SIPL SIGNAL & IMAGE PROCESSING LAB

Quality-Preserving Footprint-Reduction of Concatenative Text-To-Speech Synthesizers

Summary of research towards M.Sc. by

Tamar Shoham

Supervisor: Prof. David Malah

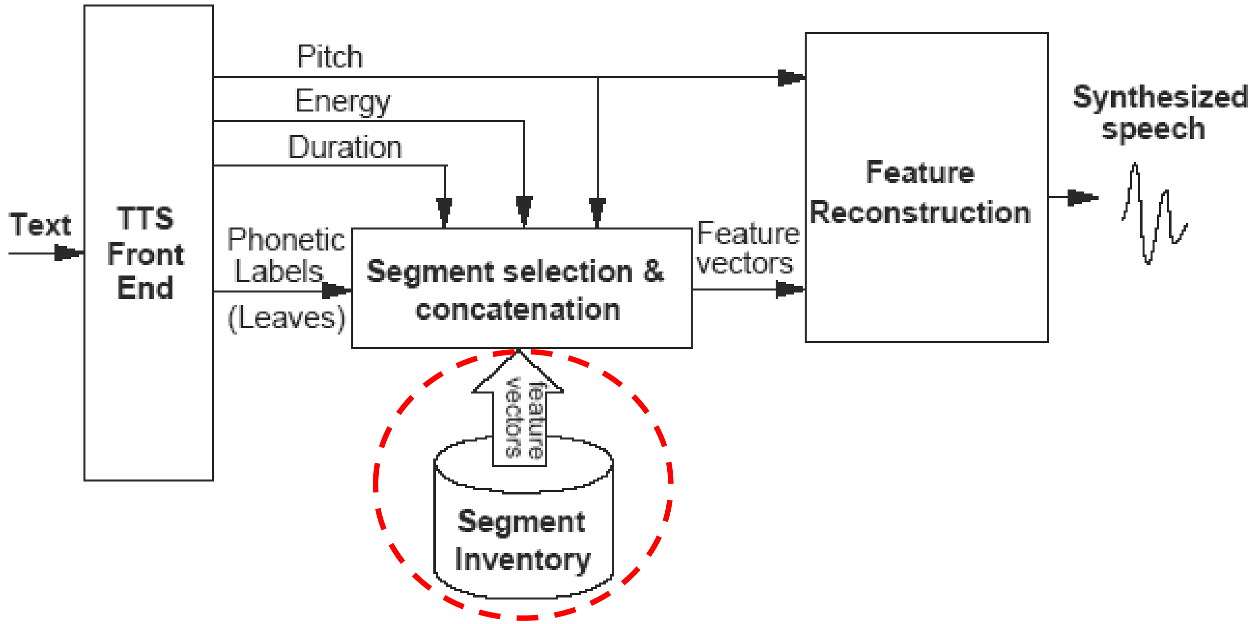
This research was performed in collaboration with IBM Haifa Research Labs

28th April, 2010

Outline

- Introduction to Concatenative Text-To-Speech
- Problem statement
- IBM small footprint CTTS speech compression model
- Prior work
- Proposed compression approaches:
  - Vectorial Polynomial Temporal Decomposition
  - 3D Shape Adaptive DCT
- Segment reordering
- Experimental results
- Conclusion

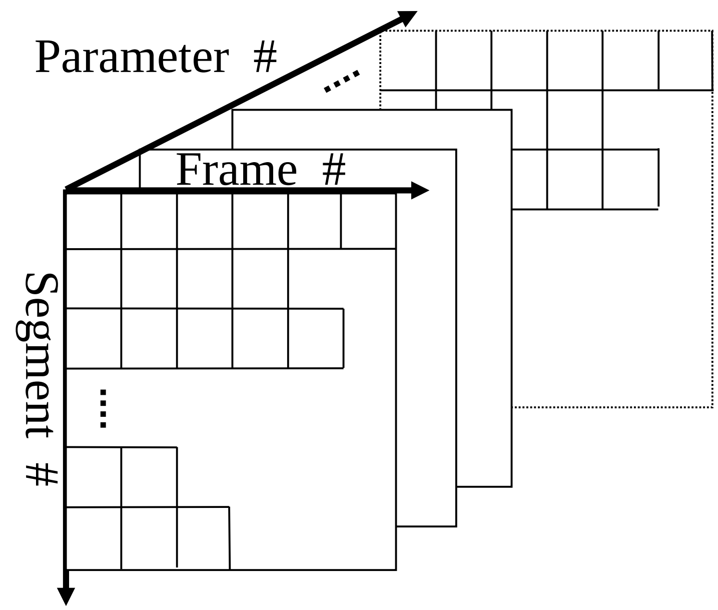
Introduction to CTTS



- Front end: phonetic analysis to define appropriate sub-phonemes, their pitch, energy, duration, and context parameters.
- Segment selection:
  - Choose acoustic leaf according to sub-phoneme and its context.
  - Choose segment with lowest target and concatenation costs.
- Most of the footprint is due to the segment inventory or database.

CTTS database structure

- Database consists of acoustic leaves, each corresponding to a specific sub-phoneme in a specific context.
- A number of speech segments are stored in each acoustic leaf.
- Each speech segment consists of one or more speech frames.
- Each speech frame is represented by a set of parameters, usually using a spectral model.



Problem Statement

- Reduce the footprint of a CTTS synthesizer without compromising obtained perceptual quality.
- Develop a (re) compression algorithm for a set of 3D data structures, containing parameters that exhibit redundancy, such as the acoustic leaves in a CTTS database.
- Algorithm should not be tightly bound to specific database characteristics.
- Additional requirement: Low decoding complexity.

IBM small footprint CTTS: speech model - 1

- Based on polar form of the complex spectral envelope of the speech frame:

$$S(f) = A(f)e^{j\varphi(f)}$$

- We concentrate on amplitude parameters as they account for most of the footprint (5.7MB vs. 1.6MB for phase).
- A warped frequency scale, the Mel-scale, is used:

$$f' = 2595\log_{10}\left(1 + \frac{f}{700}\right)$$

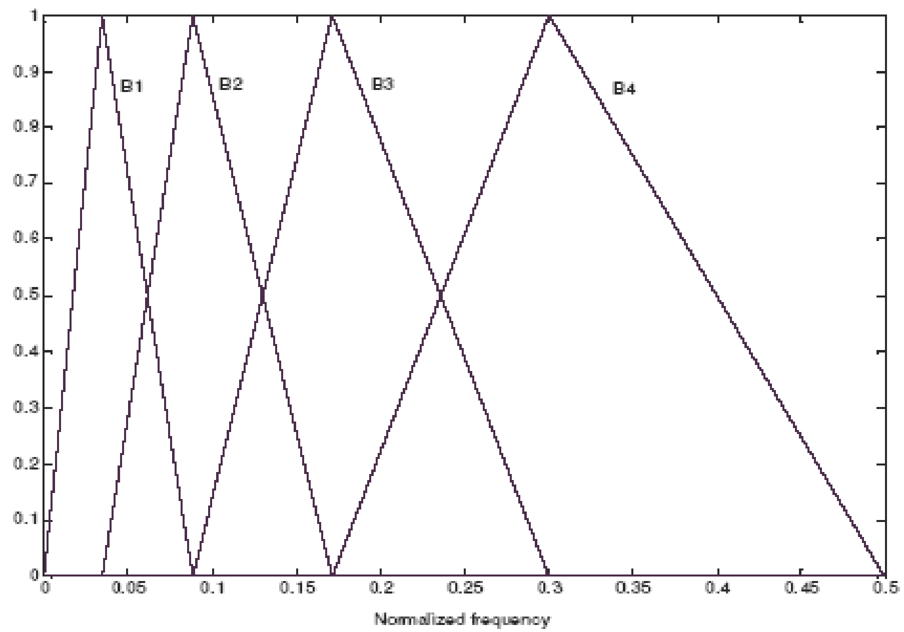
(*) "Small footprint Concatenative text-to-speech synthesis using complex envelope modeling", Chazan *et* al., INTERSPEECH 2005.

IBM small footprint CTTS: speech model – 2

- The log-amplitude spectrum of each frame is modeled by a linear combination of L basis functions:

$$\log(A(f')) = \sum_{n=1}^{L} c_n B_n(f'); \ (L = 32)$$

- ***Bn*** are triangular.
- In Mel-scale they have equal widths and half overlap.



