# A DISCRIMINATIVELY TRAINED HOUGH TRANSFORM FOR FRAME-LEVEL PHONEME RECOGNITION

*Jonathan Dennis, Huy Dat Tran, Haizhou Li*

Institute for Infocomm Research, A*STAR,
1 Fusionopolis Way,
Singapore 138632

*Eng Siong Chng*

School of Computer Engineering,
Nanyang Technological University,
Singapore 639798

## ABSTRACT

Despite recent advances in the use of Artificial Neural Network (ANN) architectures for automatic speech recognition (ASR), relatively little attention has been given to using feature inputs beyond MFCCs in such systems. In this paper, we propose an alternative to conventional MFCC or filterbank features, using an approach based on the Generalised Hough Transform (GHT). The GHT is a common approach used in the field of image processing for the task of object detection, where the idea is to learn the spatial distribution of a codebook of feature information relative to the location of the target class. During recognition, a simple weighted summation of the codebook activations is commonly used to detect the presence of the target classes. Here we propose to learn the weighting discriminatively in an ANN, where the aim is to optimise the static phone classification error at the output of the network. As such an ANN is common to hybrid ASR architectures, the output activations from the GHT can be considered as a novel feature for ASR. Experimental results on the TIMIT phoneme recognition task demonstrate the state-of-the-art performance of the approach.

***Index Terms***— Phoneme recognition, TIMIT, object detection, Hough Transform

## 1. INTRODUCTION

Hybrid ASR architectures have achieved many state-of-the-art results in recent years, in particular those using deep neural networks (DNNs) [1, 2, 3]. However, the features have remained unchanged from the simple MFCC or filterbank features that have been ubiquitous for decades. Despite their strong performance, training DNNs with such features has also proved computationally expensive and the information extracted at each layer of the network is difficult to interpret.

In this paper, we propose a novel ASR approach that is inspired by our previous work on using the GHT for sound event recognition [4]. The GHT is common in image processing, where it is among the state-of-the-art for object detection [5, 6]. Our idea is to combine the flexibility of the GHT for modelling the spatial distribution of feature information, with the modern hybrid-ANN architecture used in state-of-the-art ASR systems. This is achieved by placing the GHT in a discriminative framework, whereby an ANN learns the weighted mapping between the input codebook cluster activations and the target phoneme states to directly optimise the static phone-state classification performance. This has parallels with the "Max-Margin" Hough Transform presented in [6], where a formulation similar to that found in SVM was used to learn the discriminative weighting. The advantage of our approach is twofold. Firstly, the discriminative weights are learned in a framework that can be used directly in modern hybrid ASR systems. Secondly, the front-end processing of our approach enables a wide range of feature representations to be extracted from the speech data, such as formant-based features.

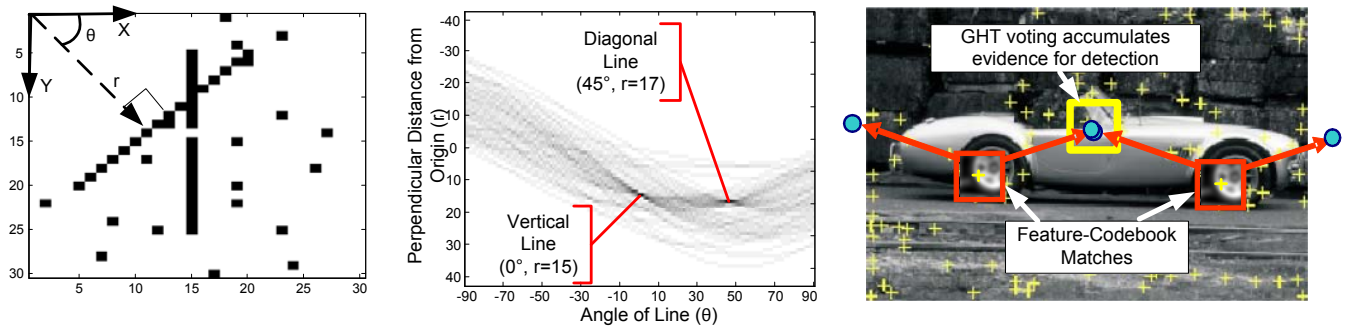
The remainder of this work is organised as follows. Section 2 introduces the GHT as a framework for hybrid ASR. Section 3 details the experimental evaluation on the TIMIT speech database. Section 4 then concludes the work.

