Static Gesture Recognition Combining Graph and Appearance Features

Marimpis Avraam
Computer Science Department
Aristotle University of Thessaloniki, AUTH
Thessaloniki, Greece

***Abstract*—In this paper we propose the combination of graph-based characteristics and appearance-based descriptors such as detected edges for modeling static gestures. Initially we convolve the original image with a Gaussian kernel and blur the image. Canny edges are then extracted. The blurring is performed in order to enhance some characteristics in the image that are crucial for the topology of the gesture (especially when the fingers are overlapping). There are a large number of properties that can describe a graph, one of which is the adjacency matrix that describes the topology of the graph itself. We approximate the topology of the hand utilizing Neural Gas with Competitive Hebbian Learning, generating a graph. From the graph we extract the Laplacian matrix and calculate its spectrum. Both canny edges and Laplacian spectrum are used as features. As a classifier we employ Linear Discriminant Analysis with Bayes' Rule. We apply our method on a published American Sign Language dataset (ten classes) and the results are very promising and further study of this approach is imminent from the authors.**

***Keywords*— Gesture Recognition; Neural Gas; Linear Discriminant Analysis; Bayes Rule; Laplacian Matrix**

I. Introduction

The purpose of a gesture recognition process is to interpret gestures performed by humans [24]. The domain of such processes can vary significantly, from remote robot control, virtual reality worlds to smart home systems [25]. Most events that are recognized as gestures originate from the body's motions or state, but with most commonly origin the face or the hands.

A topic highly interwoven with gesture recognition and it is equally significant is the sign language recognition (SLR). A sign language is composed of three types of features. Manual features such as, hand shapes, their orientation and movement. Non-manual features are related to arms, the body and the facial expressions. Finally, finger-spelling that corresponds to letters and words in natural languages [19]. Specifically the non manual features are used to both individually form part of a sign or support other signs and modify their bearing meaning [26].

The ultimate purpose of sign language recognition systems is to build automated systems that are able recognize signs and convert them into text or even speech. This could ideally lead to a translation system to communicate with the deaf people. Finally, another challenge in this field is that there is no international sign language, thus every region defines its own and there is no imposed constraint that is should be based on the spoken language of the region [27].

In the literature have been proposed methods that utilize features extracted in real time and identify the gestures as they are performed. We, on the other, will focus on a series of static images. Our purpose is to demonstrate a novel combination of features, classic appearance-based characteristics and graph descriptors (e.g. adjacency matrix). Through experimentation, we concluded that these features are sufficient to describe in a discriminative way a gesture.

II. Related Work

Soft computing techniques include a plethora of well-established algorithms such as Self-Organizing Neural Gas (SGONG). In [17], the authors employ SGONG by generating an approximation of the hand topology. Based on a number of presumptions such as the hand should be placed vertical, etc they produce a statistical model to recognize the gestures. Moreover, they are able to identify the fingers individually by the number of neighbors that a neuron has. By combining Self-Organizing Maps (SOM) and Markov chains in [18] the authors propose a novel probabilistic classification model. SOMs are used to model the position of the gesture while via a quantization process they describe the hand direction. These features are cast in discrete symbols and then used to construct a Markov model for each hand.

In relation to image processing methods and edge detection methods specifically, in [23] Canny edges in order to produce thinned skeletons of the gestures. While in [19] the Canny edges are further processed by Hough transformation in order to conclude into the gesture features. Simpler but very effective, in [21] the authors utilize the difference between the detected (strong & weak) edges as features. An extensive work, based solely on edges as presented in [20], by which the edges are processed and clip in order to describe the fingers individually. Edges have also been proposed in conjunction with other low level features such as salient points and lines for sparse feature representation in [22].

Each of these proposals manages achieve very high recognition rates utilizing either graph or appearance-based features but not an explicit combination of the two. To our humble knowledge our proposal is one of the few, if not the only one, gesture recognition system (even testing on a simple dataset) that combines appearance-based features and graph properties simultaneously.

