Technion - Israel Institute of Technology
Department of Electrical Engineering
Signal and Image Processing Laboratory

Packet Loss Concealment for Audio Streaming

Hadas Ofir

M.Sc. Research under the supervision of
Prof. David Malah

May 2006

Overview

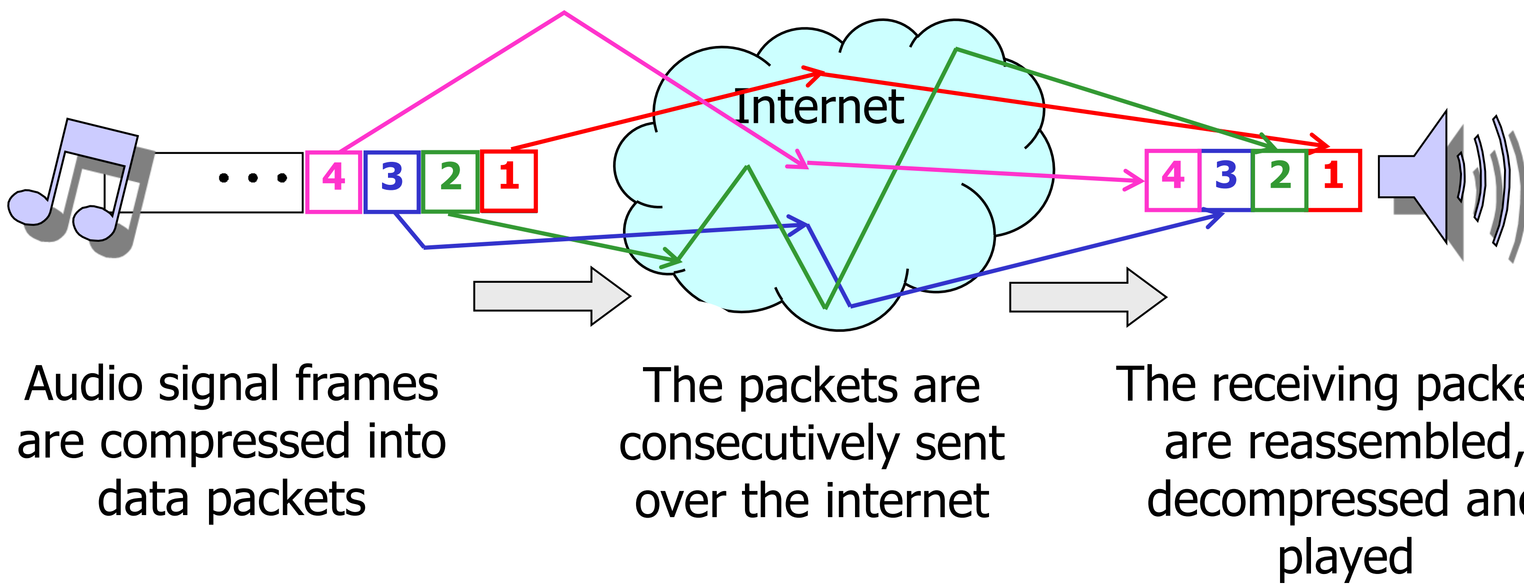
- A new algorithm for packet loss concealment
  - Audio streaming applications.
  - MPEG-audio coders.
  - Based only on the data available at the receiver.
- Missing data is interpolated in the DSTFT domain (GAPES & MAPES algorithms).
- Implemented on MP3 coder, suitable also for MPEG-2/4 AAC.
- Subjectively tested and found to perform better than previous works, even at high loss rates.

Presentation Outline

- Internet Audio Streaming and Packet Loss problem.
- MP3 compression, MDCT.
- Concealment domain alternatives.
- Previous Works
- MDCT $\leftrightarrow$ DSTFT conversion.
- GAPES and MAPES interpolation algorithms.
- Proposed concealment algorithm.
- Subjective tests results.
- Conclusion & Future directions.

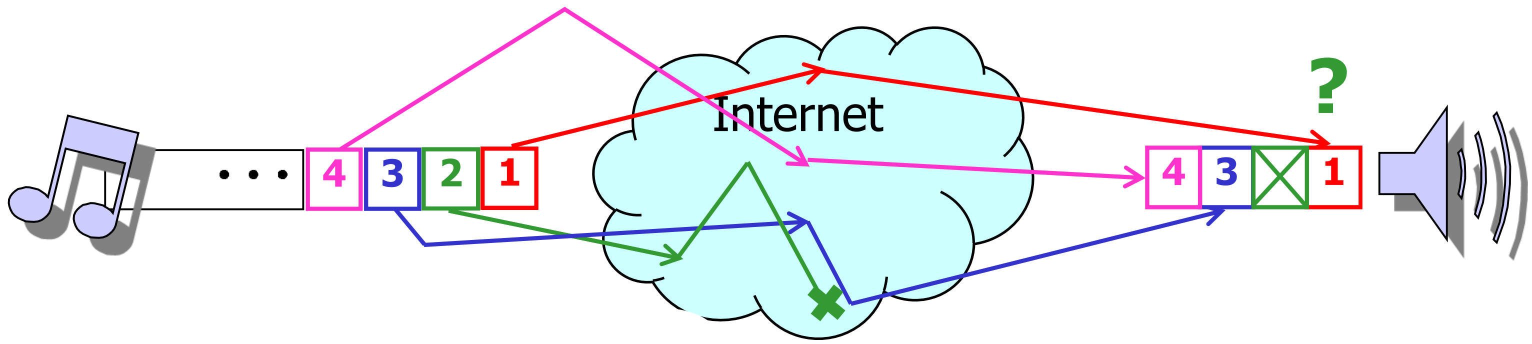
Internet Audio Streaming

- A Real-Time application.
- Connectionless protocol:
  Each packet may use a different route.



Packet Loss

- Internet broadcasting doesn't assure quality of service (QoS).
  - Data packets are often delayed or discarded during network congestions.