IBM small footprint CTTS: speech model – 3

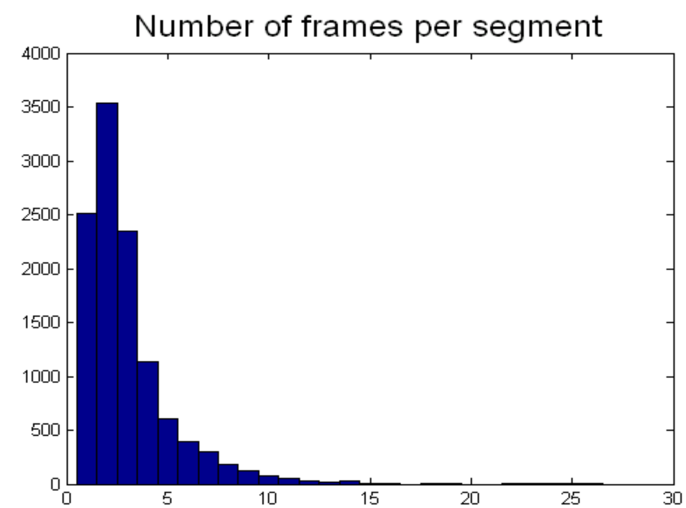
- ***G***, frame energy, is embedded as follows:

$$C_n = c_n - \tfrac{1}{L}\left(\sum_{k=1}^{L} c_k - \log G\right)$$

- $\boldsymbol{C}n$, the representing parameters, are used for segment selection, speech morphing and synthesis.
- The 32 amplitude parameters of each frame are quantized using an 86 bit split-VQ scheme.
- The VQ is applied to the parameter differences, equivalent to the spectral ratios. The quantizer favors the low frequency data.

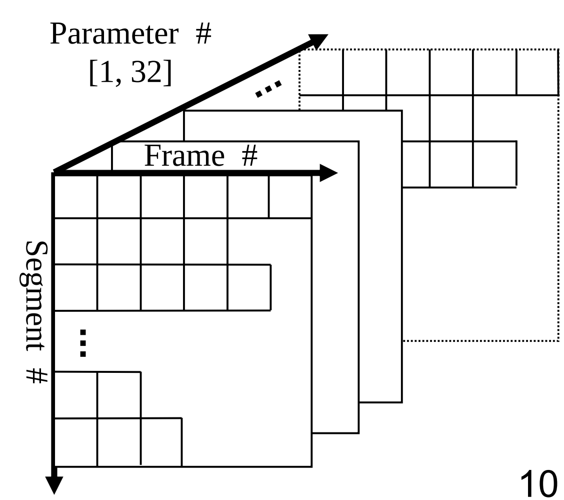
Acoustic leaf (re) compression

- We wish to remove inter-frame redundancies.
- Distribution of segment lengths in a sample database --->



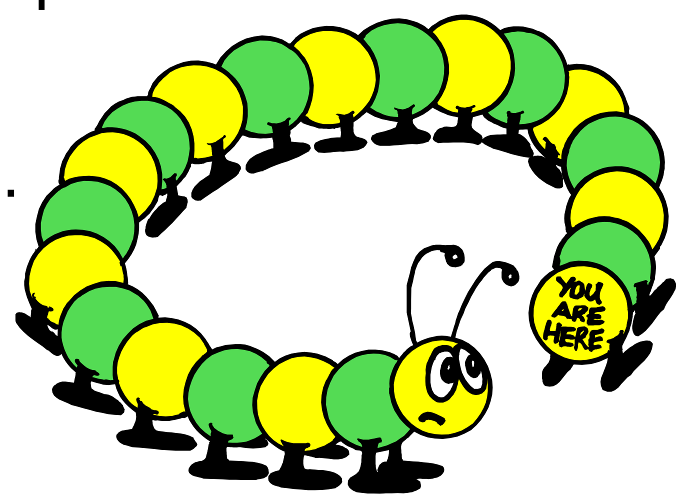
- Conclusion: Use a **multi-segment** approach.

- Natural candidate - the acoustic leaf.
  - Provides a good sized data chunk.
  - Expect similarities between segments.
  - Maintains database modularity.



Where were we…

- We aim to reduce the CTTS footprint, most of which stems from the stored speech segments.
- Test system uses a parametric spectral model: 32 amplitude parameters per speech frame.
- We want to compress further without compromising perceptual quality of synthesized speech.
- Next: let's review some prior art…



Previous work - 1

- Alternative speech compression schemes:
  - Sinusoidal coding (McAulay and Quatiere, 1986).
  - Harmonic+noise model (Stylianou, 2001).
  - Sinusoidal model adapted for TTS applications (Macon and Clements, 1999).
  - Decomposition of spectra into periodic and a-periodic components (d'Alessandro *et* al.,1998).
  - Iterative signal subtraction for sinusoidal model based analysis by synthesis (George and Smith, 1997).
  - Mel-frequency Cepstral Coefficients based coding (Chazan *et* al.,2002).
- These are allow for easy pitch modification and speech morphing, but do not deal with inter-frame redundancies.
- Adaptive speech compression schemes, such as Code Excited Linear Prediction, can be used. (Embedded CTTS, Karabetsos et. al, 2009).
- In this research we reused the existing signal compression scheme.

Previous work - 2

Acoustic inventory compression using asynchronous interpolation (Kain and van Santen, 2002 and 2007).

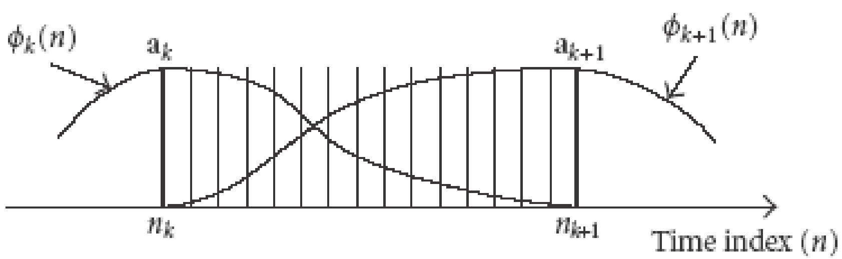
- Approximate each diphone by interpolating between a left and right phoneme template, using two non-linear interpolation functions.
- High compression ratios at the price of poor perceptual quality.
- Low flexibility- provides a single possible working point.
- Creating the template database is a complex process.

Previous work - 3

Temporal Decomposition (TD) approaches:

- Underlying concept: remove temporal redundancy by modeling the paramater evolution over time.
- Vector TD is applied to data or parameter vectors.
- Seek a set of target vectors and interpolation functions.

Used for low rate speech coding:



- Nguyen, 2002 ;
- Athaudage *et* al, 2003 ;
- Shechtman and Malah, 2004 emphasis on efficiency and perceptual quality.

Previous work - 4

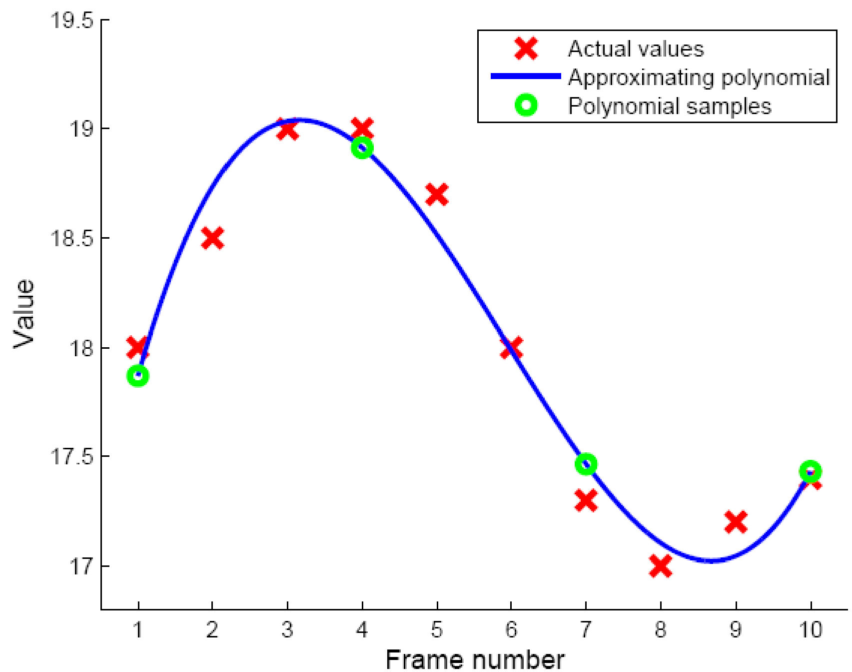
Temporal Decomposition (TD) approaches:

- Scalar TD: models the trajectory of a scalar, or a single vector parameter.
- Seek a P^{th} order model for N values.
  - DCT based model for the trajectories sinusoidal coding parameters, (Girin *et* al., 2007).
  - Polynomial TD - **Coming Soon to a seminar near you...**

ReCompression using Vectorial Polynomial Temporal Decomposition (TD)