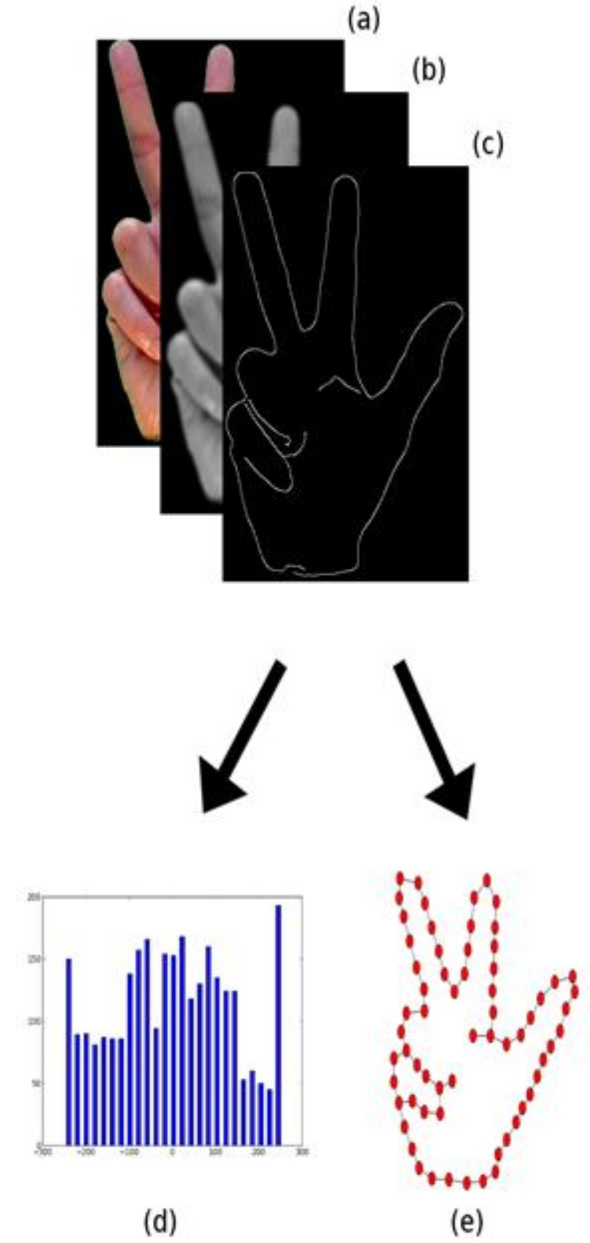

Fig. 1. (a) Orignal gesture image. (b) Grayscaled and passed with a Gaussian Blur filter. (c) Canny edges detected. (e) Histogram of 1 dimensional PCA projection. (f) Graph extracted by Neural Gas with CHL.

III. Proposed Method

As stated earlier our method utilizes appearance-based features as well as properties that are induced from graphs. On one hand the Canny edges provide a more general and rich description, without any specificity. On the other, the graph-related features are sufficient to describe the topology of a gesture, but fail to discriminate gestures with similar topographies. We concluded to this combination because these features complement each other.

In the following paragraphs we will provide a detailed description of our system's overview.

A. *Image Preprocessing*

Before anything else, we have to segment the hand from the image. Because of the nature of the specific dataset, a simple thresholding is sufficient; the background is just the black color.

The dimensions of the images in the dataset [4] are variable, because a number of transform operations have been applied. Some gestures have bee scaled along the X or Y axis while others have been rotated. The only property that we modify is the dimension of each image. We resize all images to $256x256$ pixels in order to restrict the possible variances of the generated graph topologies.

Finally, in order to restrict the Canny edge detector that we will user later on, we convolve a Gaussian blur kernel with the image as suggested in [16]. This way, we reduce the qualitative resolution of image and the edge detector will highlight only what we have issued to be, the most significant edges on the hand. The edges on the fingers will be greatly enhanced.

