



Communications
Research Centre
Canada

An Agency of
Industry Canada

Centre de recherches
sur les communications
Canada

Un organisme
d'Industrie Canada

SPARSE AUDIO CODING VIA TARGETED DITHERING AND COMBINATORIAL DECODING

*F. Mustiere^{1,2}, H. Najaf-Zadeh¹, R. Pichevar¹,
H. Lahdili¹, L. Thibault¹, M. Bouchard²*

¹Communications Research Centre, Advanced Audio Systems, Ottawa, Canada

²School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada

- Introduction: Transform domain coding
- Proposed solution overview
- Encoder side: Subframe decomposition
- Encoder side: Targeted dithering
- Encoder side: Choice of filler vectors
- Encoder side: Header information
- Decoder side: Combinatorial decoding
- Practical results and future work

- Introduction: Transform domain coding
- Proposed solution overview
- Encoder side: Subframe decomposition
- Encoder side: Targeted dithering
- Encoder side: Choice of filler vectors
- Encoder side: Header information
- Decoder side: Combinatorial decoding
- Practical results and future work

- Sparse signal coding context
- Transform domain representation of audio
- From input signal \mathbf{n} , obtain $\mathbf{x} = \psi(\mathbf{n})$
- With appropriate transformation, most elements in \mathbf{x} have negligible perceptual value and can be zeroed.

Inherent problem in transform coding

- Transmission of sparse vectors:
 - Quantized active/non-negligible **values**
 - Their **position** in the sparse vector
- **Problem**: potentially large overhead required to encode the positions
- **Some solutions**: RLE, Huffman-coding of position vectors, etc.
- We propose an alternative solution

- Introduction: Transform domain coding
- **Proposed solution overview**
- Encoder side: Subframe decomposition
- Encoder side: Targeted dithering
- Encoder side: Choice of filler vectors
- Encoder side: Header information
- Decoder side: Combinatorial decoding
- Practical results and future work

Proposed solution overview

- **Embed** some position information in quantized values
- **Adjust quantized values** so as to minimize some function of both values and positions
- **Perceptually controlled** adjustment
- Decoder progresses through reduced set of possible positions, and verifies which one minimizes the known function

- Introduction: Transform domain coding
- Proposed solution overview
- **Encoder side: Subframe decomposition**
- Encoder side: Targeted dithering
- Encoder side: Choice of filler vectors
- Encoder side: Header information
- Decoder side: Combinatorial decoding
- Practical results and future work