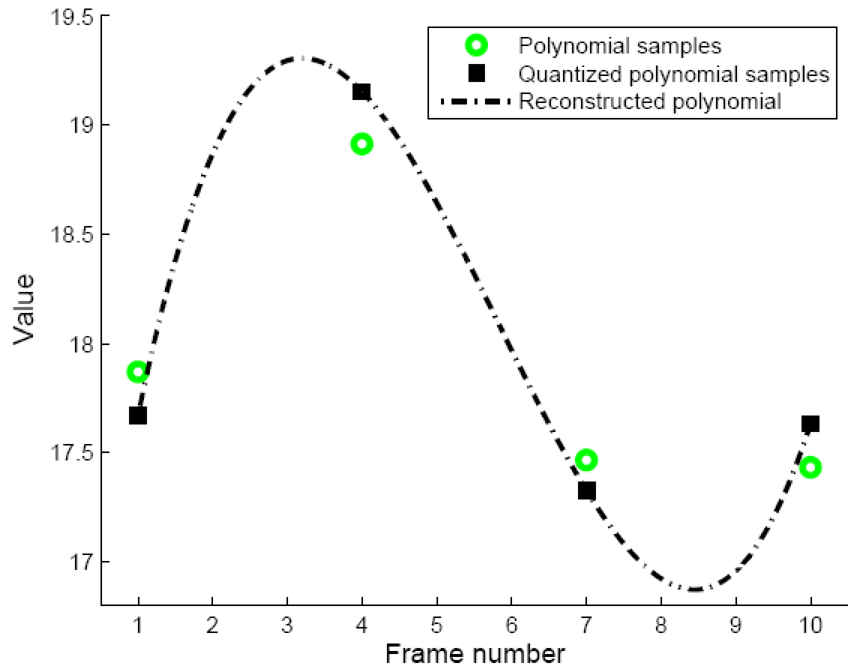
Polynomial TD

- Proposed for speech compression by Dusan et al., (2007).
- Represent the trajectory of N data points, such as the i^{th} coefficient in N frames, with the approximating polynomial of order P ($P<N-1$).
- Represent the polynomial by its P+1 samples.



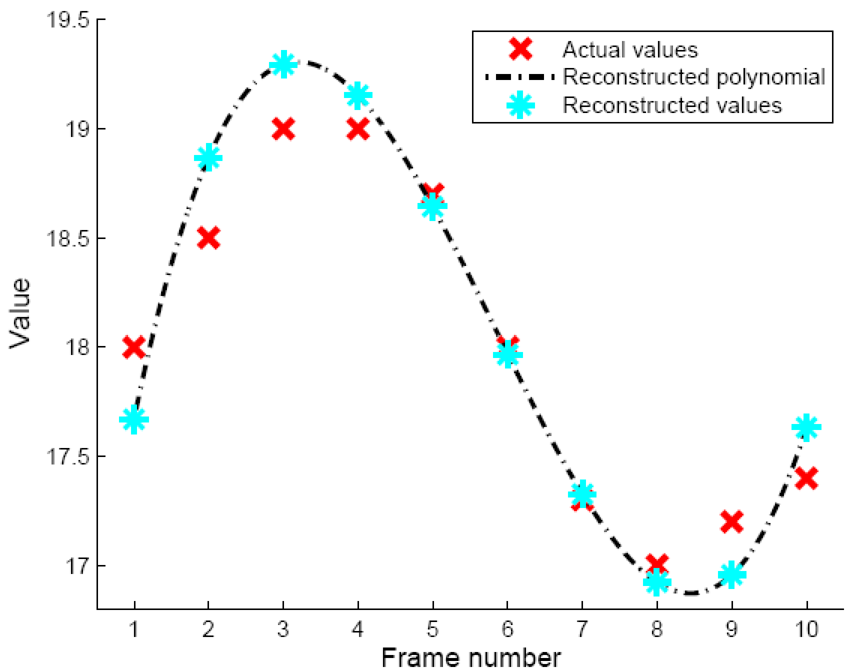
Polynomial TD

- Proposed for speech compression by Dusan et al., (2007).
- Represent the trajectory of N data points, such as the *i*th LSF coefficient in N frames, with the approximating polynomial of order P ($P<N-1$).
- Represent the polynomial by its P+1 samples.



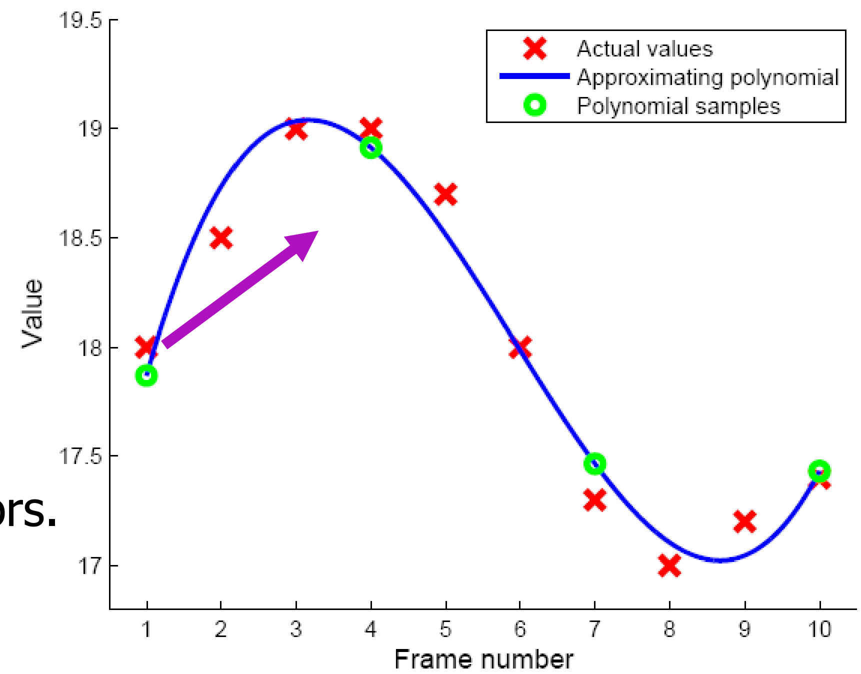
Polynomial TD

- Proposed for speech compression by Dusan et al., (2007).
- Represent the trajectory of N data points, such as the i^{th} LSF coefficient in N frames, with the approximating polynomial of order P (P<N-1).
- Represent the polynomial by its P+1 samples.



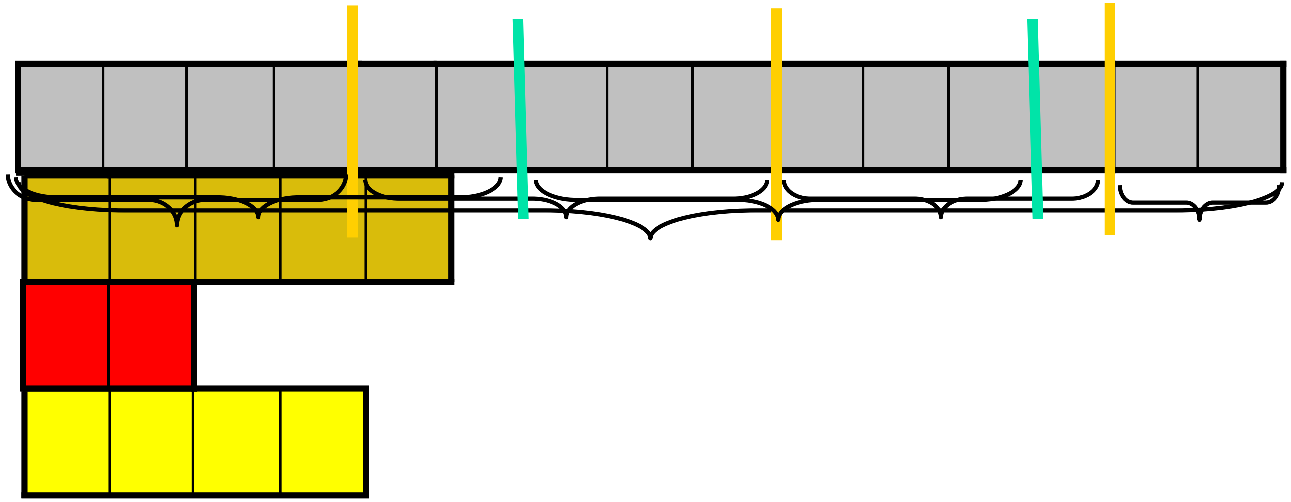
Polynomial TD

- Proposed for speech compression by Dusan et al., (2007).
- Represent the trajectory of N data points, such as the i^{th} LSF coefficient in N frames, with the approximating polynomial of order P ($P<N-1$).
- Represent the polynomial by its P+1 samples.
- We propose a vectorial form:
  - Apply to amplitude vectors.
  - Obtain P+1 representing vectors.



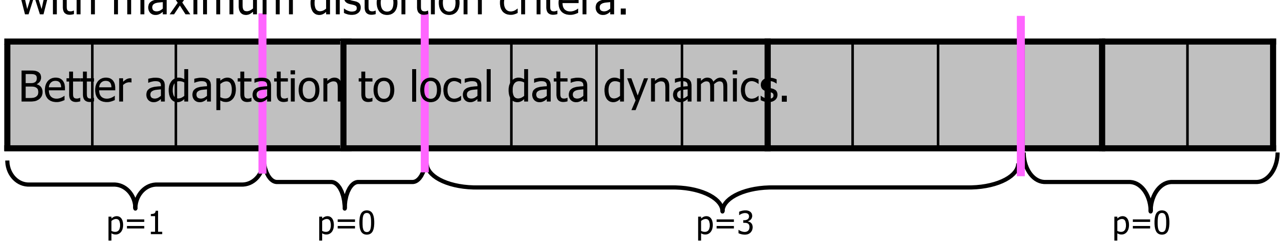
Polynomial TD for acoustic leaf

- Segments in each acoustic leaf are concatenated into a single *super-segment*.
- Concatenation order is selected either according to sequential order or using the re-ordering presented later.
- What segments should we use for polynomial TD?