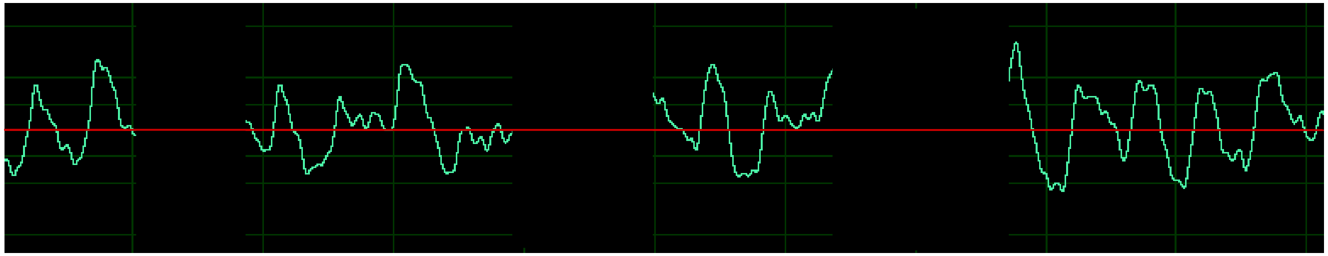


- Each loss, unless concealed, produces an annoying disturbance.
  - Example

Original 10% loss 20% loss 30% loss

Packet Loss Concealment

- Fill-in the gap with an approximation to the original signal.



- Goal: Generate a good enough replacement.
  - Good enough = won't be noticed by a human listener.
- Problem
  - A typical audio packet takes around 1000 samples.
  - Even a single lost packet creates a very wide gap that is difficult to interpolate.
- So, what is the best concealment method?...

Packet Loss Recovery Techniques

- Sender-based techniques
  - Forward Error Correction (FEC)
  - Data Interleaving
  - Data-Hiding
- Receiver-based techniques
  - Insertion-based interpolation:
    - Noise substitution
    - Packet Repetition
  - Time-domain interpolation:
    - Waveform substitution
    - Pitch replication
  - Advanced interpolation:
    - Parametric
    - Compressed domain

The proposed solution is receiver-based

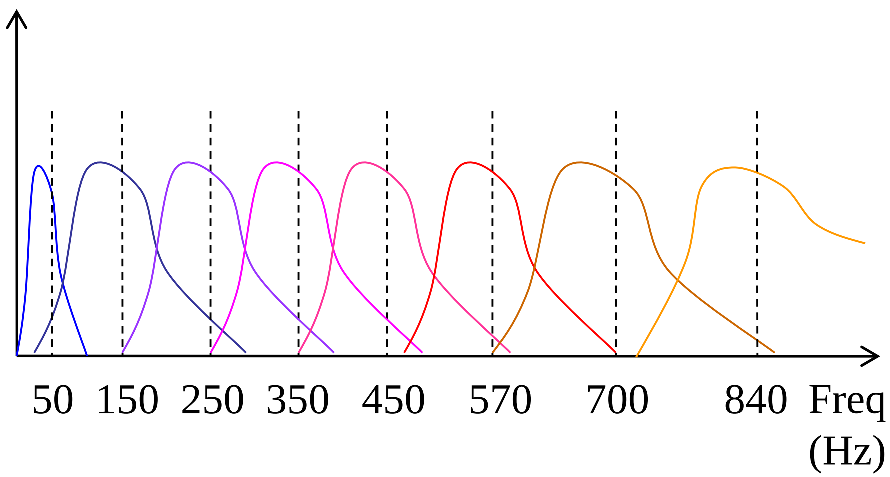
Platform: MP3 Coder

- MPEG-1 Audio Layer III, **a.k.a. MP3**:
  - Used for internet audio delivery.
  - An efficient way to store audio files.
  - Successors: MPEG-2/4 AAC.
- Compresses signals sampled at rates of 32, 44.1 or 48 kHz, to rates of 32 to 320 kbps per channel.
- MPEG-audio coders are perceptual audio coders.
- The coded signal is divided into frames of 576 samples. Every two frames form an **MP3 packet**.

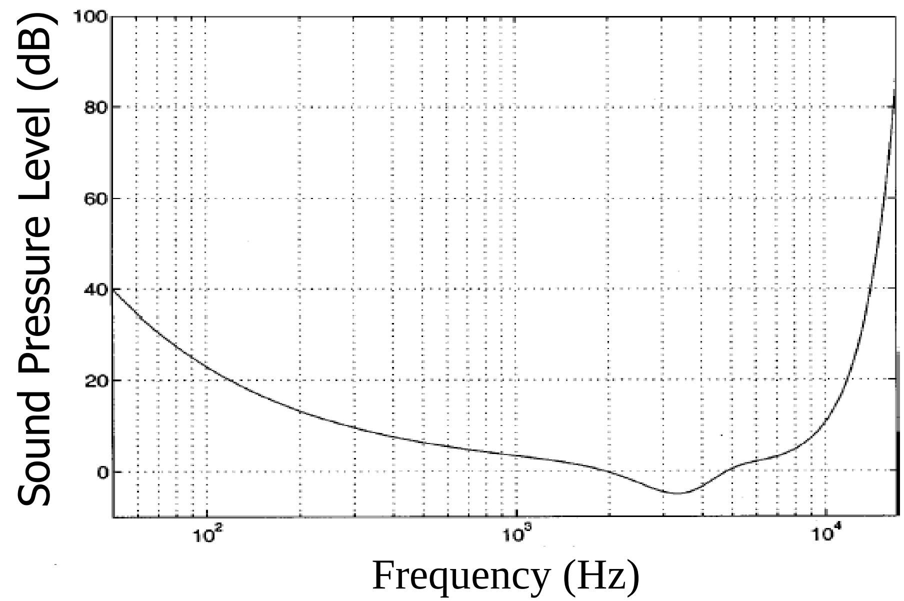
MP3 Coding – Psychoacoustics 1

- Critical Bands

  A filter-bank designed by evolution.

