even with 4 frames it is not possible yet to identify which gesture

TABLE I
PREDICTION RESULTS FOR THE FIRSTS FOUR FRAMES IN EACH PREDICTION SYSTEM

System	Result	F1	F2	F3	F4
HMM	Predict. Rate(%)	37.9	43.6	49.9	54.4
HMM	Std. Dev.	6.5	5.6	4.1	5.3
HMM	Predict. Time(ms)	0.10	0.17	0.26	0.34
DTW	Predict. Rate(ms)(%)	17.18	17.42	17.06	17.34
DTW	Std. Dev.	0	0.27	0.70	0.47
DTW	Predict. Time(ms)	9.7	9.5	29.0	40.1

TABLE II
PREDICTION RESULTS FOR THE FRAMES FIVE TO EIGHT IN EACH PREDICTION SYSTEM

System	Result	F5	F6	F7	F8
HMM	Predict. Rate(%)	66.0	68.38	73.12	75.98
HMM	Std. Dev.	5.5	4.8	6.0	4.7
HMM	Predict. Time(ms)	0.42	0.50	0.59	0.67
DTW	Predict. Rate(ms)(%)	17.29	19.16	23.43	35.20
DTW	Std. Dev.	0.3	2.0	3.4	4.5
DTW	Predict. Time(ms)	48.3	59.0	68.8	80.6

is being executed.

The next set of results, shown in Table II, contains the results of the addition of the frames four to eight and confirms the constant evolution of the HMM prediction system. The prediction rate increases from 66%, for four frames, to 75%, with eight frames. The execution time is increased too, almost in a linear constant. The DTW based prediction system starts to increase the prediction rate only with five frames. Despite being irrelevant for the first 7, the recognition rate grows faster than the HMM one after that.

The correct recognition increases in this set still small, but it happens faster than the HMM.

The results for the addition of frames nine to twelve, shown in Table III, shows a stabilization in the HMM prediction system. It continuously increased the prediction result and reached a point of stabilization: 80% of recognition. The DTW based system shows a quick evolution in the prediction rate, overpassing the HMM results after twelve frames, reaching 94% of recognition.

The addition of the last two frames, showed in Table IV, confirms the stabilization of the HMM predictions system in 80%. The DTW system achieves a higher recognition rate: 98.84% with the entire set of frames.

This results show that the CLCS can be used for gesture prediction with the two selected prediction systems. The HMM based prediction system presented a better result for a early

TABLE III
PREDICTION RESULTS FOR THE FRAMES NINE TO TWELVE IN EACH PREDICTION SYSTEM

System	Result	F9	F10	F11	F12
HMM	Predict. Rate(%)	76.7	80.0	82.2	81.7
HMM	Std. Dev.	5.2	4.6	6.0	5.8
HMM	Predict. Time(ms)	0.76	0.83	0.92	1.00
DTW	Predict. Rate(ms)(%)	49.58	64.06	76.77	94.84
DTW	Std. Dev.	4.9	5.3	6.8	4.0
DTW	Predict. Time(ms)	90.23	98.06	104.87	80.6

TABLE IV
PREDICTION RESULTS FOR THE FRAMES THIRTEEN AND FOURTEEN

System	Result	F13	F14
HMM	Predict. Rate(%)	80.15	80.72
HMM	Std. Dev.	5.9	5.4
HMM	Predict. Time(ms)	1.09	1.18
DTW	Predict. Rate(ms)(%)	98.76	98.84
DTW	Std. Dev.	1.14	1.11
DTW	Predict. Time(ms)	115.34	123.04

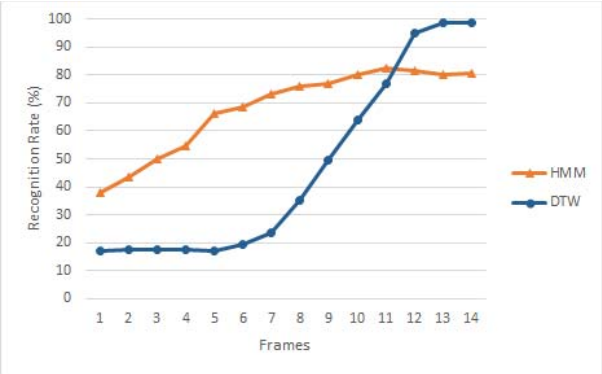

Fig. 7. DTW and HMM based prediction system correct prediction rate versus frames captured.

capture of the gesture, being able to achieve a classification rate higher than 70% with only half of the frames that composed the gesture. It also showed a gradual evolution in the recognition, which is shown in Figure 6 chart. This chart shows the evolution of the correct prediction rate versus the amount of frames captured.