## 2. GENERALISED HOUGH TRANSFORM FRAMEWORK

### 2.1. Introduction to the Hough Transform

The Hough transform (HT) was originally designed to detect parametrised lines and curves [7], and was only expanded later to cover arbitrary shapes through the GHT [8]. To understand the detection mechanism, consider the simple example shown in Fig. 1a, which contains two lines against a noisy background. Here, each point at location $l_i = [x_i, y_i]$ is considered as a feature, and casts votes for possible lines into the Hough accumulator in Fig. 1b. As limited feature information is available, the voting function is simply the distribution of all possible straight lines that the point could belong to, covering all possible rotations: $-90 < \theta < 90$. Using the polar equation of a straight line, the Hough accumulator, $H(r, \theta)$, is therefore the summation of the evidence as follows:

$$H(r,\theta) = \sum_{l_i} \begin{cases} 1, & \forall r = x_i \cos\theta + y_i \sin\theta \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$




(a) Image containing two lines and noise.

(b) Hough accumulator showing the presence of the two lines detected in the image in (a).

(c) Extension to the Generalised Hough Transform with voting based on feature-codebook matching.

**Fig. 1**. Simple example of the Hough transform for overlapping straight lines in noise. The result is two strong local maxima in the Hough accumulator indicating the hypotheses for the two lines.

where $r$ is the perpendicular distance from the origin and $\theta$ the angle from the horizontal axis. Local maxima in the Hough accumulator in Fig. 1b correspond to the combined evidence from the individual points in the image for a line with a given $(r, \theta)$. The Hough transform has two desirable properties. Firstly, the voting procedure is a summation, as opposed to multiplication, hence missing points on the line do not adversely affect the result. Secondly, the Hough accumulator space is sparse and separable, such that a strong peak occurs for each line representing evidence for the detection.

### 2.2. Generalised Hough Transform

The extension of the Hough transform to the GHT is shown in Fig. 1c, which allows the detection of arbitrary shapes that cannot be represented by an analytical equation. Features are first matched against a codebook that is learnt during training. This codebook stores both the feature template and a spatial voting function, which models the distribution of the codebook entry in the training, relative to the occurrences of each target class to be detected. During recognition, the matching codebook entries for each feature casts votes for possible locations of the target into the Hough accumulator for each target class. As before, this voting is a weighted summation of evidence, and peaks in the accumulator correspond to detections for the target class. This therefore maintains the key principles of the Hough transform, such as independent feature voting and a sparse and separable accumulator space.

In previous works on the implicit shape model [5, 6], the GHT is cast into a probabilistic framework. Following the notation used in [6], let $f_j$ denote a feature observed at location $l_j$, where $l$ represents the time-index of the feature in the case of frame-based audio processing. In addition, let $S(O_n, x)$ denote the score of target class $O_n$ at time location $x$, and let $C_i$ denote the $i$th codebook entry representing the feature space. While there can be considerable flexibility in defining the codebook and features, for simplicity and understanding in the context of speech recognition, it can be assumed that the features $f_j$ are MFCCs, and the codebook $C$ consists of GMM models trained to represent the state-level feature observations in an HMM.