Polynomial TD for acoustic leaf

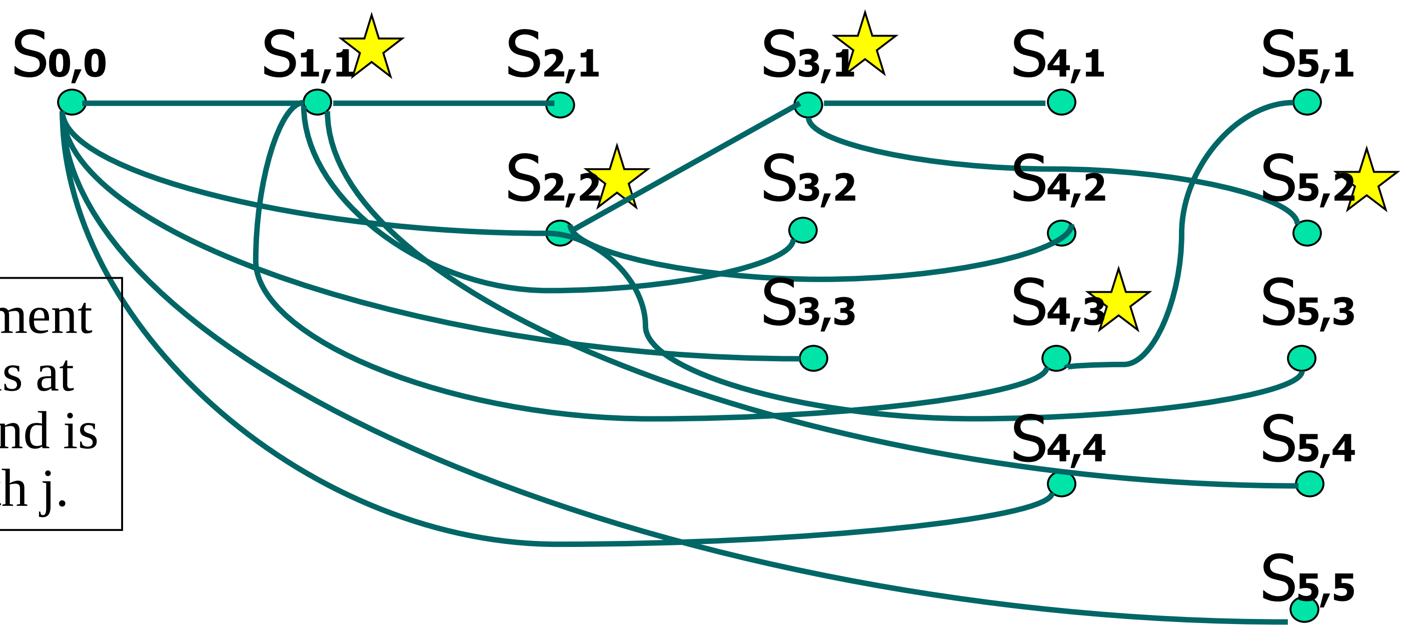
- Segments in each acoustic leaf are concatenated into a single *super-segment*.
- Concatenation order is selected either according to sequential order or using the re-ordering presented later.
- Split *super-segment* into short TD segments and fit each with a set of low order polynomials.
  - Low order polynomials (Low decoder complexity ; Less sensitive to quantization) with maximum distortion critera.
  - Better adaptation to local data dynamics.



TD Segmentation and order selection

- Based on “R/D optimal linear prediction”, Prandoni *et* al. (2000).
- First, build graph with all possible segmentations.
- For each segment find lowest polynomial order that guarantees ... nding rate.

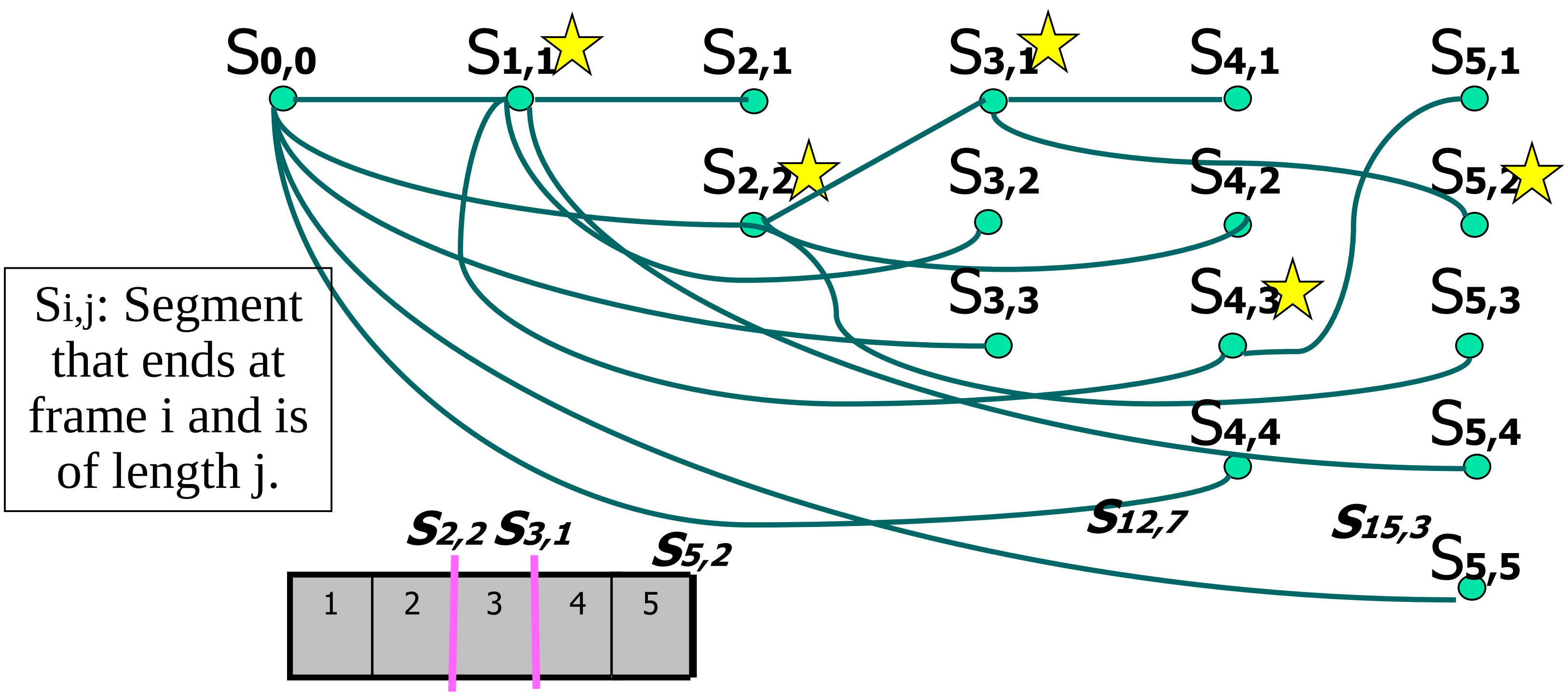
- Find lowest cost path across graph using backtracking.



$S_{i,j}$: Segment that ends at frame i and is of length j.

TD Segmentation and order selection

- Based on "R/D optimal linear prediction", Prandoni *et* al. (2000).
- First, build graph with all possible segmentations.
- For each segment find lowest polynomial order that guarantees target distortion; assign a cost based on the corresponding rate.
- Find lowest cost path across graph using backtracking.



