A LEVEL-MAP APPROACH TO TRANSFORM COEFFICIENT CODING

Jingning Han, Ching-Han Chiang, and Yaowu Xu

WebM Codec Team, Google Inc.
1600 Amphitheatre Parkway, Mountain View, CA 94043
Emails: {jingning, angiebird, yaowu}@google.com

ABSTRACT

Transform coding is widely used in the video and image codec to largely remove the spatial correlation. The magnitude of transform coefficient is weakly correlated to a number of factors, including its frequency band, the neighboring coefficient magnitudes, luma/chroma planes, etc. To exploit such correlations for efficient entropy coding, one would build a probability model conditioned on the available contexts. However, the interaction of these factors creates a high dimensional space, a direct use of which would easily fall into the over-fitting problem. How to construct a compact context set which effectively captures the underlying correlations remains a major challenge in video and image compression. Prior research work primarily relies on bucketizing the previously coded coefficients into a small number of categories as the context model for next coefficient. Certain information loss is inevitable due to the classification process. To fully exploit the available context in a limited model space, a level map approach is proposed in this work. It decomposes the coding of coefficient magnitudes into consecutive runs of binary map coding, each corresponds to whether a coefficient is equal to or greater than the given level. Under the Markov assumption, nearly all the reference symbols available to each level map can be approximated as binary random variables. It hence allows the context model to account for all the surrounding coefficients information provided by the lower level maps, while retaining a compact size. Experimental evidence demonstrates that the proposed coding scheme provides considerable compression performance gains consistently over a large test settings.

Index Terms— Transform coding, probability context modeling, level map, adaptive coefficient scan

1. INTRODUCTION

Transform coding is commonly employed in video and image compression system to decorrelate the spatial redundancy for efficient quantization and entropy coding. The discrete cosine transform (DCT) has long been used as a low complexity surrogate of the Karhunen-Love transform. Recent research demonstrates that a class of asymmetric discrete sine transforms (ADST) closely approximate the optimal transform for certain prediction residuals [1]. Late standardization efforts [2, 3] also incorporate variable transform block sizes to better capture the variations in signal statistics. The interaction of multiple transform kernels and sizes provides a rather flexible framework to represent the signals in a compact form.

While the coefficient signs are considered largely independent within a transform block, the magnitudes exhibit weak correlations to a number of factors, including frequency band, transform size, neighboring coefficient magnitudes, etc. To fully capture these underlying correlations, one would need a multi-dimensional context probability model, which incurs a giant condition set and makes it hard to get statistically sound results for a general purpose codec. A common practice is to bucketize the conditions into a small set of categories, and use which to build context model. In VP9 [2], the frequency index is classified into 5 clusters and the previously coded coefficient magnitude is quantized into 6 levels. The entropy coder processes the coefficients in a 2-D array sequentially following a transform kernel dependent scan order as shown in [1]. The probability model for a given coefficient coding is conditioned on its nearest above and left neighbors' magnitude levels, its frequency band, as well as the transform size and luma/chroma plane. A reverse processing order is employed in the H.264/AVC [4] and HEVC codec [5, 6], where a non-zero coefficient map is first coded and the magnitudes are processed from the highest frequency backward towards lowest frequency position. The probability model relies on the bottom and right neighbors magnitude information, including cumulative numbers of level one and two in the tailing coefficients. Such categorization process would naturally lose certain context information and might cause sub-optimal compression performance.

A level-map coefficient coding scheme is proposed in this work to optimize the trade off between context modeling efficiency and the model size. Unlike the prior coding engines that run through a 2-D transform coefficient array and process each coefficient value sequentially, the proposed system employs a multi-run level-map approach that breaks down the coding of coefficient value into a series of binary decisions, each corresponds to a magnitude level. For instance, a binary decision of coefficient (r, c) at level k is defined as:

$$level_k[r][c] = 1 \text{ if } abs(coeff[r][c]) > k, \quad (1)$$

$$= 0 \text{ if } abs(coeff[r][c]) \leq k. \quad (2)$$

All the binaries at the same level across the 2-D transform coefficient array form a level-map. A transform coefficient value can be hence decomposed into a series of level binaries and a residue, in addition to a sign value for a non-zero magnitude:

$$coeff[r][c] = \{(\sum_{k=0}^{T} level_k[r][c]) + residue[r][c]\} \times sign[r][c], \quad (3)$$

where

$$residue[r][c] = abs(coeff[r][c]) - T \quad (4)$$

$$sign[r][c] = 1 \text{ if } coeff[r][c] > 0 \quad (5)$$

$$= -1 \text{ if } coeff[r][c] < 0 \quad (6)$$

The proposed scheme codes the level-maps sequentially in the ascending order from $k = 0$ to a maximum level denoted by T. If the transform block contains coefficient value above T, the last round coding will process the residues per coefficient.