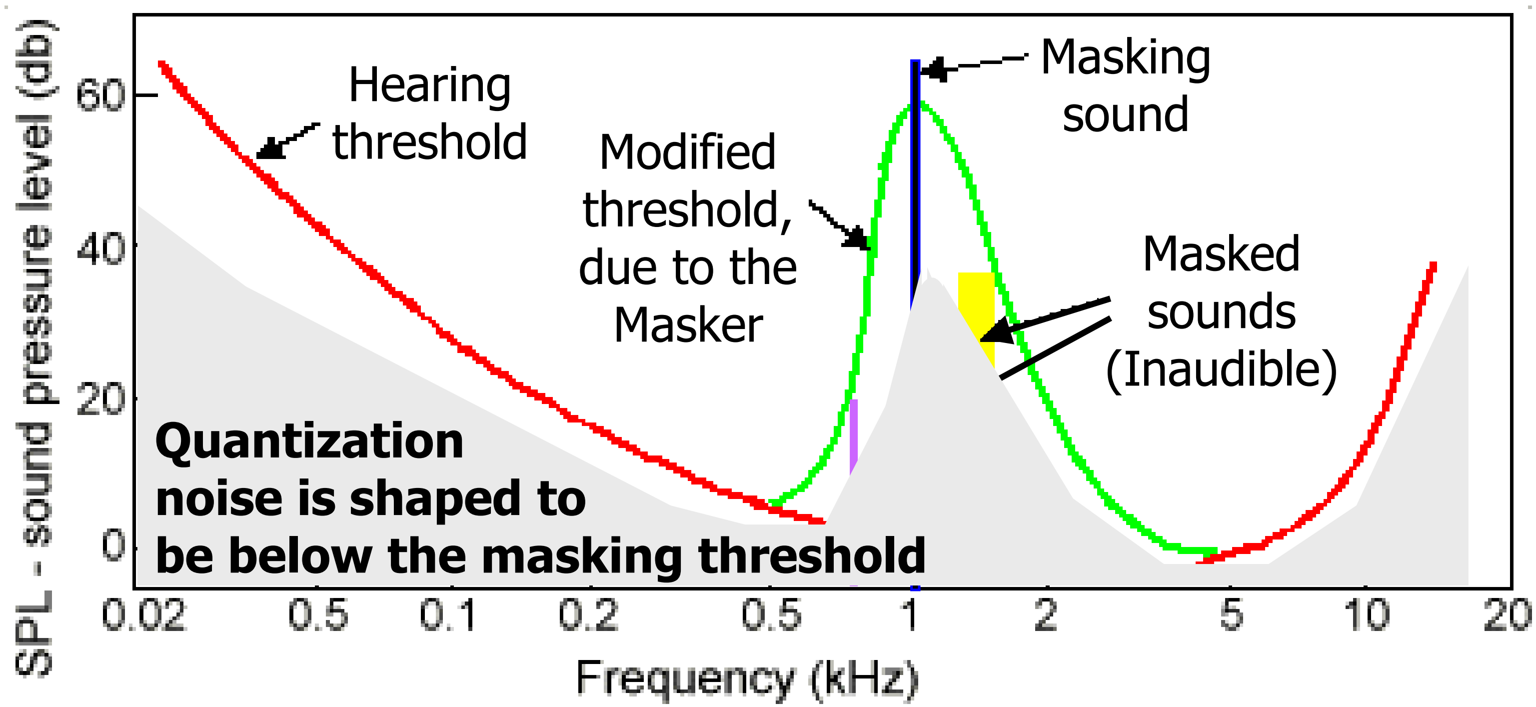


- Absolute threshold of hearing



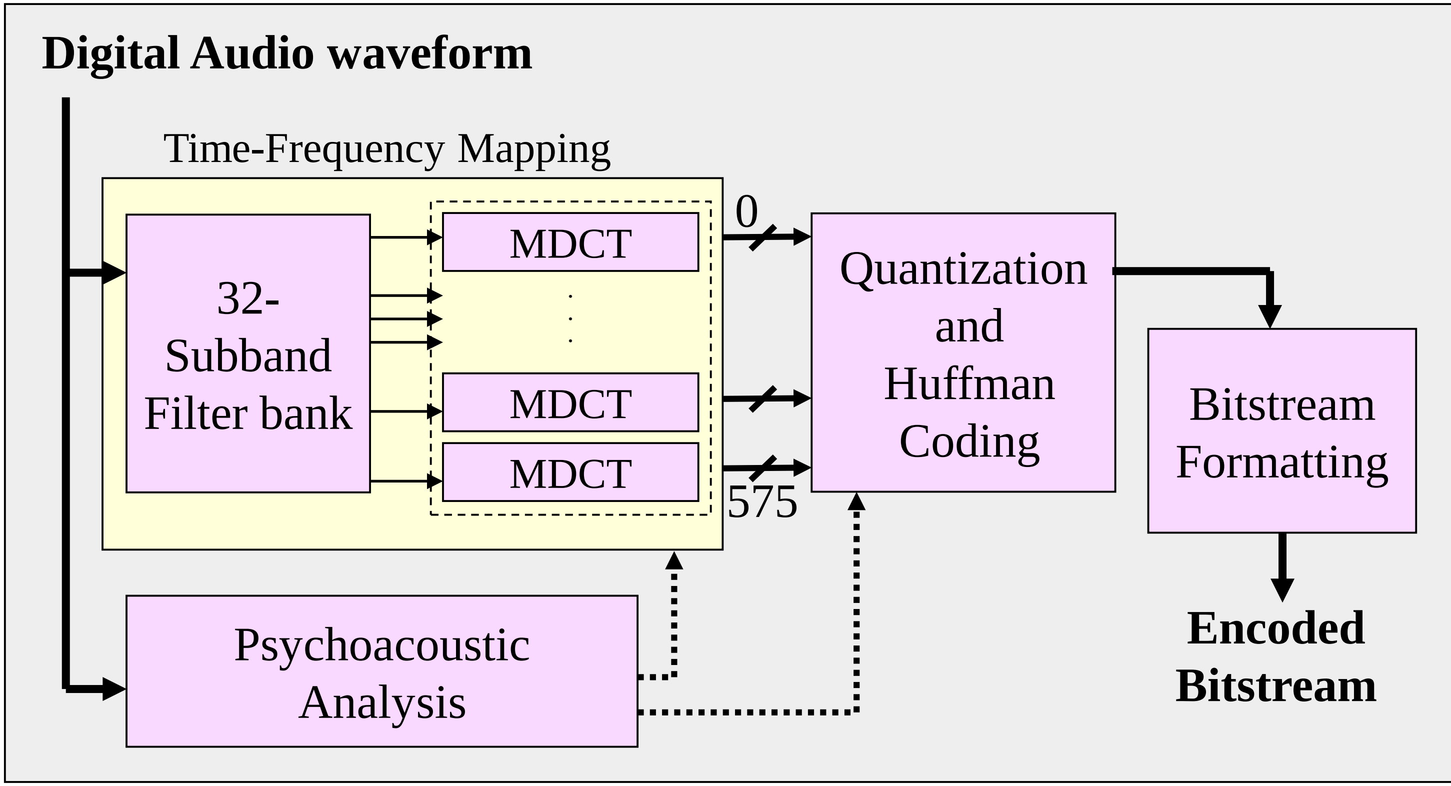
MP3 Coding – Psychoacoustics 2

- Frequency Masking



- Temporal Masking

MP3 Coding - Encoder



One lost MP3 packet (1152 samples) is equivalent to **only** 2 lost MDCT coefficients per frequency bin!