Acoustic inventory compression using Polynomial TD

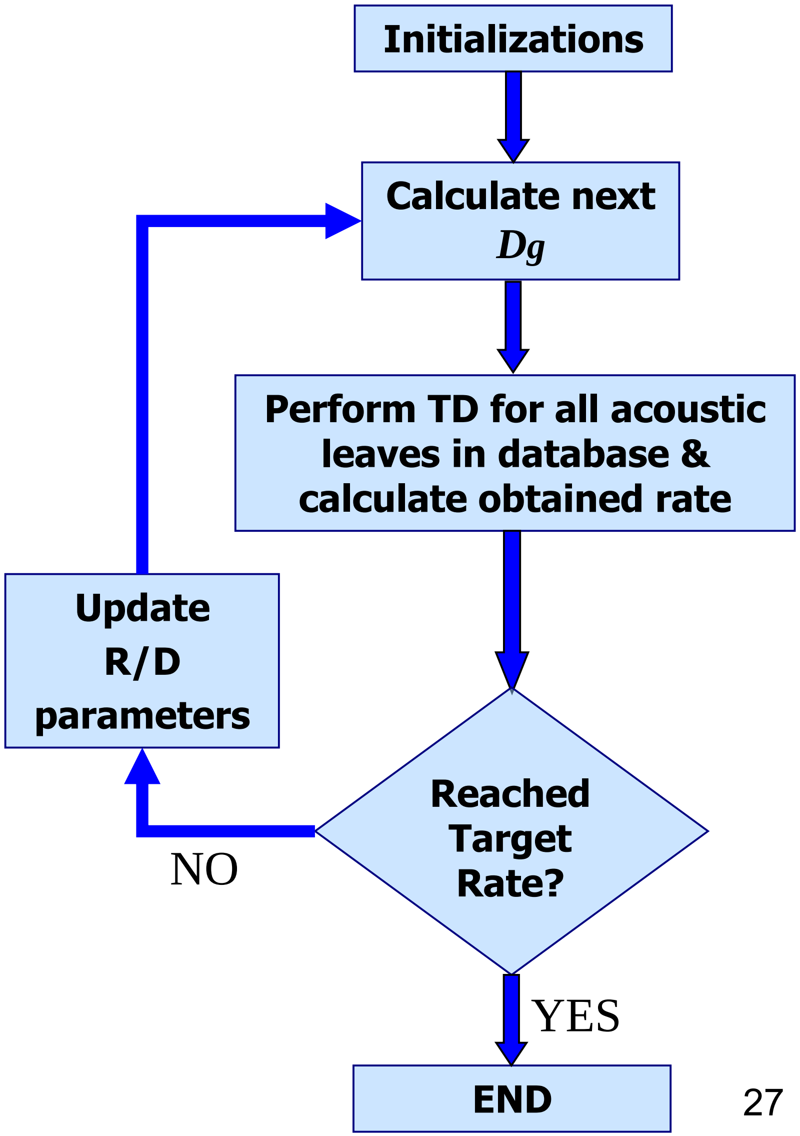
- Target rate, or compression ratio, is defined over the entire inventory.
- Goal: Obtain target rate (average) @ maximum quality (consistent).
- Distortion:
  - Distortion value is the maximum allowed for each frame in each segment.
  - For frame with original values *V* and reconstructed values *V'*, distortion is:

$$D_f = \frac{1}{32}\sum_{n=1}^{32}\left(V(n) - V'(n)\right)^2$$

  - MinMax MSE gave best results (compared to LSD, min-mean and more).
- Rate:
  - Each representing vector is quantized using the current, 86 bit per vector, split VQ quantizer.
  - When calculating the obtained rate, we also count algorithmic overhead bits.

Iterative rate–distortion algorithm

- We seek the minimum Dg for which rate = target rate, using a Bi-section search.
- Dg is the maximum allowed distortion among all frames in all segments in all leaves.
- In the TD, we look for segmentations and polynomial orders that will provide the lowest rate, given Dg.
- R/D parameters: rate and distortion values on active search interval edges.



Some results

PESQ scores for x2 recompression (evaluated on 10 sentences):

Setup / PESQ	Max polynomial order = 4	Max polynomial order = 1	Naïve Down-sampling 2:1
Minimum	3.45	3.39	2.48
Average	**3.55**	**3.66**	**2.84**

(*) these will improve slightly when combined with segment re-ordering.

Samples:

	Original	Max poly. order 4	Max poly. order 1
S.8 (worst)			
S.1 (avg.)			

ReCompression using Vectorial Polynomial TD - Summary

- We proposed a vectorial form of polynomial TD.
- We combined TD with jointly optimized sub-segmentation and polynomial order selection, under distortion & complexity constraints.
- We obtain much higher quality than linear interpolation, even when using only polynomials of order 0 and 1.
- An iterative algorithm converges to (any) target rate with minmax distortion criteria.
- Important feature: The compressed data lies in the same space as the original data set, thus enabling reuse of existing quantizers.

ReCompression using 3D-SADCT

"Do you think they mean us?"

2D DCT (reminder)

Discrete Cosine Transform definition:

$$F(u,v)=\frac{2}{n}\cdot C(u)\cdot C(v)\cdot\sum_{k=0}^{n-1}\sum_{l=0}^{n-1}f(k,l)\cdot\cos\left[\frac{(2k+1)\cdot u\pi}{2n}\right]\cdot\cos\left[\frac{(2l+1)\cdot v\pi}{2n}\right]$$

$$f(k,l)=\frac{2}{n}\cdot\sum_{u=0}^{n-1}\sum_{v=0}^{n-1}C(u)\cdot C(v)\cdot F(u,v)\cdot\cos\left[\frac{(2k+1)\cdot u\pi}{2n}\right]\cdot\cos\left[\frac{(2l+1)\cdot v\pi}{2n}\right]$$

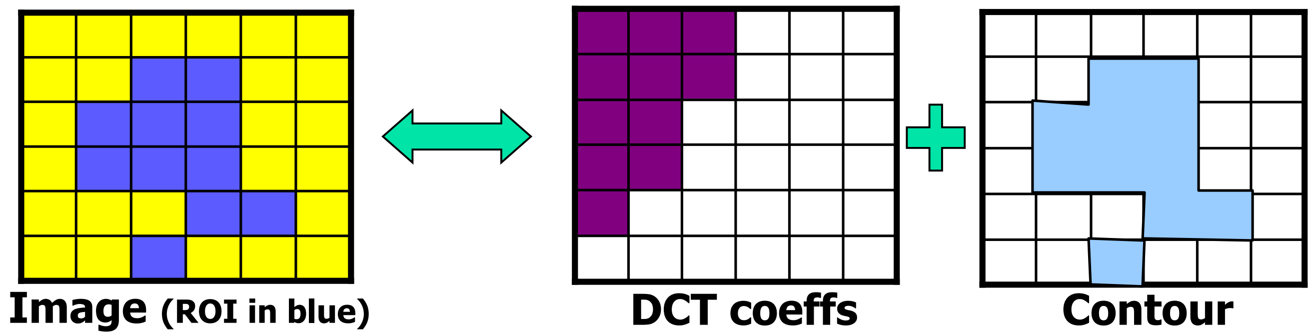
where: $$C(w)=\begin{cases}\frac{1}{\sqrt{2}} & for \quad w=0\\ 1 & otherwise\end{cases}$$

Properties:

- Energy preserving reversible transform.
- Removes redundancies (energy compaction).
- Separable, real valued and easy to compute.

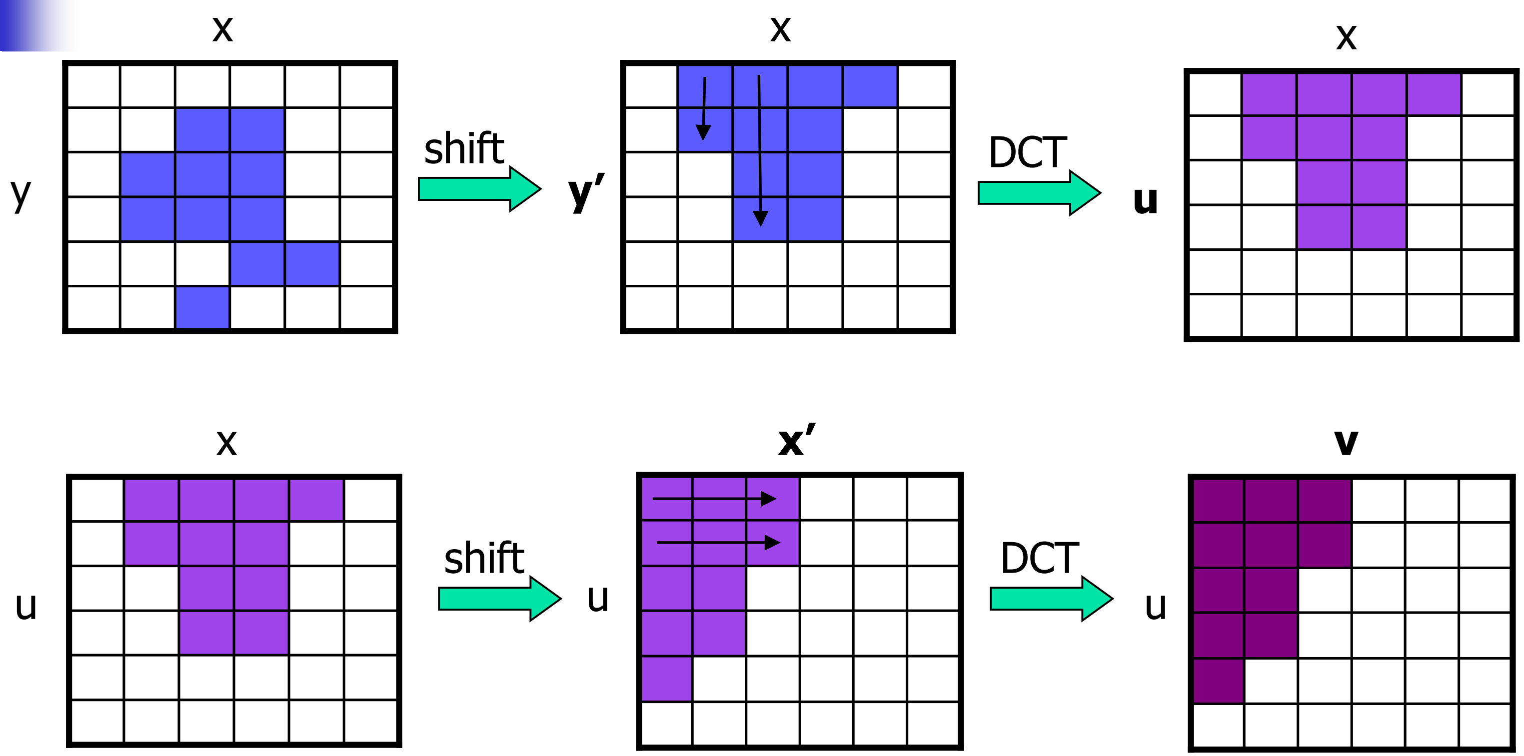
Shape Adaptive DCT (SADCT)

- Motivation: coding of an arbitrary shaped object
- Perform DCT only for pixels that belong to our object.