The first step is to match the feature $f_j$ against the codebook $C_i$. Only a subset of the codebook entries will be matched, each with probability $p(C_i|f_j, l_j)$. Next, every matching codebook entry casts votes for possible locations $x$ of the associated target classes $O_n$ using their learned spatial distributions $p(O_n, x|C_i, l)$. The overall Hough accumulator score $S(O_n, x)$ is then obtained by adding up the individual probabilities over all the feature observations:

$$S(O_n, x) = \sum_{i,j} p(O_n, x|C_i, l_j)p(C_i|f_j, l_j) \qquad (2)$$

This can be simplified using the fact that the matching of the codebook entries is independent of their location, and can be further expanded to give the following:

$$S(O_n, x) = \sum_{i,j} p(x|O_n, C_i, l_j)p(O_n|C_i, l_j)p(C_i|f_j) \qquad (3)$$

The first term is the probabilistic Hough vote for a target class position given its label and the feature interpretation. In our experiments, we maintain a binned estimate of $p(x|O_n, C_i, l_j)$ within a temporal context window relative to the target class location. The second term specifies a confidence that the codebook entry $C_i$ at location $l_j$ is really matched on the target class. Finally, the last term is the likelihood that the codebook entry $C_i$ generated the feature $f_j$ . As suggested in [5], we base this on a Gibbs-like distribution of the distance of the codebook entry to the feature as follows:

$$p(C_i|f) = \begin{cases} \frac{1}{Z} exp(-\gamma d(C_i, f)), & \text{if } d(C_i, f) \geq t \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

where $Z$ is a normalising constant to make $p(C_i|f)$ a probability distribution, $d(C_i, f)$ is the negative log-likelihood of the codebook match, and $\gamma, t$ are positive constants.

### 2.3. Discriminative Weighting

In equation (3), the second term $p(O_n|C_i, l_j)$ is a weighting that captures how confident we are that the codebook entry $C_i$ at location $l_j$ matches the class $O_n$ as opposed to the rest. Assuming that $p(O_n|C_i, l_j)$ is independent of the $l_j$, then a simple method for estimating the weights is based on the relative frequency of the codebook entry across the different target classes as follows [6]:

$$p(O_n|C_i, l) = p(O_n|C_i) \propto \frac{p(C_i|O_n)}{p(C_i)} \tag{5}$$

where $p(C_i|O_n)$ is the relative frequency of codebook entry $C_i$ in target class $O_n$, while $P(C_i)$ is the relative frequency across all of the training data.

An improved method would be to learn this weighting in a discriminative framework to optimise the classification performance across all of the target classes. This was the approach taken in the max-margin framework of [6], where it was shown that equation (3) can be rewritten by factoring $p(O_n|C_i)$ as a separate summation as follows:

$$\begin{aligned} S(O_n, x) &= \sum_i p(O_n|C_i) \sum_j p(x|O_n, C_i, l_j) p(C_i|f_j) \\ &= \sum_i w_{i,n} \times a_{i,n}(x) = w^T A_n(x) \end{aligned} \tag{6}$$

where $A_n^T = [a_{1,n}, a_{2,n}, a_{i,n}, \ldots a_{K,n}]$ is the activation vector across the $i = 1, 2, \ldots K$ codebook entries, and $a_{i,n}$ is given by the following equation:

$$a_{i,n}(x) = \sum_j p(x|O_n, C_i, l_j) p(C_i|f_j) \tag{7}$$

Unlike the max-margin approach in [6], here we propose to use an ANN to learn the discriminative weighting $w_{i,n}$. The input layer of the network is the activation vector $A_n$, while the output is the target classes $O_n$, which in our case will be the phone-state identities found in the training data.

### 2.4. Summary of the HT-ANN Processing Architecture

A summary of the overall processing architecture is shown in Fig. 2, where we refer to the overall system as HT-ANN. Here it can be seen that the required steps are to first learn both a codebook of feature information, and the voting distribution relative to the observations of the target classes in the training data. Note that we use a single target location for each phone-state occurrence as the reference point for the distribution information, with the reference point placed above the central frame of the state duration. The GHT is then used to compute the activations $A_n(x)$ at every time frame location $x$, by adding up the votes according to the equation (7) for each matching cluster $C_i$ for the feature $f_j$ found at location $l_j$. This activation vector forms the input to the ANN, which performs discriminative training to optimise the classification performance on the training set.