MDCT - Definition

- The MDCT is a real-valued transform, turning 2*N* time samples into *N* MDCT coefficients:

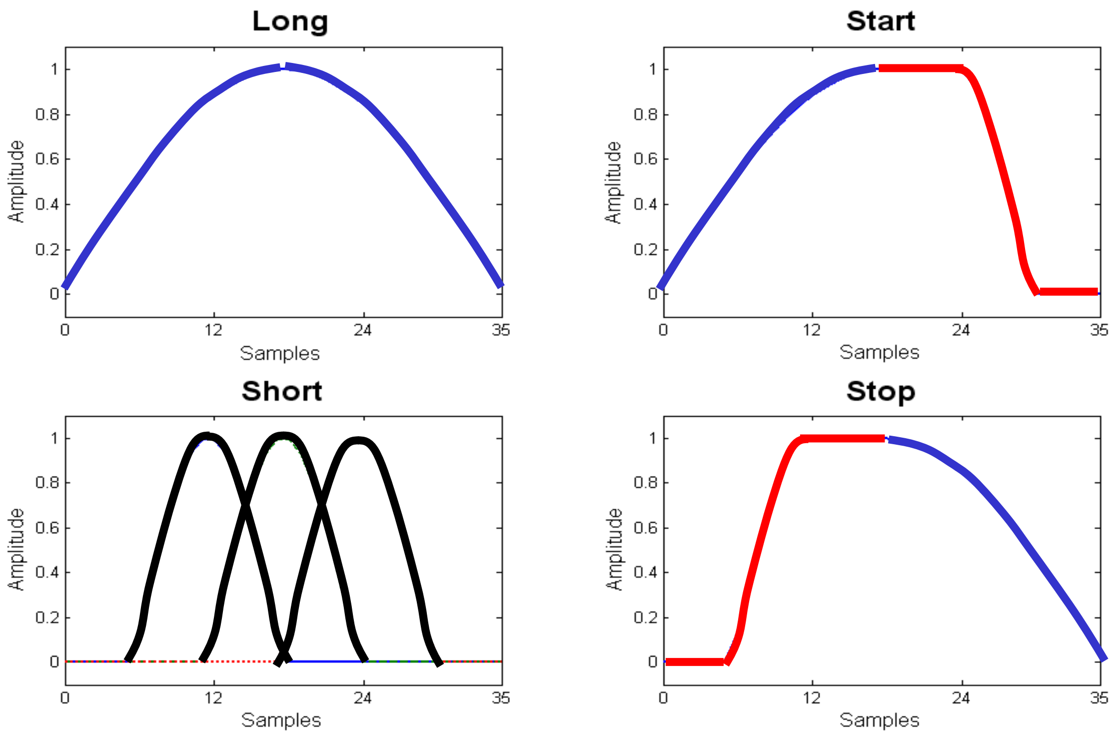
$$X^{MDCT}[k] = \sum_{n=0}^{2N-1} x[n] \cdot w[n] \cdot \cos\left(\frac{\pi}{N} \cdot \left(n + \frac{N+1}{2}\right)\left(k + \frac{1}{2}\right)\right) \qquad ,0 \le k \le N-1$$

- **Lossless** transform, if certain conditions are satisfied:
  - 50% overlap between successive transform windows.
  - Using specially designed window functions, $w[n]$.
- The samples are reconstructed using an overlap & add (OLA) procedure on the output of the inverse transform.

Reconstruction of a whole segment (2N samples) requires 3 consecutive MDCT blocks.

MDCT – Windows

- MP3 defines 4 window functions for the MDCT.
  - **Long**: better frequency resolution for stationary segments.
  - **Short**: better time resolution for transients.
  - The windows are built using 3 basic half-window units.

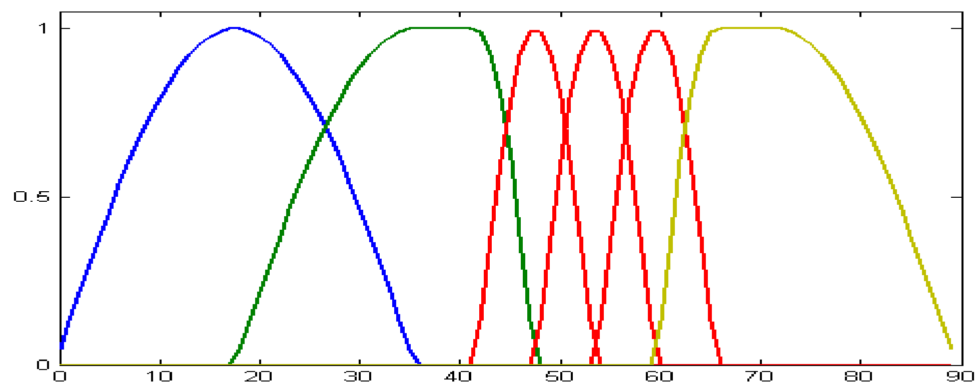


MDCT – Window Ordering

- A transition window always comes between Short and Long windows.
- The ordering rules can sometimes help restore the original window type in case of packet loss.