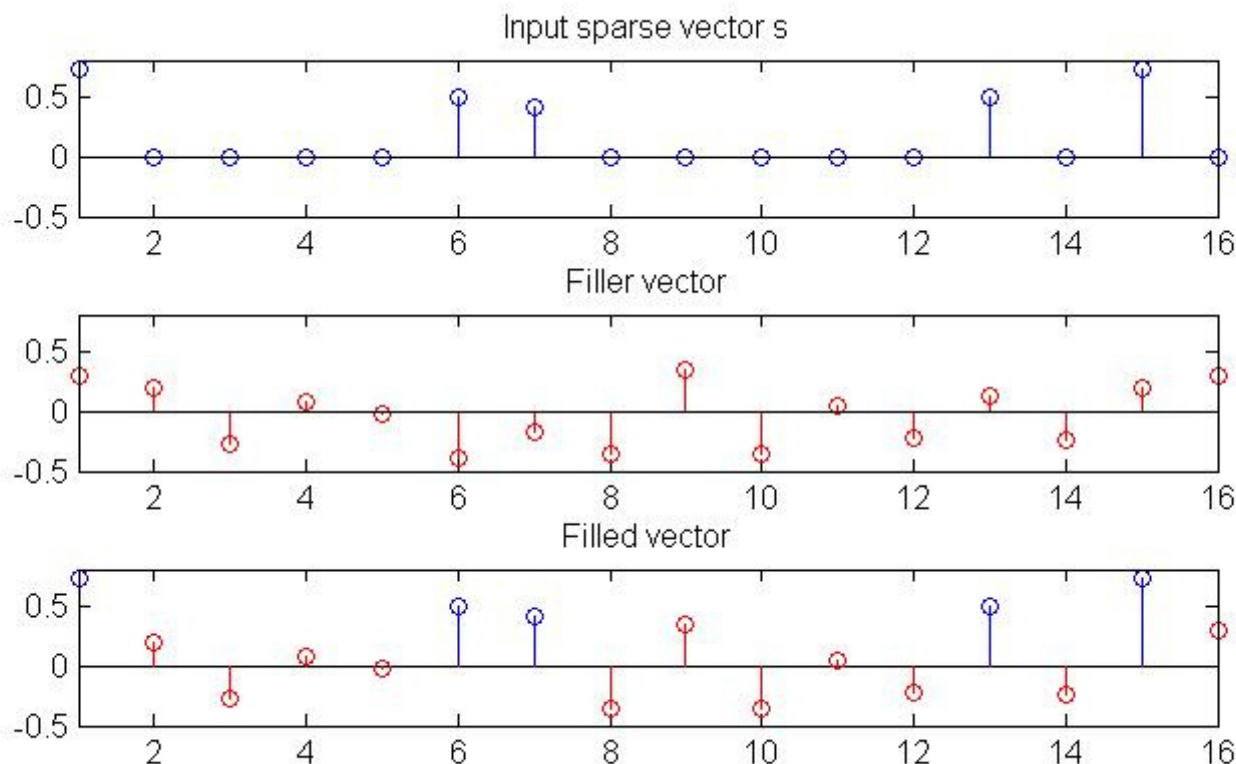
Subframe decomposition

- Further decompose into small non-overlapping frames
- Reduce set of possibilities for decoder
- **Compromise required** (not too small!): each subframe needs a header, and smaller frames mean less room for transparent adjustments!

- Introduction: Transform domain coding
- Proposed solution overview
- Encoder side: Subframe decomposition
- **Encoder side: Targeted dithering**
- Encoder side: Choice of filler vectors
- Encoder side: Header information
- Decoder side: Combinatorial decoding
- Practical results and future work

Targeted dithering stage

- Obtain the quantized sparse input
- Using a fixed, unquantized **filler vector**, define vector \mathbf{v}



\mathbf{v} consists of the red dots on the bottom graph. It essentially completes \mathbf{s}_q with values from the filler vector

Targeted dithering stage

- Denote by T a certain **target number**
- We can then define a **targeting function**:

$$f(\mathbf{s}_q, \mathbf{v}) = T + e$$

designed such that e is small

- Goal: adjust \mathbf{s}_q such that e is minimized

- E.g.:
$$f(\mathbf{s}_q, \mathbf{v}) = \sum_n (\mathbf{s}_q(n) + \mathbf{v}(n))$$
$$T = [f(\mathbf{s}_q, \mathbf{v})]$$

- Why would we do this?
- Suppose that s_q has been adjusted and e is minimized
- The K quantized and adjusted values are sent to the decoder
- With *only* these K values, the decoder can go through the set of possible positions and stop whenever the following is smaller than an established threshold ϵ :

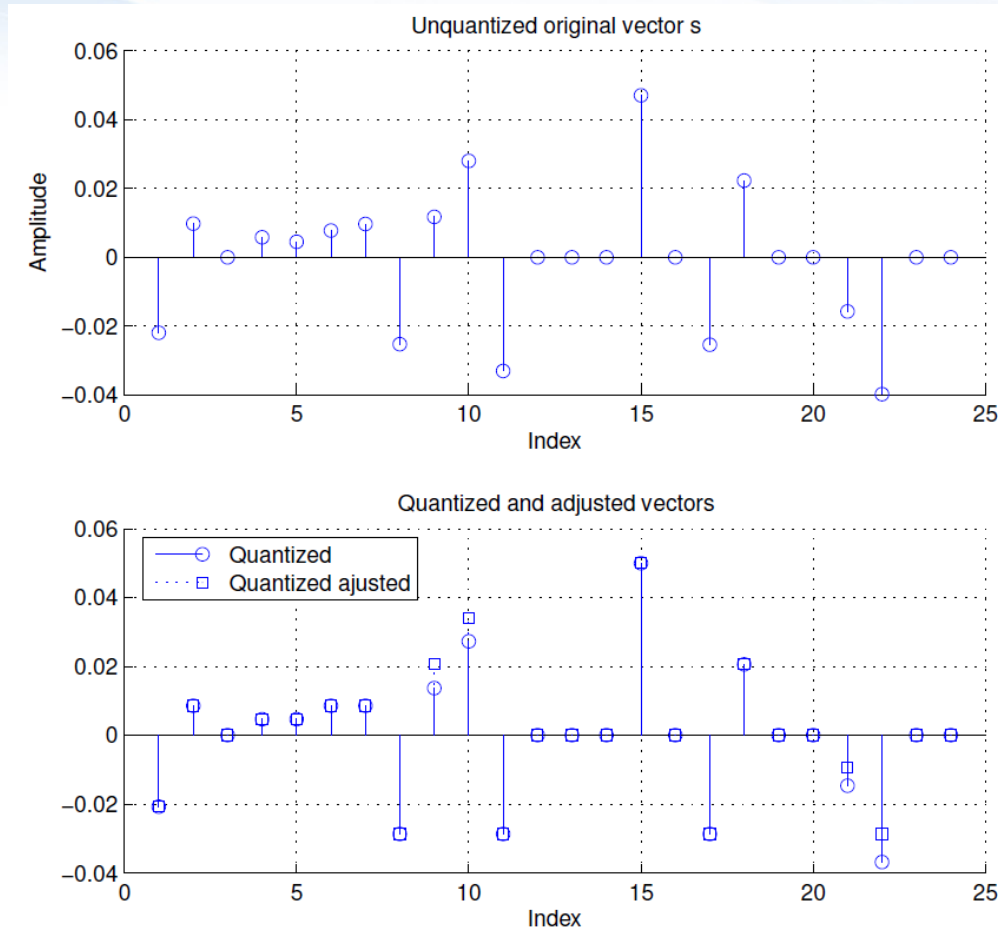
$$e = f(s_q, \mathbf{v}) - [f(s_q, \mathbf{v})] \leq \epsilon$$