It has observed the DTW prediction system outperform the HMM one, given a minimum number of frames as input. This happens because the sequence distance has a fast decrease with the additions of more frames to the captured input. It can be used for dynamic gesture prediction and reached a high correct prediction rate for 12 frames, higher than the HMM based one. However, the prediction evolution of this system is slow and it is adequate to use it only when you have almost all the frames captured or if a higher recognition rate is required. Figure 7 shows the evolution chart for the DTW based prediction system.

IV. CONCLUSION

The Convexity Local Contour Sequences(CLCS) creates a minimized feature vector for dynamic gesture input. It uses a dynamic selection of points, based on the point significance in the hand posture model. In this paper, it was applied for the dynamic gesture prediction task with two different dynamic gesture prediction systems: the first based on Hidden Markov Model and the second on Dynamic Time Warping.

The HMM based prediction system uses the HMM to model each gesture and the DTW one uses the distance of the input with a set of samples to predict a gesture. The RPPDI dynamic gesture dataset is used for evaluation. Each test consist in using

an incomplete version of the gesture, first containing only the first frame of each gesture, then two frames, and continuing until all the fourteen frames that represents a gesture are considered.

The results showed that the HMM based prediction system has a continuous improvement in the prediction, reaching 75.98% of correct predicted gestures for 8 frames, almost half of a complete gesture sequence. With the DTW based prediction system the recognition rate only reaches the same result of HMM when the 11th frame is added. However, the correct prediction rate for 12 frames is 94%, a higher rate than the HMM can reach.

This paper showed that the CLCS reaches a high prediction rate through the extraction of each hand posture at each time. With this characteristic it can be used to achieve a real time prediction task that can be used to help the final recognition task.

The future works can be listed as: test of the CLCS with other prediction system models, the improvement of the CLCS to achieve the feature extraction with complex background datasets and the application of this models in a larger full recognition system, containing facial expression recognition and tracking of others parts of the human body.

ACKNOWLEDGMENT

This work was partially supported by brazilian agencies: CNPq, CAPES and FACEPE. This work was developed with resources available in the project APQ-0949-1.03/10 - Reconhecimento de gestos uma aplicação para reconhecimento de sinais de surdos com dispositivos móveis sponsored by FACEPE.

REFERENCES

[1] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 3, pp. 311–324, 2007.

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

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

[4] D. Stavrakoudis and J. Theocharis, "A recurrent fuzzy neural network for adaptive speech prediction," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, 2007, pp. 2056–2061.

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

[6] E. Varoglu and K. Hacioglu, "Speech prediction using recurrent neural networks," *Electronics Letters*, vol. 35, no. 16, pp. 1353–1355, 1999.

[7] K. Satya, A. Gogoi, and G. Sahu, "Regressive linear prediction with doublet for speech signals," in *Control System, Computing and Engineering (ICCSCE), 2011 IEEE International Conference on*, 2011, pp. 33–36.

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

[9] P. V. A. Barros, N. T. M. 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*, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 34–39. [Online]. Available: http://doi.acm.org/10.1145/2480362.2480371

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

[11] H. Sakoe and S. Chiba, "Readings in speech recognition," A. Waibel and K.-F. Lee, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, ch. Dynamic programming algorithm optimization for spoken word recognition, pp. 159–165. [Online]. Available: http://dl.acm.org/citation.cfm?id=108235.108244

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

[13] 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.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 1, pp. 62 –66, jan. 1979.

[15] R. Gonzales, *Processamento Digital de Imagens*, 3rd ed. Pearson Education, 2010.

[16] P. S. Heckbert and M. Garland, "Survey of polygonal surface simplification algorithms," 1997.

[17] J. Sklansky, "Finding the convex hull of a simple polygon," *Pattern Recogn. Lett.*, vol. 1, no. 2, pp. 79–83, Dec. 1982. [Online]. Available: http://dx.doi.org/10.1016/0167-8655(82)90016-2