When coding the level-k map, the fully coded level-$(k-1)$ map and partially coded level-k map are used as context information for probability modeling. Under the Markov assumption, all the maps below level-$(k-1)$ can be discarded. As compared to the prior transform coefficient coding system that processes one coefficient value at a time before moving on to the next one, the proposed scheme provides the advantage of reducing the cardinality of the reference sample set – all the information from level-$(k-1)$ map and partially coded level-k map is in binary form. It allows the room to create more sophisticated spatial neighboring context for probability modeling, while retaining a relatively compact model size.

From the system complexity perspective, only those positions (r, c) where $level_{k-1}[r][c] = 1$ need to be processed in level-k map coding. Statistically this would substantially reduce the amount of binary coding operations, since the majority of the quantized transform coefficients would be of small magnitudes.

The level-map coefficient coding scheme is composed of four stages: (1) non-zero map and end-of-block (EOB) map, which indicates whether a position is the last non-zero coefficient; (2) multiple runs of level-maps of non-zero transform block coefficients; (3) the residues of remaining coefficients; and (4) sign map for non-zero coefficient. We will discuss with more details in next sections. The proposed scheme is implemented in the AV1 codec, a successor of the VP9 codec jointly developed by the Alliance of Open Media [7]. It is experimentally shown to provide considerable compression performance in a wide range of test sets.

2. NON-ZERO MAP AND EOB MAP

A non-zero map, $nz\_map[r][c]$ can be considered as level-0 map. Unlike other level maps, it does not have lower level reference. The coding process starts from lowest frequency following the transform dependent scan order. An adaptive scan order approach that tailors to frame level statistics is proposed in [8], which can potentially be integrated in the non-zero map coding, although it is outside the scope of this paper.

An EOB map, $eob\_map[r][c]$, is used to indicate whether a non-zero coefficient is the last non-zero coefficient in the transform block, with respect to the given scan order. The non-zero map, $nz\_map[r][c]$, and the EOB map, $eob\_map[r][c]$, are interleaved and coded. At position (r, c), a $nz\_map[r][c]$ is coded first. If $nz\_map[r][c]$ is 1, a $eob\_map[r][c]$ is coded next to indicate if the position (r, c) is the last non-zero in the scan pattern. This coding process finishes when $eob\_map[r][c]$ equals to 1, or the last position in the transform block is reached.

The context information is formed as the sum of nz_map values of previously coded neighbors in the reference region (Fig. 1):

$$nz\_map\_nb\_sum(r, c) = \sum_{(\bar{r},\bar{c})\in \text{neighbor}(r,c)} nz\_map(\bar{r}, \bar{c}).$$

The positions in the transform block are further separated into four groups according to its frequency location: (1) $r = 0$ and $c = 0$; (2) $r = 0$ and $c > 0$; (3) $r > 0$ and $c = 0$; and (4) $r > 0$ and $c > 0$. Every $nz\_map\_nb\_sum$ value in each position group is associated with a unique context index.

A frequency band map is used as context to model $eob\_map[r][c]$. The general design principle is to allow higher resolution at low frequency end. An example of such partition in a transform block is shown in Fig. 2.

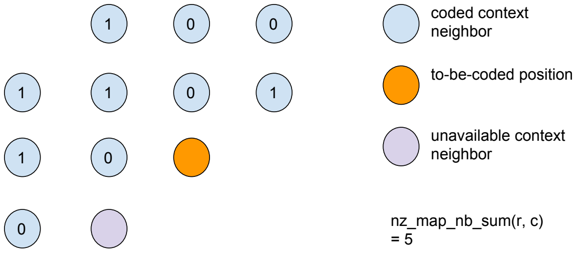

Fig. 1. The reference region of the nz_map coding.