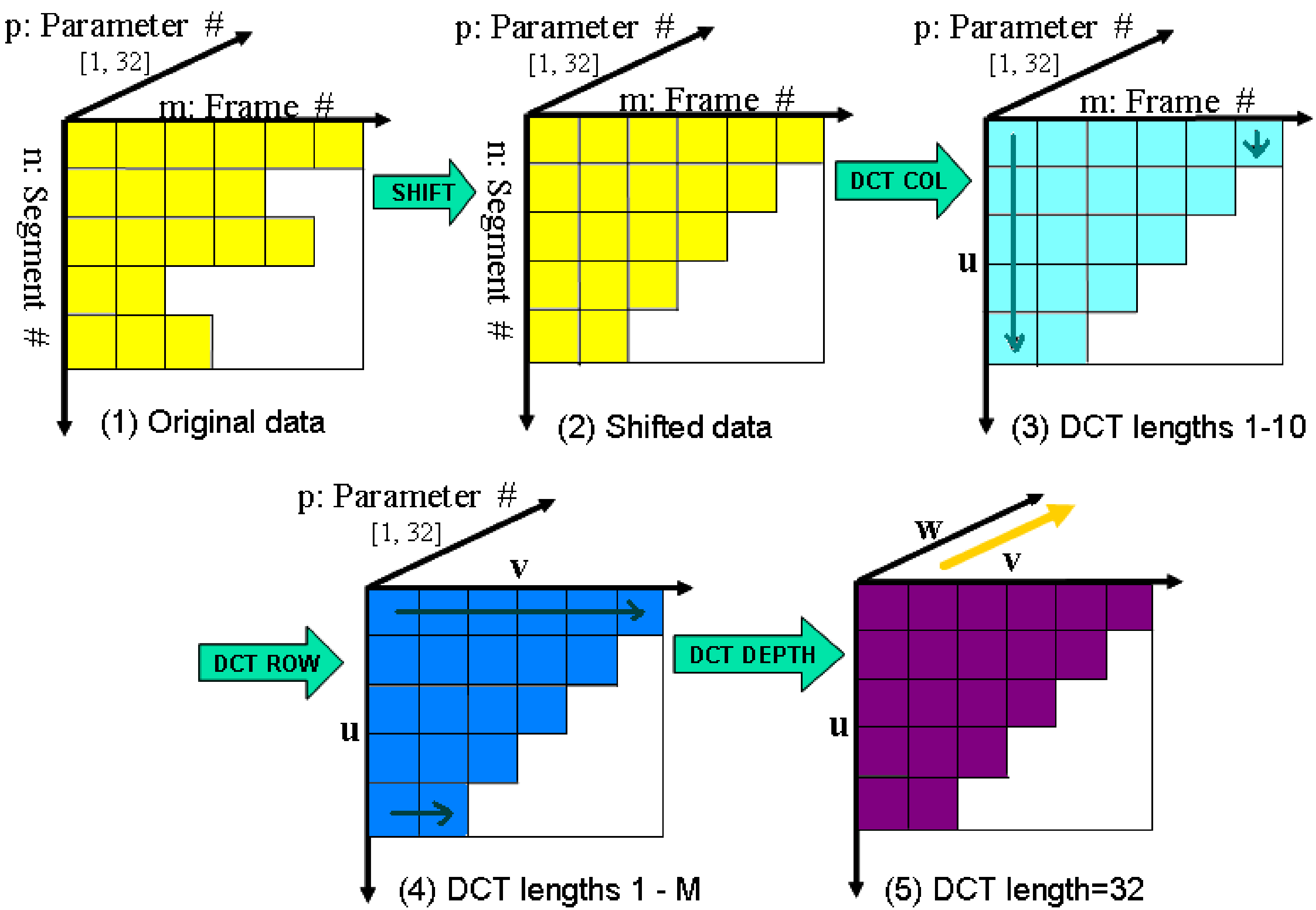
- Proposed for use within the MPEG-4 toolset for coding of audio-visual objects (Sikora and B. Makai, 1995).
- Extended to 3D for coding of hyperspectral images (Markman and Malah, 2001).

2D - SADCT

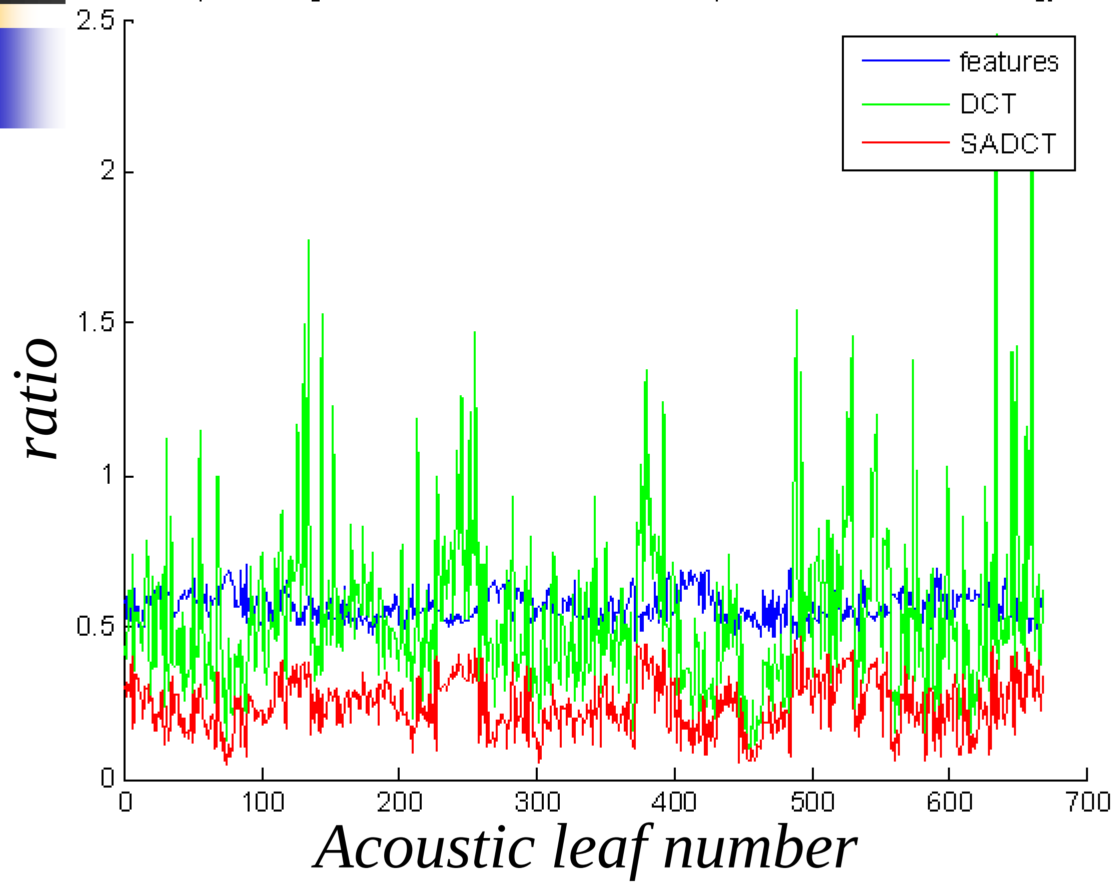


Top: DCT along columns
Bottom: DCT along rows

3D-SADCT for Acoustic Leaf



Potential of 3D-SADCT for acoustic leaf compression



Improved energy compaction!

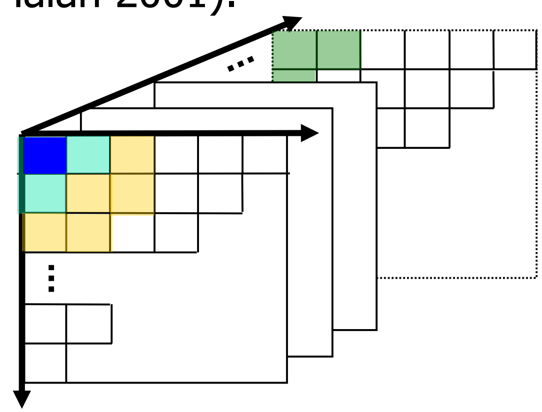
$$ratio = \frac{\text{number of coefficients that contain 95\% of the total energy}}{\text{number of features in acoustic leaf}}$$

Quantizer design

- 3D-SADCT results in a 3D data set with higher energy compactness.
- x2 recompression allows 43 bits per 32 element vector => VQ.
- We seek a set of quantizers that prioritize low frequency data (x 3 dims).