B. *Appearance-based features*

In Canny edge detector algorithm [3], initially the raw image is convolved with a Gaussian filter in order to reduce its susceptibility to noise. Then, utilizing two filters for each of the directions as in horizontal and vertical it computes the gradient [13]. Finally, it calculates the magnitude and the orientation of the gradient and with the usage a simple threshold technique it suppresses edges with low values.

For our purposes, after the edges are extracted we use PCA [14] and project them into one dimension and calculate the histogram.

C. *Graph-induced features*

1) Neural Gas with Competitive Hebbian Learning (NGCHL)

Martinetz and Schulten introduced the Neural Gas manifold learning algorithm in [5] and were inspired by Kohonen's Self Organizing Maps. The most notable difference between these two is that the first does not define explicitly a neighborhood. To overcome this issue an enhanced version of Neural Gas was proposed, integrating Competitive Hebbian Learning (CHL) [1, 2].

This new alternative approach follows closely the description of Neural Gas, that of a feature manifold learning method but through the generation of topology preserving maps. The learning function that is incorporated into the algorithm is described by a gradient descent as in the original algorithm. As in the original Neural Gas, the projection of a feature vector marks the adaptation of all neurons with a step size from the closest to the furthest. The key difference is that the Neural Gas with CHL also creates or destroys the synaptic links between the neurons. Each such connection is described by a synaptic strength. This indicator is used to decide whether or not a connection should be refreshed and kept or removed from the graph.

Below is a short description of the Neural Gas with Competitive Hebbian Learning algorithm. We must declare and set the following parameters: $T_{\max}$ (maximum number of iterations), α_T (final edge age), a_0 (initial edge age), e_T (final learning rate), e_0 (initial learning rate), λ_T (initial neighborhood reference), λ_0 (final neighborhood reference).

Create a number of neurons and assign them random positions and with no connections between them (the adjacency matrix is empty).

Step 1: Project $\vec{x}$ (drawn from R^n space) to our network.

Step 2: Sort the nodes based on their distances from $\vec{x}$ (keep the relative index position to k).

Step 3: Adapt the nodes' position vectors ($\vec{w}$) using:

$$N_{\vec{w}} \leftarrow N_{\vec{w}} + [N_{\vec{w}} * e(t) * h(k) * (\vec{x} - N_{\vec{w}})] \quad , \quad \forall N \in GraphNodes$$

where: $h(t) = \exp \frac{-k}{\sigma^2}(t)$, $\sigma^2 = \lambda_0 (\frac{\lambda_T}{\lambda_0})^{\frac{t}{T_{\max}}}$,

$$e(t) = e_0 (\frac{e_T}{e_0})^{\frac{t}{T_{\max}}}$$

Step 4: Connect the two closest neurons (according to their distance from $\vec{x}$). If already connected, refresh the age of the connection to 0.

Step 5: Increase the age by 1, of all edges connected with the best matching unit (the neuron closest to $\vec{x}$).

Step 6: Check for old edges in the graph and remove them if their age is exceeding the threshold given by the following formula:

$$\alpha(t) = \alpha_0 (\frac{\alpha_T}{\alpha_0})^{\frac{t}{T_{\max}}}$$

The above function is a gradient descent. It uses the ranking of nodes in the aggregation of step 3. Increase the iteration counter and repeat from step 2 until the required criteria are met, usually $t = T_{\max}$.

2) Graph Spectra

Having obtained an undirected graph from NGCHL we are now able to apply specific linear algebra concepts. One of the most interesting notions in graph theory is the Laplacian matrix [6, 7].

In simple terms, given an adjacency matrix A and a degree matrix D , the Laplacian matrix is given by: $L = D - A$. These matrices have a numerous variations, properties and applications but these topics are out of the scope of this paper.

We would like to use the Laplacian matrix in order to extract its spectrum [8]. The computed Eigenvalues are very important and widely researched (especially the λ_2) [9] in the field of graph analysis and theory and can be employed when comparing graphs [12]. In our case we employ all of the Eigenvalues as a strong discriminative feature.