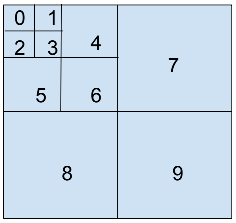

Fig. 2. Frequency band map for coeff_map context model.

3. LEVEL-MAP CODING

By setting a lower-range integer threshold T, the transform block coefficients can be decomposed into T level-maps and larger-than-T coefficient residues. This threshold can be set statically or adaptively according to transform block statistics. The level maps are coded sequentially from lower level to higher level. Each level-map is coded in the reverse scan order. An example of level-map coding scheme with $T = 2$ is provided in Fig. 3. At each level, only the previously non-zero positions (highlighted) need to be coded, others can be inferred from previous coded information. To construct context information for level-k map at position (r, c), the position (r, c)'s neighbor sum in level-$(k-1)$ map is calculated first. It then checks if the previously coded nearest right and bottom neighbors have any value 1 in level-k map. These two factors in conjunction with its frequency band, which is separated into four groups: (1) $r = 0$ and $c = 0$, (2) $r = 0$ and $c > 0$, (3) $r > 0$ and $c = 0$, and (4) $r > 0$ and $c > 0$, form the context information.

There exist multiple ways to define spatial reference region for level-k map coding. A square neighborhood that includes 8 nearest coefficient positions is shown in Fig.4. The previously coded positions typically contain four coefficients to the right or bottom of the current position. The diamond shape reference region is formed as a 3×3 centered at the current position with additional 4 positions: $(2, 0)$, $(-2, 0)$, $(0, 2)$, $(0, -2)$ as in Fig.5. These 4 additional positions are added because our analysis of the quantized transform coefficients shows that they possess certain correlations with the current position.

4. HIGHER-RANGE COEFFICIENT RESIDUES

For coding the remaining higher-range coefficient residues, geometric distribution or Pareto distribution may be used to describe the residue statistics. In our implementation, the geometric distribu-

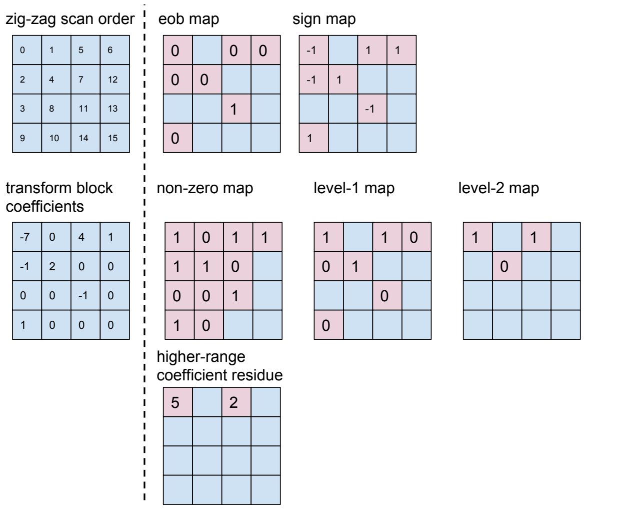

Fig. 3. An example of a two-level map coding pipeline with a zig-zag scan order.

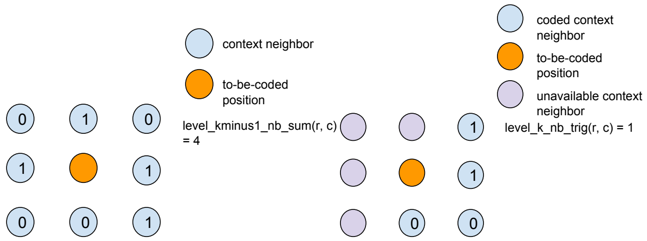

Fig. 4. Square region: the spatial reference pattern from level-$(k-1)$ and the previously coded reference symbols in level-k.

tion is used conditioned on its immediate right and bottom neighbors magnitude category level. Note that the tailing distribution of the actual data may slightly divert from that of the geometric distribution. However, such divergence appears not to affect the overall coding performance significantly, due to its low volume of appearance.