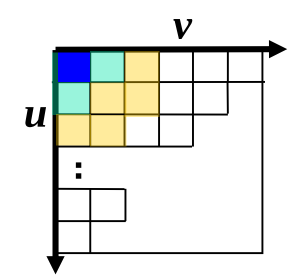
- Matrix quantization (Xydeas and Papanastasiou, 1999), or run-length coding do not apply well due to varying dimensionality and low bit-rate.
- Prior works on sub-band coding assume pre-known split between bands, or make assumption on the data or distortion functions that don't apply here. (Shoham & Gersho, 1988; Chatterjeet & Sreenivas 2008, Markman & Malah 2001).

- We propose an algorithm for methodical splitting and bit allocation, applied twice:
  - **Split all vectors into M vector groups.**
  - **Split vectors in each group into N sub-vectors.**



Splitting into groups

- Let ***Nu,v*** be the number of vectors in the database for bin $\{u,v\}$.
- Calculate ***STDu,v*** standard deviation of the vectors for each $\{u,v\}$.
- separate DC bin and allocate with 50 bits ; Initialize $(R_{avg})_1$;
- Allocate bits for each AC bin using:



$$\left(R^{opt}_{u,v}\right)_k = \left(R_{avg}\right)_k + \tfrac{1}{2}\log_2 \frac{\sigma^2_{u,v}}{\left(\prod_{p.q}\sigma^2_{p,q}\right)^{\frac{1}{Q}}} + \tfrac{1}{2}\log_2 \frac{W^2_{u,v}}{\left(\prod_{p,q}W^2_{p,q}\right)^{\frac{1}{Q}}}$$

Where:

$$\sigma_{u,v} = STD_{u,v}\cdot N_{u,v}\,; \quad Q = number\ of\ \{p,q\}\ for\ which\ N_{p,q}>0; \quad W_{u,v} = \frac{1}{u\cdot v}$$

- Cluster the obtained ***Ru,v*** values into M-1 groups (M=5).
- Set bit allocation of each group to the cluster centroid.
- Calculate ***Ravg***, if needed repeat until target rate is reached.

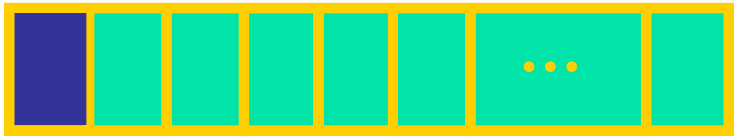
Obtained quantizer setup - 1

<table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14... end</th></tr>
<tr><td>1</td><td>50</td><td>46</td><td>46</td><td>42</td><td>42</td><td>42</td><td>39</td><td>39</td><td>39</td><td>39</td><td>39</td><td>33</td><td>33</td><td rowspan="10">If N(I,j)>0:
33
Otherwise:
0</td></tr>
<tr><td>2</td><td>46</td><td>46</td><td>42</td><td>42</td><td>39</td><td>39</td><td>39</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td></tr>
<tr><td>3</td><td>46</td><td>46</td><td>42</td><td>39</td><td>39</td><td>39</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td></tr>
<tr><td>4</td><td>46</td><td>42</td><td>42</td><td>39</td><td>39</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td></tr>
<tr><td>5</td><td>46</td><td>42</td><td>39</td><td>39</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td></tr>
<tr><td>6</td><td>42</td><td>42</td><td>39</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td></tr>
<tr><td>7</td><td>42</td><td>39</td><td>39</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td></tr>
<tr><td>8</td><td>42</td><td>39</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td></tr>
<tr><td>9</td><td>42</td><td>39</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td></tr>
<tr><td>10</td><td>39</td><td>39</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>33</td><td>0</td><td>0</td></tr>
</table>

Training is performed on the full acoustic leaf database, to avoid over-fitting.

Vector splitting

- For each group we wish to design a split VQ.
- DC allocation:
  - 8 bits for DC group (m=1).
  - $8 \cdot \frac{R(m)}{R(1)}$ for m=2,..,5.



- Then for each group, AC elements are each allocated bits using:

$$R_w^{opt} = R_{avg} + \tfrac{1}{2}\log_2 \frac{\sigma_w^2}{\left(\prod_{l=2}^{32}\sigma_l^2\right)^{\frac{1}{31}}} + \tfrac{1}{2}\log_2 \frac{W_w^2}{\left(\prod_{l=2}^{32}W_l^2\right)^{\frac{1}{31}}}$$

Where:

$$R_{avg} = \tfrac{R(m)-R_{DC}(m)}{31}; \quad W_w = \frac{1}{w},\ w = 2,\ldots,32$$

- Cluster the obtained $\boldsymbol{R_w}$ into clusters, s.t. the largest cluster contains 8 elements at most (limit on codebook size).
- Bit allocation for each sub vector is the sum of the allocations of its elements.

Obtained quantizer setup - 2

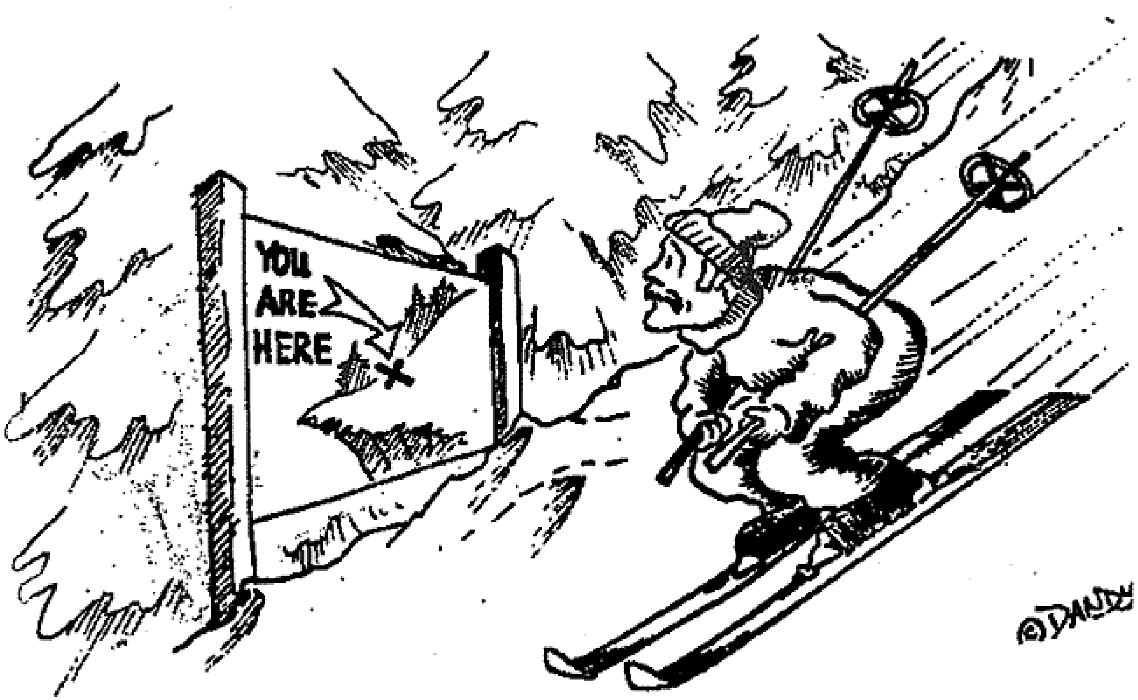
Group #1 Tot: 50	Elements	1	2	3	4	5:6	7:8	9:10	11:14	15:21	22:32
	length	1	1	1	1	2	2	2	4	7	11
	Alloc.	8	5	4	4	6	5	4	6	6	2
Group #2 Tot: 46	Elements	1	2	3	4	5:6	7:8	9:13	14:21	22:32	-
	length	1	1	1	1	2	2	5	8	11	-
	Alloc.	7	5	4	4	6	5	8	6	1	-
Group #3 Tot: 42	Elements	1	2	3:4	5:6	7:8	9:13	14:21	22:32	-	-
	length	1	1	2	2	2	5	8	11	-	-
	Alloc.	7	5	8	6	4	7	5	0	-	-
Group #4 Tot: 39	Elements	1	2	3:4	5:6	7:8	9:13	14:21	22:32	-	-
	length	1	1	2	2	2	5	8	11	-	-
	Alloc.	6	5	7	5	4	7	5	0	-	-
Group #5 Tot: 33	Elements	1	2	3:4	5:6	7:10	11:18	19:32	-	-	-
	length	1	1	2	2	4	8	14	-	-	-
	Alloc.	5	5	7	5	6	5	0	-	-	-

For each sub-vector of each group, a VQ is designed with the LBG algorithm (Linde, Buzo and Gray 1980).