- Example:
  Long-missing$_1$-missing$_2$-Stop could only match one possible pattern:

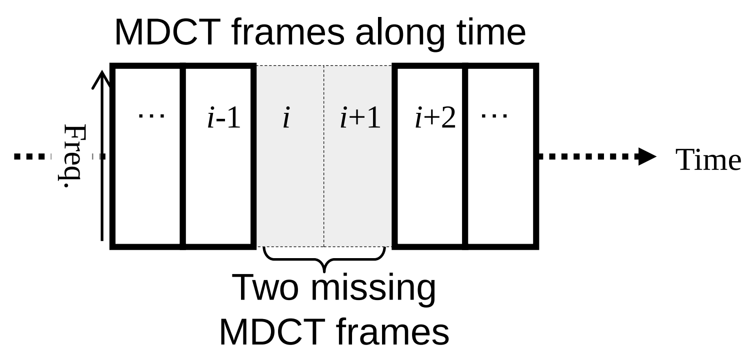
  missing$_1$ = **Start**
  missing$_2$ = **Short**



Transition Between Domains

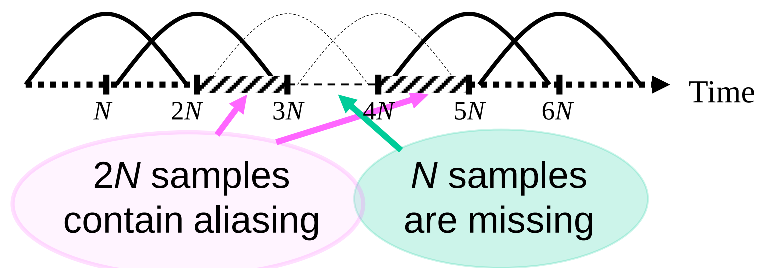
In the MDCT domain

- MDCT frame = N coefficients.
- Each frame represents $2N$ samples.



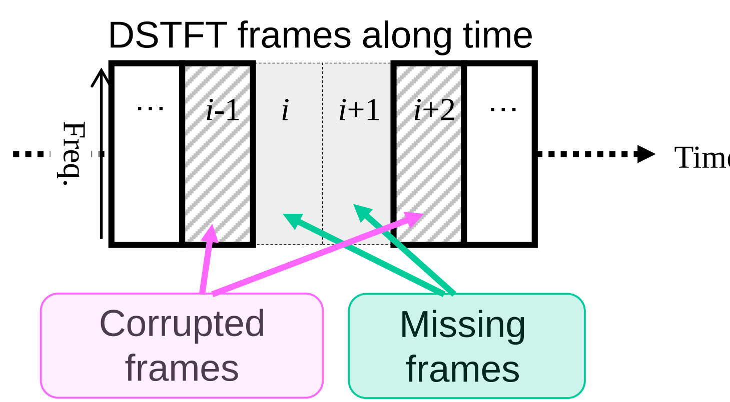
In the time domain

- Time segment = $2N$ samples.
- 50% overlap between segments.
- 2 lost MDCT frames ➡ $3N$ samples are affected.



In the DSTFT domain

- DSTFT frame = $2N$ conjugate-symmetric coefficients.
- Each frame represents $2N$ samples.
- 2 lost MDCT frames ➡ 2 missing + 2 corrupted DSTFT frames.



Concealment Domain

Q consecutive missing packets are equal to…

- Time domain: (2Q+1)·576 missing samples.
- ☑ MDCT domain: 2Q missing coeff. per frequency bin.
- ☑ DSTFT domain: (2Q+2) missing coeff. per frequency bin.

Previous Works

- Packet Repetition [MP3 Standard, Annex E]
  - The lost packet is replaced by a copy of the packet that was received last.
  - Complexity: O(1).
- Statistical Interpolation (SI)
  [Quackenbush and Driessen, 115th AES conv., Oct. 2003]
  - A sample-restoration algorithm, originally designed for auto-regressive (AR) time-domain signals, is applied in the MDCT domain.
  - The coefficients of each frequency bin along time are considered as a separate sequence with missing samples.
  - Benefits: Applied directly in the compressed domain.
  - Limitations: Limited loss patterns.
  - Complexity: $O(L^2)+O(P_m^3)$ where L is the AR order and $P_m = 2Q$ is the number of missing samples.

Limitations of concealment in the MDCT Domain

The loss gap is easier to handle in the MDCT domain, but:

- MDCT coefficients along time show rapid sign changes.

  Solution: Use a domain with less signal fluctuations.

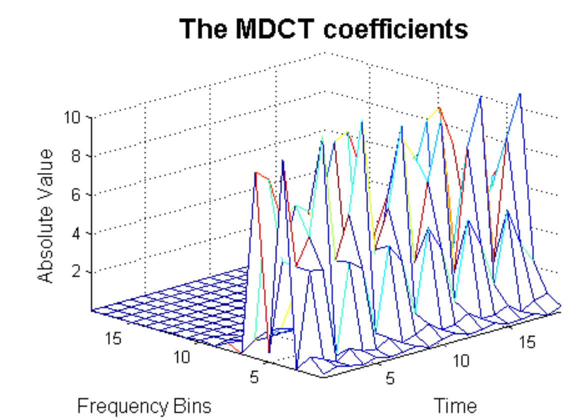
- Different window types have different frequency resolutions.

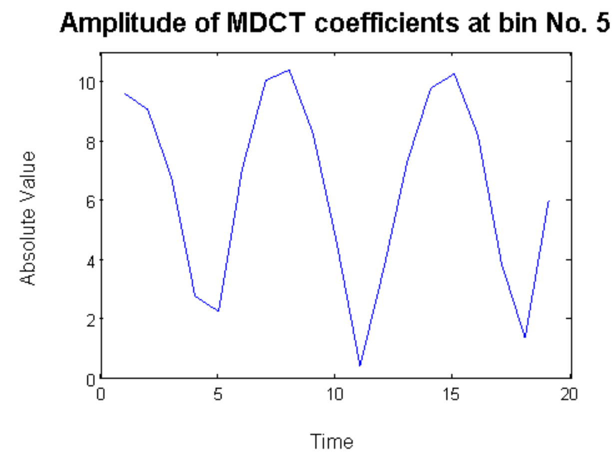
  Solution: Use a single window type when converting the data to another domain.