5. COEFFICIENT SIGN MAP

Every non-zero coefficient is assigned with a sign value. To exploit the correlations across nearby transform blocks, a context modeling approach is used to code the signal value of DC coefficient, dc_sign. It accounts for the above and left neighbor blocks' dc_sign value, all weighted by the length of their intersection with current transform block, i.e.,

$$dc\_sum = \sum_{i \in \text{neighbor\_blocks}} dc\_sign(i) \times overlap(i, curr\_block). \quad (7)$$

The context index for dc_sign is thus designed as:

$$\begin{aligned} dc\_ctx &= 0 \text{ if } dc\_sum = 0, && (8)\\ &= 1 \text{ if } dc\_sum < 0, && (9)\\ &= 2 \text{ if } dc\_sum > 0. && (10)\end{aligned}$$

All the rest AC coefficients are coded as a single bit.

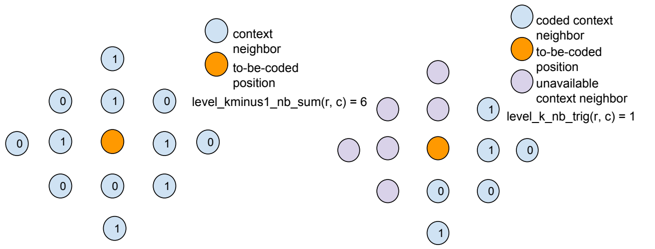

Fig. 5. Diamon shape region: the spatial reference pattern from level-$(k-1)$ and the previously coded reference symbols in level-k.

6. EXPERIMENTAL RESULTS

The proposed scheme is implemented in the AV1 codec, where the baseline supports multiple transform kernels selected at coding block level for both intra and inter prediction modes, in addition to variable transform block sizes. The baseline also employs transform dependent scan order as discussed in [1, 2]. The current implementation inherits same scan approach as the baseline. The diamond shape reference region as shown in Fig.5 is chosen for the level map coding. The maximum level T is set to be 3.

We evaluate the proposed transform coefficient engine on a large test set and over a wide range of bit-rates. The coding performance can be found in Table.1-3. Evidently it consistently provides considerable compression performance gains in all test settings.

7. CONCLUSIONS

A level map based transform coefficient coding system is proposed in this work. It decomposes the integer magnitude coding into a series of binary coding. Under certain Markov assumption, each stage is able to fully exploit the available information for context modeling, while retaining a rather compact model size. Integrated in the AV1 framework, it is experimentally shown to provide consistent coding performance gains.

Table 1. Coding performance on low-resolution dataset in terms of BD rate reduction. The low, mid, and high tabs represent the bit-rate range break downs, each contains 4 - 5 operating points.

	res	avg (%)	low	mid	high
akiyo	CIF	-0.974	-1.136	-0.709	-1.178
basketballpass	240p	-0.913	-0.683	-1.001	-1.459
blowingbubbles	240p	-1.618	-1.638	-1.689	-1.818
bowing	CIF	-1.953	-2.193	-1.729	-1.883
bqsquare	240p	-0.874	-0.955	-0.742	-0.956
bridge_close	CIF	-2.913	-2.493	-3.461	-3.025
bridge_far	CIF	-4.542	-4.224	-1.735	-9.538
bus	CIF	-2.029	-1.865	-2.327	-2.124
cheer	SIF	-0.762	-0.792	-0.754	-0.788
city	CIF	-1.359	-1.349	-1.454	-1.436
coastguard	CIF	-3.648	-4.091	-3.571	-3.332
container	CIF	-2.356	-2.723	-1.925	-1.580
crew	CIF	-1.797	-1.938	-1.872	-1.802
deadline	CIF	-1.530	-1.526	-1.192	-1.113
flower	CIF	-2.236	-2.393	-2.353	-1.862