ReCompression using 3D SADCT - Summary

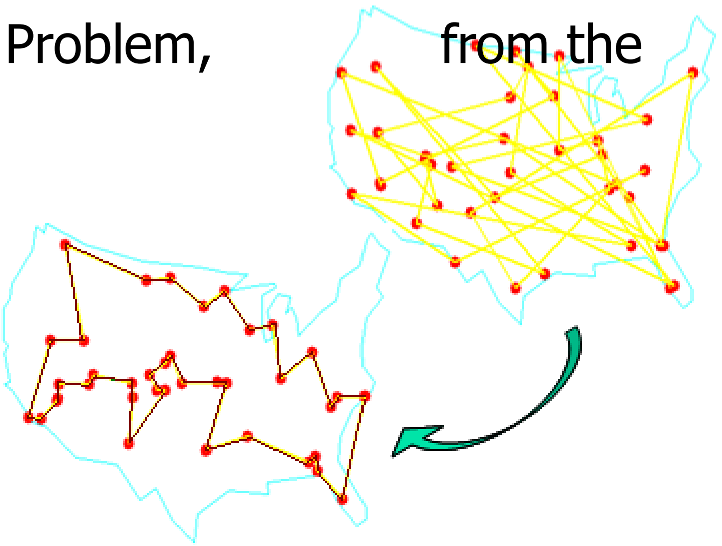
- We apply the 3D SADCT, conventionally used for image/video coding, to a novel setup, thus enabling efficient compression of CTTS acoustic leaves.
- We propose a methodical approach to split VQ design, which provides splitting points and bit allocation based on data statistics. (we used two variants of this algorithm).
- Pro: Obtained PESQ score at x2 recompression: 3.84.
- Con: Algorithm has low flexibility. To obtain a new working point full quantizer re-design must be performed.

Segment reordering



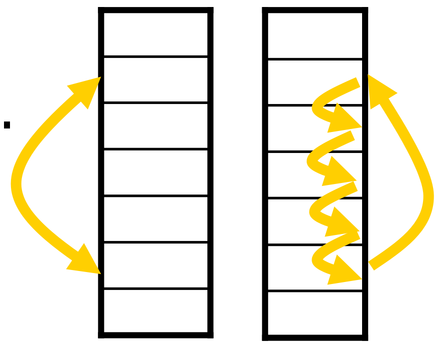
Segment Ordering

- Segments in each acoustic leaf have an arbitrary order.
- We wish to find the 'best' order, offline, prior to compression.
  - For Polynomial TD: determines concatenation order.
  - For 3D SADCT: affects energy compaction of transform along columns.
- A form of the Traveling Salesperson Problem, from the realm of combinatorial optimization.
- Not all TSP solutions apply since our cost function isn't Euclidean.



Segment Ordering – cont.

- Possible solutions:
  - Binary Switching Algorithm (Zeger & Gersho, 1990).
  - Enhancement of BSA (Spira & Malah, 2000).
  - Simulated Annealing (Kirpatrick, 1983).



- We propose a combined approach, based on the Metropolis algorithm (Metropolis, 1953) :
  - Complexity similar to the enhanced Binary Switching Algorithm.
  - Advantage: can exit local minima, as in the SA approach.

- The algorithm goal is to find the order that minimizes a specified cost function.

Ordering cost functions: Polynomial TD

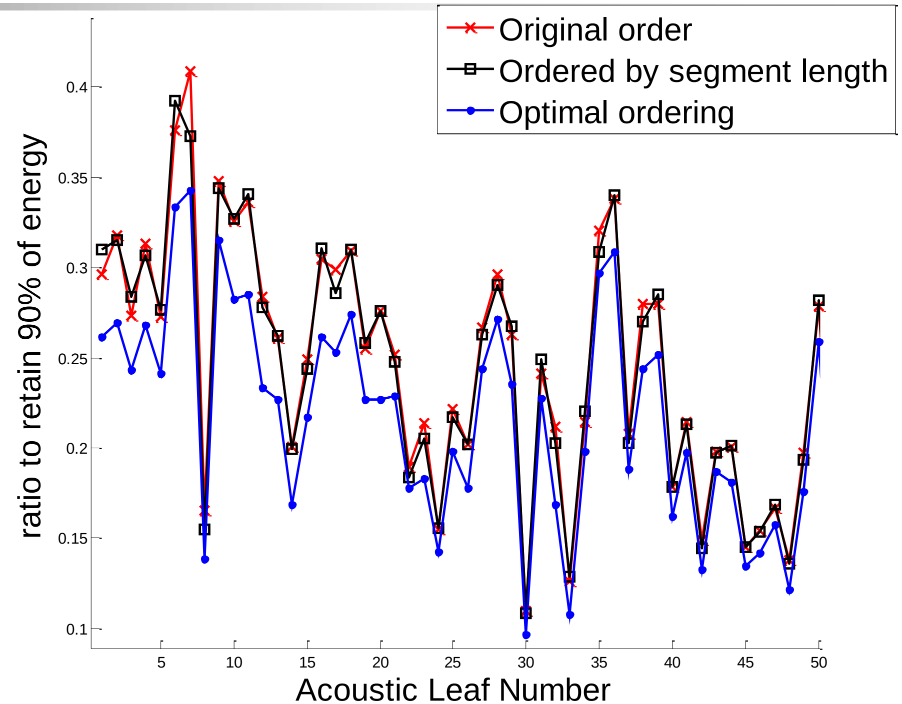
- Define the *super-segment* as the concatenation of all segments in a specified order.
- For the i^{th} parameter, $(i=1,..,32)$, approximate its trajectory along the *super-segment* with a second order polynomial – Pol_i. Then the cost is:

$$C_{TD} = \sum_{i=1}^{32} w_i \sum_{n=1}^{N} \left(V_{n,i} - Pol_i(n)\right)^2 \; ; \quad w_i = \frac{const}{i}$$

  i.e., the weighted MSE between $V_{n,i}$, the actual value of parameter i at frame n of the *super-segment*, and its polynomial approximation.
- This cost measures the smoothness of the *super-segment*, while prioritizing parameters corresponding to lower frequencies.

Ordering cost functions: 3D SADCT

- For 3D-SADCT the ordering affects the vertical transform, thus affecting overall obtained energy compaction.



- We wish to maximize the energy in G_2, the first non-DC group.
- The cost is defined as:

$$C_{SADCT} = 1 - \frac{\sum_{\{u,v\}\in G_2} \sum_{w=1}^{32} F_{u,v,w}^2}{\sum_{\{u,v\}\in G_{2,\dots,M}} \sum_{w=1}^{32} F_{u,v,w}^2}$$

Proposed ordering algorithm

- For small leaves (7 segments or less) all possible arrangements are evaluated and the one with lowest cost is kept.
- For large leaves, we use a Metropolis Based Ordering approach:
- Given a cost function, a 'move' generator and iteration budget:
  1. Initialize: set initial order, and set T to desired temperature.
  2. Calculate current cost, C.
  3. Perform a random move, calculate C_{new} and the $\Delta C = C_{new} - C$.
  4. If $\Delta C < 0$ keep the move.
  5. If $\Delta C > 0$ and $\exp\{-\Delta C/T\} > rand(0,1)$, also keep the move.
  6. If number of iterations below budget: GOTO 2.
  7. Select point with lowest cost (non real-time).

Where were we…

- We presented two recompression algorithms:
  - Vectorial polynomial TD with optimal segmentation and polynomial order selection.
  - 3D-SADCT with automatic multi-split VQ design.
- We presented an algorithm for segment reordering.
- Next: Results…

Experimental results

PESQ scores for x2 recompression (evaluated on 10 sentences):

Setup / PESQ	POL TD Max poly. order=4		POL TD Max poly. order=1		SADCT	
	No ReOrd	w. ReOrd	No ReOrd	w. ReOrd	No ReOrd	w. ReOrd
Minimum	3.45	3.49	3.39	3.51	3.53	3.65
Average	**3.55**	**3.67**	**3.66**	**3.69**	**3.84**	**3.85**

Samples:

	Original	POl TD; max order=1		SADCT	
		No Reo	W.Reo	No Reo	W. Reo
S.8 (worst)					
S.1 (avg.)					

Summary

- Two recompression approaches were presented:
  - Vectorial Polynomial TD with adaptive segmentation and polynomial order selection.
  - 3D SADCT with methodical quantizer design.
- A Metropolis based segment reordering algorithm was proposed.
- Applying these algorithms to small footprint CTTS acoustic leaf re-compression, provides a factor of 2, without degrading perceptual quality.
- The proposed algorithms are generic and may be applied to a variety of re-compression challenges.

Future work

In the scope of acoustic leaf compression:

- Examining the recompression and overall performance when using alternative signal models, such as sinusoidal modeling.
- Applying the proposed algorithms to phase parameters.

In the scope of the proposed algorithms:

- Trying to develop quantizer design that is even more 'automatic'.
- Applying the proposed algorithms to other test cases, such as:
  - Sign language databases (http://archive.ics.uci.edu/ml/datasets/Libras+Movement).
  - Image classification databases (Bag of Words).
  - Personalized content recommendation databases.

Thank you