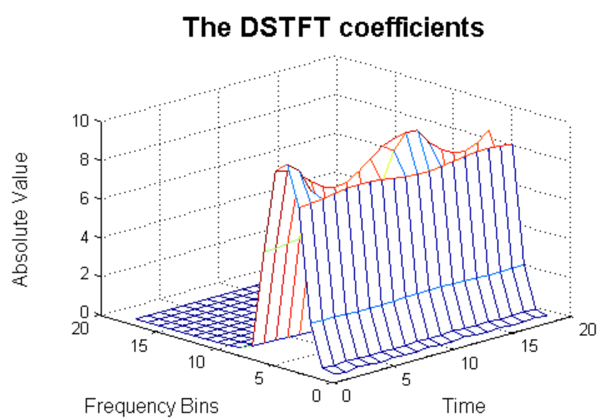
Our choice: The DSTFT domain

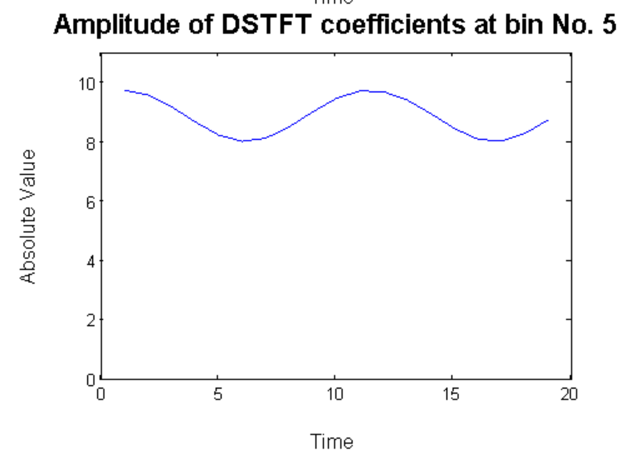
MDCT vs. DSTFT – Example

Time Signal: $\cos\left(\frac{\pi}{N}(5+0.15)n+0.7\right)+\cos\left(\frac{\pi}{N}(3+0.34)n\right)$



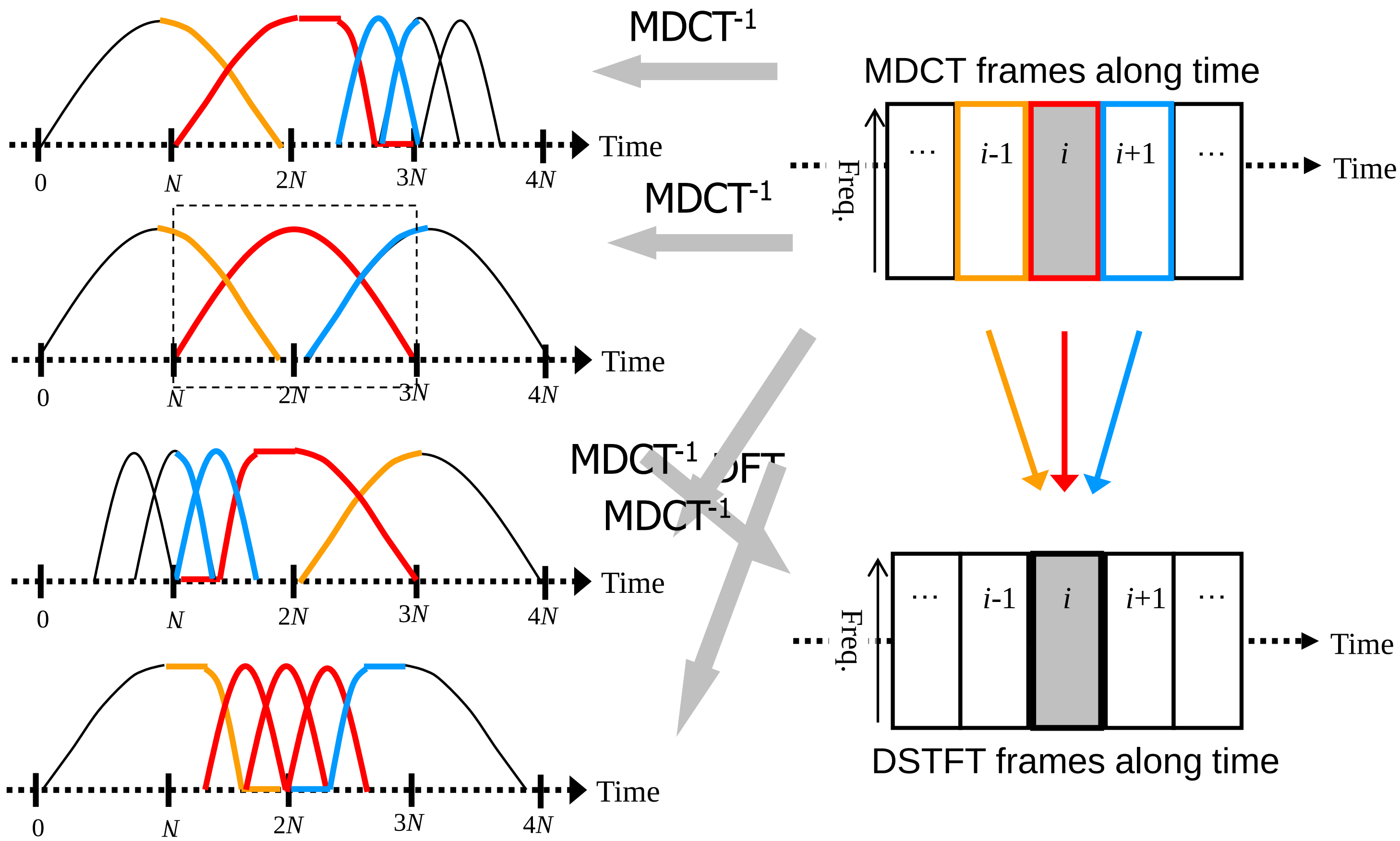






MDCT ↔ DSTFT Conversion 1

4 MDCT windows ➡ 12 conversion expressions



... and 8 more!

MDCT ↔ DSTFT Conversion 2

Solution

A single expression for each conversion direction.

- For example, MDCT→DSTFT Conversion:

$$X_n^{DSTFT}[m] = \sum_{k=0}^{N-1} X_n^{MDCT}[k] \cdot \left(g_d^1[m,k] + (-1)^m \cdot g_r^2[m,k] \right) \qquad ,0 \le m \le N$$

$$+ \sum_{k=0}^{N\text{-}1} X_{n-1}^{MDCT}[k] \cdot g_d^2[m,k] + \sum_{k=0}^{N-1} X_{n+1}^{MDCT}[k] \cdot \left((-1)^m \cdot g_r^1[m,k] \right)$$

- The $g_{d/r}^1[m,k], g_{d/r}^2[m,k]$ functions:
  - Selected according to the window types of frames $X_{n\text{-}1}$, X_n , X_{n+1}.
  - There are 12 such functions, calculated off-line.
  - Each function contains $2N^2$ real values to be stored.