Performing the adjustment

- The problem can be cast as a **binary integer program**, i.e., a constrained minimization with binary variables
- **Efficient branch-and-bound** can be used
- The solution **minimizes the required adjustments**
- The problem is **feasible** for at least one choice of **threshold**

Example of targeted dithering

- Example: $\epsilon = 5 \times 10^{-5}, T = 2$



$$f(s_q, \mathbf{v}) \simeq 2.28987$$

$$f(s_q, \mathbf{v}) \simeq 1.99997$$

- Introduction: Transform domain coding
- Proposed solution overview
- Encoder side: Subframe decomposition
- Encoder side: Targeted dithering
- **Encoder side: Choice of filler vectors**
- Encoder side: Header information
- Decoder side: Combinatorial decoding
- Practical results and future work

- Choice of targeting function and filler vector is important
- Need smallest possible epsilon with the smallest amount of changes
- Also, should yield significantly different function values for incorrect positioning
- Open problem, but a few findings in the paper
 - Adaptive filler vectors
 - Zero-mean Gaussian, smaller variance than input signal yields operational solution
 - Bank of filler vectors

- Introduction: Transform domain coding
- Proposed solution overview
- Encoder side: Subframe decomposition
- Encoder side: Targeted dithering
- Encoder side: Choice of filler vectors
- **Encoder side: Header information**
- Decoder side: Combinatorial decoding
- Practical results and future work

- If everything goes “as planned”, all that is needed is the **number of active elements**
- Encoder can easily verify whether everything will go well for the decoder
- If not: header can provide **additional hints**, e.g.:
 - Exponent of the error
 - Target integer
 - Rough amount of trials required (can save time!)
 - Etc.

- Introduction: Transform domain coding
- Proposed solution overview
- Encoder side: Subframe decomposition
- Encoder side: Targeted dithering
- Encoder side: Choice of filler vectors
- Encoder side: Header information
- **Decoder side: Combinatorial decoding**
- Practical results and future work

- Decoder is **very simple in essence**
- For each subframe:
 - Read header to get number of active elements (and any other “hint”)
 - Decode the amplitudes
 - Go through a list of position vector candidates. For each possibility, extract filler vector and test targeting function
- List exhaustion scheme, required amount of trials, and other details derived in the paper

- Introduction: Transform domain coding
- Proposed solution overview
- Encoder side: Subframe decomposition
- Encoder side: Targeted dithering
- Encoder side: Choice of filler vectors
- Encoder side: Header information
- Decoder side: Combinatorial decoding
- **Practical results and future work**

- Paper contains a **detailed operational codec**
- Bit allocation scheme, “tricks” to reduce amount of trials required, and to ensure success of targeted dithering
- Preliminary tests: **good quality for 44.1 kHz** sampling rate at **about 50 kbps**.
- Adjusted vs. unadjusted: example with 6 bits per coefficient and nearly undistinguishable results
- **Drawback**: significantly slower and more demanding than available state-of-the-art audio codecs



Questions?

Frederic Mustiere
fmustier@crc.gc.ca