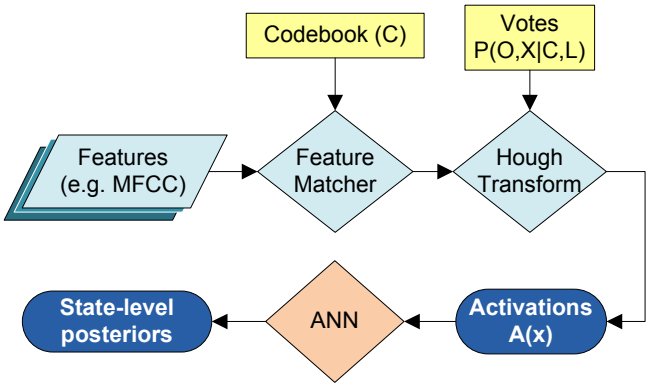



**Fig. 2**. Overview of the processing pipeline for the proposed HT-ANN hybrid architecture.

It can be seen that the activation vector can be considered as a novel feature for hybrid ASR, since the output of the ANN are conventional state-level phone posteriors, which be used in the subsequent processing. It is also important to note that the emphasis of this hybrid architecture is less focussed on the learning ability of the ANN, which has previously been the goal of DNN hybrid systems [9, 10]. This is because the GHT forms a first layer of classification, and the task of the ANN is simply to learn the optimal mapping of the activation vectors to the output labels, which should largely consist of separating the most easily confused classes. This is opposed to DNNs, which are given the more challenging task of performing a multilayer abstraction of the MFCC features, which is necessary to achieve the output classification.

## 3. EXPERIMENTAL EVALUATION

### 3.1. Experimental Setup

Phone recognition experiments are performed on the TIMIT corpus. The standard training set consisting of 462 speakers is used for training with all SA sentences removed. The standard development set of 50 speakers is used for model tuning. Results are reported using the standard 24-speaker core test set consisting of 192 sentences with 7,333 phone tokens. The speech is represented using the 1st-12th-order MFCCs and energy, along with their first and second temporal derivatives, normalised to have zero mean and unit variance, extracted using HTK [11].

We choose frame-level phone-state classification error rate as the main evaluation criterion as this is commonly used in previous works [9, 10, 12, 13]. We also show frame-level phone classification error rates, when the errors in the state within the same phone are not counted, for both the full 61 phone classes and the folded set of 39 phone classes.

| Method | Hidden Layers / Hidden Units | Frame-level State Err (%) (183 classes) |
|---|---|---|
| SVM ([12]) | - | 60.3 |
| OMP ([12]) | - | 48.9 |
| DCN ([13]) | 6 / 7000 | 44.04 |
| DSN ([9]) | 8 / 6000 | 43.86 |
| K-DCNRF ([10]) | 4 / 44000 | 42.87 |
| HT-ANN (proposed) | 1 / 3000 | 39.98 |

**Table 1**. Frame-level classification error rates of the 183 phoneme states on the TIMIT core test set.

### 3.2. System Architecture

The HT-ANN framework described in Section 2 is implemented in Matlab for this preliminary evaluation. This system is first bootstrapped by performing conventional GMM-HMM training in HTK [11] using a 3-state monophone model with 10 mixtures per state. This is used to generate the state labels for the ANN target classes, which are obtained by HMM forced alignment. This gives us a total of 183 state classes, 3 for each of the 61 phone labels defined in the TIMIT training set.

The codebook for the GHT is also directly taken from the GMM-HMM model, using the 183 GMMs learnt for the phone states. The input activation vector for the ANN is therefore a sparse vector of dimension $183 \times 183 = 33489$, since there are 183 entries in the codebook and 183 target classes. To overcome the memory limitation problem, we first perform PCA on a small sample of training data to reduce the number of dimensions down to 1500. A mini-batch size of 256 is used to train the ANN, which has a single hidden layer with a sigmoid non-linearity, to simplify the training and to overcome the difficulty of training DNNs. For the spatial distribution of the features used in the GHT, the temporal extent is limited to a context window of 21 frames, such that the temporal information is relevant to the target state without spreading too far into the neighbouring phones.

| Hidden Units | Frame-level Test Phone Err % (39 classes) | Frame-level Test Phone Err % (61 classes) | Frame-level Test State Err % (183 classes) |
|---|---|---|---|
| 1000 | 26.79 | 32.54 | 41.39 |
| 2000 | 26.05 | 31.77 | 40.29 |
| 3000 | 25.99 | 31.59 | 39.98 |

**Table 2**. Frame-level classification error rates of phones (61 or folded 39 classes), and of phone states (183 classes) as a function of hidden layer size.