flowervase	240p	-2.260	-2.601	-1.847	-1.473
football	CIF	-1.429	-0.960	-1.820	-2.117
foreman	CIF	-0.948	-0.784	-0.864	-1.554
garden	SIF	-1.733	-1.404	-2.090	-1.824
hallmonitor	CIF	-1.924	-1.261	-2.466	-2.270
harbour	CIF	-2.557	-2.437	-2.725	-2.829
highway	CIF	-4.206	-2.821	-5.525	-4.534
husky	CIF	-2.233	-1.722	-2.481	-2.571
ice	CIF	-1.487	-1.494	-1.625	-1.420
keiba	240p	-1.665	-1.714	-1.710	-1.766
mobile	CIF	-1.352	-1.033	-1.656	-1.888
mobisode2	240p	-2.664	-3.026	-1.708	-1.505
motherdaughter	CIF	-1.654	-1.835	-1.138	-1.558
news	CIF	-0.598	-0.734	-0.473	-0.706
pamphlet	CIF	-1.167	-1.812	-0.189	-1.366
paris	CIF	-1.783	-2.524	-1.150	-0.914
racehorses	240p	-0.855	-0.611	-1.044	-1.229
signirene	CIF	-1.157	-1.143	-1.130	-1.128
silent	CIF	-1.203	-1.233	-1.214	-1.664
soccer	CIF	-1.804	-1.935	-1.723	-1.789
stefan	SIF	-2.338	-2.363	-2.654	-2.008
students	CIF	-1.103	-1.231	-0.976	-0.826
tempete	CIF	-1.201	-1.034	-1.356	-1.448
tennis	SIF	-1.053	-0.818	-1.298	-1.462
waterfall	CIF	-0.676	-0.601	-0.199	-1.457
OVERALL		-1.764	-1.728	-1.689	-1.930

Table 2. Coding performance on mid-resolution dataset in terms of BD rate reduction. The low, mid, and high tabs represent the bit-rate range break downs, each contains 4 - 5 operating points.

	res	avg(%)	low	mid	high
aspen	480p	-0.437	-0.558	-0.518	-0.654
BasketballDrill	480p	-1.260	-1.373	-0.933	-1.277
BDrillText	480p	-1.275	-1.388	-1.174	-1.265
BQMall	480p	-1.675	-1.533	-1.949	-1.994
city	4CIF	-1.918	-1.496	-2.808	-2.763
controlled_burn	480p	-2.715	-3.417	-2.158	-1.261
crew	4CIF	-1.744	-2.112	-1.567	-1.789
crowd_run	480p	-0.771	-0.070	-0.922	-1.372
ducks_take_off	480p	-4.462	-5.779	-4.427	-4.087
Flowervase	480p	-3.123	-3.397	-3.480	-2.261
harbour	4CIF	-2.622	-3.010	-2.589	-2.670
ice	4CIF	-0.527	-0.036	-1.273	-1.327
into_tree	480p	-0.777	-0.796	-0.603	-1.105
Keiba	480p	-2.140	-2.044	-2.506	-2.242
Mobisode2	480p	-2.361	-2.757	-2.286	-2.154
old_town_cross	480p	-0.992	-0.552	-1.796	-1.163
park_joy	480p	-0.867	-0.731	-0.788	-1.277
PartyScene	480p	-1.424	-1.576	-1.294	-1.675
RaceHorses	480p	-0.658	-0.192	-0.574	-1.075
red_kayak	480p	-1.809	-1.940	-1.606	-2.285
rush_field_cuts	480p	-0.763	-0.972	-0.855	-1.056
sintel_trailer_2k	480p	-1.700	-1.927	-1.386	-1.250
snow_mnt	480p	-1.274	-1.786	-0.680	-0.931
soccer	4CIF	-2.418	-1.614	-3.181	-3.331
speed_bag	480p	-1.558	-1.672	-1.420	-1.489
station2	480p	-0.355	-0.539	-0.199	-0.141
tears_of_steel1	480p	-2.238	-2.372	-2.299	-1.985
tears_of_steel2	480p	-2.160	-2.316	-2.068	-1.881
touchdown_pass	480p	-1.205	-1.078	-1.361	-1.567
west_wind_easy	480p	-0.462	0.215	-0.445	-1.392
OVERALL		-1.590	-1.627	-1.638	-1.691

Table 3. Coding performance on high-resolution dataset in terms of BD rate reduction. The low, mid, and high tabs represent the bit-rate range break downs, each contains 4 - 5 operating points.