D. Linear Discriminant Analysis (LDA) and Bayes Optimal Classification

In our proposed method we use LDA to estimate the within class density (based on the assumptions that the distributions are Gaussians and the classes share a common covariance matrix), resulting into a linear classifier model [10, 11].

Based on Bayes' rule, we will assign a gesture to the class with the maximum posterior probability given the feature vector $\vec{X}$. So the conditional probability of class G given X is:

$$\Pr(G = k \mid X = x) = \frac{fk(x)\pi k}{\sum_{l=1}^{k} fl(x)\pi l}$$

Based on the earlier assumption of the normal distributions we can derive the following linear discriminant model:

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^t \Sigma^{-1}(x - \mu_i) + \ln P(\omega_i)$$

This assigns a feature vector x into class i with the maximum g_i . Notice that the above equation contains a term that it is a Mahalanobis distance between the feature vector to and each the c mean vectors, and assign x to the category of the nearest mean [10, 15].

E. American Sign Language Dataset

We applied our proposed in the dataset proposed in [4]. It is a dataset containing 36 American Signs (ASL) each one of which is performed by five different persons. Each gesture is captured using a neutral-colored background (a green background), making the segmentation easier. This enables us to focus our research on the feature extraction and the classification methodology. In order to extend the dataset and provide more realistic (as "in the wild") poses, the original authors of the dataset, rotated each image in different angles and scales resulting into 2425 images.

In this paper, we focus on the ten classes corresponding to the numbering gestures from one to ten. Each class is composed of 65 samples.

IV. Experimental Results

We applied our proposal in the aforementioned dataset. Ten different gestures (classes), each one a gesture sign counting from zero to nine. The topographical network graph was set in 128 nodes. Finally as far the classification is concerned we followed a 5-fold cross validation scheme. For training we used 50 samples, while the rest 15 composed the testing set. Below we tabulate the results acquired.

TABLE I. CLASSIFICATION ACCURACY

Bayes Classification	
Gesture Class	***Accuracy Score***
0	93%
1	86%
2	73%
3	65%
4	100%
5	93%
6	86%
7	80%
8	86%
9	69%

[a.] Mean accuracy of a 5-cross validation procedure.

The results look very promising. Two gestures, namely the "3" and the "9" seem to perform poorly compared to others. The reason for this, is that both gestures are similarly signed (performed) resulting into also similar topographic graphs.

V. FUTURE WORK

We could possibly improve the performance by considering a method that respects the geodesic distances of the manifold, meaning that the edges would be regulated. This would greatly adjust the generated graphs and ultimately it would refine the quality of the extracted spectrums. We believe that this would result into greater results. In a more extensive work we will integrate a skin detection and segmentation algorithm possible based on a Gaussian Mixture Model that have been widely utilized in the literature. This will help build a more complete system. Another field that enjoys a lot of attention is the usage of wavelets (discrete) in order to extract low approximations of an image. This could result into a new system, in which the edges from the vertical and/or horizontal approximations are used.

VI. CONCLUSION

In this paper we presented the combination of the some properties and descriptors that can be extracted from a topology graph with Canny edges. A small topology results to a highly connected (or even fully) graph where all nodes are close to each other on the other hand, in a large topology all nodes spread to cover the topology and the connections are sparse, usually limited to their nearest neighbors. Based on the adjacency and degrees matrices we extract the Laplacian spectrum. This fusion yield very satisfactory results based on a five-fold cross validation procedure in a ten-class dataset.

REFERENCES

[1] Martinetz, T., & Schulten, K. (1994). Topology representing networks. Neural Networks, 7(3), 507-522.

[2] Martinetz, T. (1993). Competitive Hebbian learning rule forms perfectly topology preserving maps. In ICANN'93 (pp. 427-434). Springer London.

[3] Canny, J. (1986). A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6), 679-698.