### 3.3. Results and Discussion

The frame-level state classification error results are shown in Table 1 and compared with recent techniques using both DNNs [13, 9, 10] and other techniques [12]. It can be seen that the proposed HT-ANN approach achieves a state-of-the-art performance, with the lowest overall classification error. This is despite the simplicity of the neural network used in comparison with the previous experiments on DNNs, that have many times the number of parameters. We also fold the 61 classes in the original TIMIT label set into the standard 39 classes. The corresponding results are presented in Table 2. Note that the results are obtained without the use of phone-bound state alignment and without any phone-level "language" model.

A further experiment is carried out where we expand the front-end feature representation by extracting formant-based features using the Mustafa-Bruce formant tracker approach presented in [14]. An additional 50 codebook clusters are added to separately model the formant information, and only voiced frames are allowed to cast votes during testing, with the remaining frames treated as missing. The results demonstrated an improvement in phone-state test error to 39.66%, which gave a 31.27% and 25.89% test phone error for 61 and 39 classes respectively. This highlights the flexibility of the HT-ANN framework, and further demonstrates the state-of-the-art performance that can be achieved.

## 4. CONCLUSION

This paper described the novel HT-ANN approach for hybrid speech recognition, which uses the GHT to generate codebook activation vectors that can replace MFCCs at the input layer of the ANN. The idea is that the GHT learns the distribution of the observed codebook-feature matches, relative to target class occurrences in the training data, and uses it as a voting function to accumulate evidence. Instead of applying a simple weighted summation of the codebook activations, the HT-ANN approach learns a discriminative weighting that optimises the classification accuracy. Therefore, the codebook activation vector can be considered as a novel feature for hybrid ASR, but has the advantage that the ANN only has to learn a simple weighting to optimise the GHT classification score, as opposed to traditional hybrid architectures, where a DNN is required to obtain different feature abstractions at each layer.

Future work will include a full exploration of the rich flexibility provided by the GHT to model different feature representations. This can include the use of local features, which are common in object detection, that may be able to capture robust information that can perform well in challenging conditions, such as noisy or distant speech. We will also evaluate the system using a dynamic programming-based decoder, to evaluate continuous phonetic or speech recognition tasks.

## 5. REFERENCES


[1] F Seide, G Li, and D Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2011, pp. 437–440.

[2] GE Dahl, D Yu, L Deng, and A Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[3] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] Jonathan Dennis, Huy Dat Tran, and Eng Siong Chng, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1085–1093, July 2013.

[5] Bastian Leibe, Aleš Leonardis, and Bernt Schiele, "Robust Object Detection with Interleaved Categorization and Segmentation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, Nov. 2008.

[6] Subhransu Maji and Jitendra Malik, "Object detection using a max-margin Hough transform," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2009, pp. 1038–1045, IEEE.

[7] R.O. Duda and P.E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[8] D H Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[9] L Deng, D Yu, and J Platt, "Scalable stacking and learning for building deep architectures," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012, pp. 2133–2136.

[10] Po-Sen Huang, Li Deng, Mark Hasegawa-Johnson, and Xiaodong He, "Random features for kernel deep convex network," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 3143–3147.

[11] Steve Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Valtcho Valtchev, and Phil Woodland, *The HTK book, version 3.4*, vol. 3, 2006.

[12] O Vinyals and L Deng, "Are sparse representations rich enough for acoustic modeling?," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2012.

[13] L Deng and D Yu, "Deep convex net: A scalable architecture for speech pattern classification," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2011.

[14] K Mustafa and IC Bruce, "Robust formant tracking for continuous speech with speaker variability," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 435–444, 2006.