	res	avg(%)	low	mid	high
basketballdrive	1080p	-2.121	-2.277	-2.441	-2.029
blue_sky	1080p	-0.816	-0.525	-1.254	-1.737
bqterrace	1080p	-1.890	-1.819	-1.992	-2.216
cactus	1080p	-1.149	-1.157	-1.390	-1.686
chinaspeed	XGA	-0.977	-1.053	-0.916	-1.490
city	720p	-1.710	-1.733	-2.046	-1.936
crew	720p	-1.842	-2.056	-1.997	-1.522
crowd_run	1080p	-0.723	-0.667	-0.691	-0.838
cyclists	720p	-1.077	-1.089	-1.310	-1.264
dinner	1080p	-1.879	-1.924	-1.872	-2.394
ducks_take_off	1080p	-3.562	-4.155	-3.549	-3.177
factory	1080p	-1.384	-1.681	-1.302	-1.907
fourpeople	720p	-1.471	-1.330	-1.784	-2.056
in_to_tree	1080p	-0.736	0.607	-1.355	-1.000
jets	720p	-2.591	-3.144	-2.898	-1.404
johnny	720p	-2.581	-2.426	-3.081	-2.557
kimono1	1080p	-1.174	-1.051	-1.460	-2.078
kristenandsara	720p	-2.339	-2.457	-2.710	-2.681
life	1080p	-1.316	-1.289	-1.435	-1.321
mobcal	720p	-1.704	-1.620	-2.400	-1.040
night	720p	-0.908	-0.877	-1.064	-0.921
old_town_cross	720p	-0.857	-0.715	-1.031	-1.584
parkjoy	1080p	-1.085	-0.780	-1.274	-1.507
parkrun	720p	-1.836	-1.717	-1.903	-2.351
parkscene	1080p	-0.711	-0.621	-0.825	-1.151
ped	1080p	-0.877	-0.829	-0.759	-1.241
riverbed	1080p	-2.213	-1.890	-2.933	-2.358
rush_hour	1080p	-0.982	-0.639	-1.372	-1.831
sheriff	720p	-2.034	-1.844	-2.223	-1.877
shields	720p	-1.328	-0.860	-1.603	-1.385
station2	1080p	-0.999	-0.453	-1.481	-2.085
stockholm_ter	720p	-1.246	-1.848	-0.642	-1.410
sunflower	720p	-0.582	-0.464	-1.178	-1.475
tennis	1080p	-1.661	-1.600	-2.148	-2.152
tractor	1080p	-1.291	-1.015	-1.630	-2.022
vidyo1	720p	-1.690	-2.421	-1.456	-1.189
vidyo3	720p	-2.687	-3.672	-2.315	-0.974
vidyo4	720p	-1.470	-1.307	-1.903	-1.247
OVERALL		-1.513	-1.484	-1.727	-1.713

8. REFERENCES

[1] Jingning Han, Ankur Saxena, Vinay Melkote, and Kenneth Rose, “Jointly optimized spatial prediction and block transform for video and image coding,” *IEEE Transactions on Image Processing*, vol. 21, pp. 1874–1884, 2012.

[2] D. Mukherjee, J. Han, J. Bankoski, R. Bultje, A. Grange, J. Koleszar, P. Wilkins, and Y. Xu, “A technical overview of vp9 - the latest open-source video codec,” *SMPTE*, vol. 2013, no. 10, pp. 1–17, 2013.

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding HEVC standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[4] Detlev Marp, Heiko Schwarz, and Thomas Wiegand, “Context-based adaptive binary arithmetic coding in the H.264AVC video compression standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, 2003.

[5] Joel Sole, Rajan Joshi, Nguyen Nguyen, Tianying Ji, Marta Karczewicz, Gordon Clare, Felix Henry, and Alberto Duenas, “Transform coefficient coding in HEVC,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1765–1777, 2012.

[6] Tung Nguyen, Heiko Schwarz, Heiner Kirchhoffer, Detlev Marpe, and Thomas Wiegand, “Improved context modeling for coding quantized transform coefficients in video compression,” *Picture Coding Symposium*, pp. 378–381, 2010.

[7] “Alliance of open media - source code repository,” *https://aomedia.googlesource.com/*.

[8] Ching-Han Chiang, Jingning Han, and Yaowu Xu, “A constrained adaptive scan order approach to transform coefficient entropy coding,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017.