[4] Barczak, A. L. C., Reyes, N. H., Abastillas, M., Piccio, A., & Susnjak, T. (2011). A new 2D static hand gesture colour image dataset for asl gestures.

[5] Martinetz, T., & Schulten, K. (1991). A" neural-gas" network learns topologies (pp. 397-402). University of Illinois at Urbana-Champaign.

[6] Newman, M. E. (2006). Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 103(23), 8577-8582.

[7] Newman, M., Barabási, A. L., & Watts, D. J. (Eds.). (2006). The structure and dynamics of networks. Princeton University Press.

[8] Newman, M. (2009). Networks: an introduction. Oxford University Press.

[9] Mohar, B., & Alavi, Y. (1991). The Laplacian spectrum of graphs. Graph theory, combinatorics, and applications, 2, 871-898.

[10] Duda, R. O., Hart, P. E., & Stork, D. G. (2012). Pattern classification. John Wiley & Sons.

[11] Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer, 27(2), 83-85.

[12] Koutra, D., Parikh, A., Ramdas, A., & Xiang, J. (2011). Algorithms for Graph Similarity and Subgraph Matching.

[13] Semmlow, J. L. (2004). Biosignal and biomedical image processing: MATLAB-based applications (Vol. 22). CRC press.

[14] Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1), 37-52.

[15] Teknomo, Kardi. Discriminant Analysis Tutorial. http://people.revoledu.com/kardi/ tutorial/LDA/

[16] Hai, W. (2002). Gesture recognition using principal component analysis, multi-scale theory, and hidden Markov models (Doctoral dissertation, Dublin City University).

[17] Stergiopoulou, E., & Papamarkos, N. (2009). Hand gesture recognition using a neural network shape fitting technique. Engineering Applications of Artificial Intelligence, 22(8), 1141-1158.

[18] Caridakis, G., Karpouzis, K., Drosopoulos, A., & Kollias, S. (2010). SOMM: Self organizing Markov map for gesture recognition. Pattern Recognition Letters, 31(1), 52-59.

[19] Munib, Q., Habeeb, M., Takruri, B., & Al-Malik, H. A. (2007). American sign language (ASL) recognition based on Hough transform and neural networks. Expert Systems with Applications, 32(1), 24-37.

[20] Ravikiran, J., Mahesh, K., Mahishi, S., Dheeraj, R., Sudheender, S., & Pujari, N. V. (2009, March). Finger detection for sign language recognition. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1, pp. 18-20).

[21] MANISHA, L., SHETE, V., & SOMANI, S. (2013). SOFT COMPUTING APPROACHES FOR HAND GESTURE RECOGNITION. International Journal of Computer Science.

[22] Georgiana, S., & Caleanu, C. D. (2013, July). Sparse feature for hand gesture recognition: A comparative study. In Telecommunications and Signal Processing (TSP), 2013 36th International Conference on (pp. 858-861). IEEE.

[23] Barkoky, A., & Charkari, N. M. (2011, July). Static hand gesture recognition of Persian sign numbers using thinning method. In Multimedia Technology (ICMT), 2011 International Conference on (pp. 6548-6551). IEEE.

[24] Chaudhary, A., Raheja, J. L., Das, K., & Raheja, S. (2011). A survey on hand gesture recognition in context of soft computing. In Advanced Computing (pp. 46-55). Springer Berlin Heidelberg.

[25] Chaudhary, A., Raheja, J. L., Das, K., & Raheja, S. (2013). Intelligent approaches to interact with machines using hand gesture recognition in natural way: a survey. arXiv preprint arXiv:1303.2292.

[26] Cooper, H., Holt, B., & Bowden, R. (2011). Sign language recognition. In Visual Analysis of Humans (pp. 539-562). Springer London.

[27] Dreuw, P., Forster, J., Gweth, Y., Stein, D., Ney, H., Martinez, G., ... & Wheatley, M. (2010, May). Signspeak–understanding, recognition, and translation of sign languages. In Proceedings of 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (pp. 22-23).