MDCT ↔ DSTFT Conversion 3

- For example:

$$g_d^1\_\mathrm{long}[m,k]=\sum_{n=0}^{N-1}\cos\left[\tfrac{\pi}{N}\cdot\left(n+\tfrac{N+1}{2}\right)\cdot\left(k+\tfrac{1}{2}\right)\right]\cdot h^{long}[n]\cdot w[n]\cdot e^{-j\frac{\pi}{N}\cdot n\cdot m}$$

- Complexity Comparison:

	Efficient conversion	Trivial Conversion*
MDCT ➔ DSTFT	$6N^2$ mults , $8N^2$ adds	$10N^2$ mults , $10N^2$ adds
DSTFT ➔ MDCT	$8N^2$ mults , $20N^2$ adds	$28N^2$ mults , $26N^2$ adds

* FFT cannot be used

APES-based Interpolation Algorithms

- APES: Amplitude and Phase Estimation (Stoica & Li, 1999).
  - An algorithm for spectral estimation.
- GAPES: Gapped-data APES (Stoica & Larsson, 2000).
  - Uses an adaptive filter-bank approach.
- MAPES: Missing-data APES (Stoica & Wang, 2005).
  - Uses an ML- estimator approach.
  - MAPES has lower complexity and can handle more loss patterns.

Comparison to SI

Benefits

- Can handle many loss patterns.
- Doesn't assume parametric modeling.
- Can be applied on complex signals.

Limitations

- High complexity: $O(P^3)+O(P_m^{\,3})$

The APES Algorithm

- Let $\{x_n\}$ be a data-sequence of length P.

<u>Problem</u>

Estimate spectral component at frequency ω_0: $\alpha(\omega_0)$.

<u>Solution</u>

- x_n is modeled as: $x_n = \alpha(\omega_0) \cdot e^{j\omega_0 n} + e_n(\omega \neq \omega_0)$
- Design a narrow-band filter $\underline{h}(\omega_0)$, of length M:
  - The filter should pass the frequency ω_0 without distortion.
  - The filter should attenuate all the other frequencies.
- By filtering $\{x_n\}$ with the filter $\underline{h}(\omega_0)$ we get:

$$h(\omega_0) * x_n \simeq \alpha(\omega_0) \cdot e^{j\omega_0 n}$$

- Use DFT on the filtered data to estimate $\alpha(\omega_0)$.

The APES Algorithm – Cont.

- Matrix notation: create data-snapshot vectors: $\underline{x}_l \in \mathbb{C}^{Mx1}$

$$\{x_1, x_2, x_3, x_4, x_5, \ldots, x_P\} \Rightarrow \begin{pmatrix} x_1 & x_2 & x_3 & & x_{P-2} \\ x_2 & x_3 & x_4 & \cdots & x_{P-1} \\ x_3 & x_4 & x_5 & & x_P \end{pmatrix}$$

M

$\underline{x}_1$ $\underline{x}_2$ $\underline{x}_3$ P-M+1

- Description as a minimization problem:

$$, \text{subject to } \underline{h}^H(\omega_0)\,\underline{a}(\omega_0) = 1$$

Where:

$\underline{h}(\omega_0) \in \mathbb{C}^{Mx1}$ is a data-dependent narrow-band filter, centered at ω_0.

$\underline{a}(\omega) \triangleq [1, e^{j\omega}, \ldots, e^{j\omega(M-1)}]^T \in \mathbb{C}^{Mx1}$ is a vector of exponents.

The GAPES Algorithm

- GAPES interpolates the missing data, assuming it has the same spectral content as the available data.
- APES minimization problem is expanded, using a pre-defined frequency grid: $\{\omega_k\}, 0 \le k \le \mathrm{K}$.
- The **missing samples**, $\{x_m\}$, are restored by solving the following minimization problem:

$$\min_{\{x_m\},\{\alpha_k,\underline{h}_k\}_{k=0}^{K-1}} \sum_{k=0}^{K-1}\sum_{l=0}^{P-M}\left|\underline{h}_k^H\cdot\underline{x}_l-\alpha_k\cdot e^{j\omega_k l}\right|^2, \text{subject to } \underline{h}_k^H\cdot\underline{a}_k=1$$

Where:

$\alpha_k \triangleq \underline{\alpha}(\omega_k)$ is the spectral component at frequency ω_k.

$\underline{h}_k \triangleq \underline{h}(\omega_k)$ is a data-dependent narrow-band filter, centered at ω_k.

$\underline{a}_k \triangleq \underline{a}(\omega_k)$

The GAPES Algorithm – Cont.

- Solved using an iterative algorithm that contains two steps:
  - Minimize with respect to $\{\alpha_k , \underline{h}_k\}$.
  - Then, minimize with respect to the missing samples.
- A single iteration:

Estimate the spectrum
based on the available samples, using the APES algorithm.
(In the first iteration the lost samples are initiated to zero)

Reconstruct the missing samples
so that their spectral content will approximate the estimated spectrum, in the Least-Squares sense.

The APES Algorithm – Different Approach

- APES has also an ML- estimator interpretation.
- Assuming $\{\underline{e}_l(\omega_0)\}$ are statistically-independent, zero-mean complex Gaussian random vectors, with unknown covariance matrix: $\mathbf{Q}(\omega_0)$.
  - APES only approximates an ML- estimator since the vectors contain overlapping data !
- Under these assumptions, the ML- estimator:
- This yields the following minimization problem:

➡ Same solution for $\alpha(\omega_0)$!...

The MAPES Algorithm

- The **missing samples**, $\{x_m\}$, are restored by solving the following maximization problem:

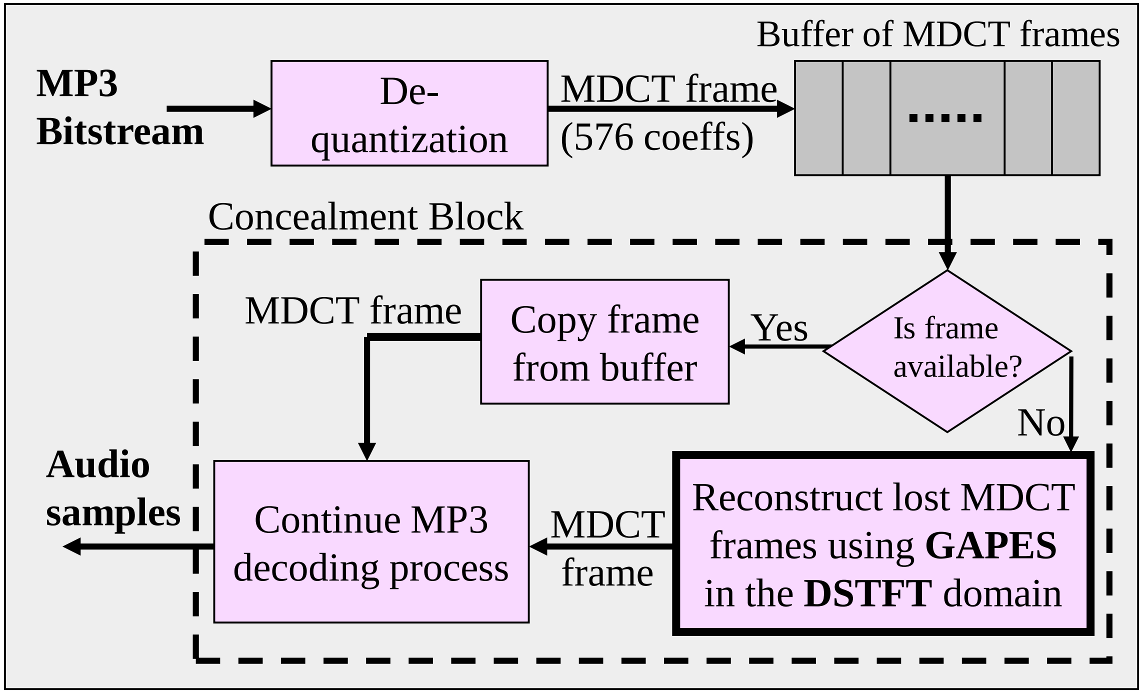
$$\max_{\{x_m\},\{\alpha_k,\mathbf{Q}_k\}_{k=0}^{K-1}} \sum_{k=0}^{K-1} \log\left(Pr\left\{ \left\{ \underline{x}_l \right\}_{l=0}^{P-M} \mid \alpha_k, \mathbf{Q}_k \right\} \right)$$

  Where $\mathbf{Q}_k$ is the covariance-matrix of the $\{\underline{e}_l(\omega_k)\}$ vectors.

- Solved using an iterative algorithm that contains two steps:
  - Solve with respect to $\{\alpha_k , \mathbf{Q}_k\}$ by applying APES.
  - Solve with respect to the missing samples.

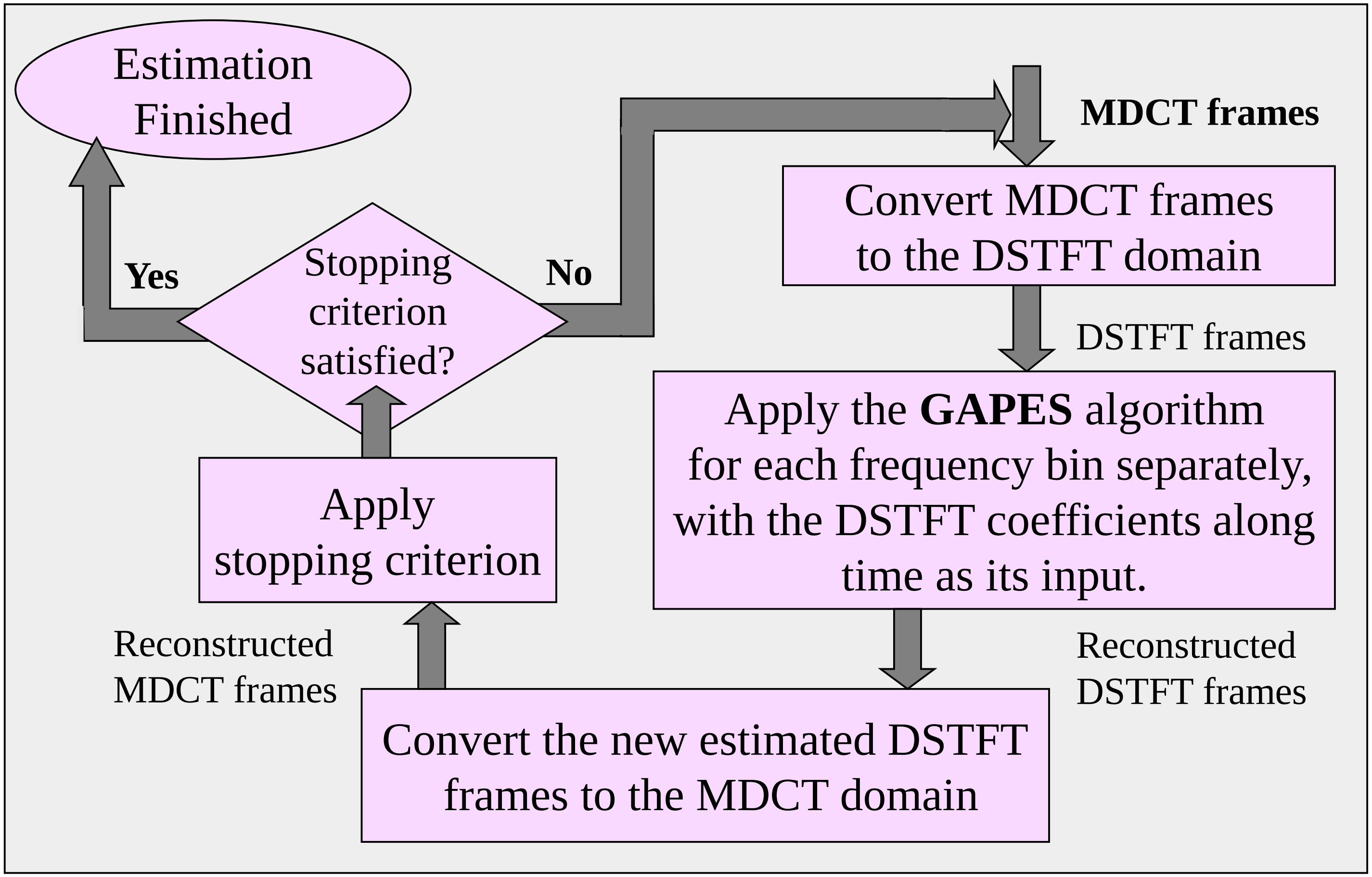
Proposed Concealment Scheme

- An iterative algorithm.
- Reconstructs the lost packets in the DSTFT domain.
- Uses GAPES or MAPES for the interpolation.



The proposed decoding process, including a concealment block

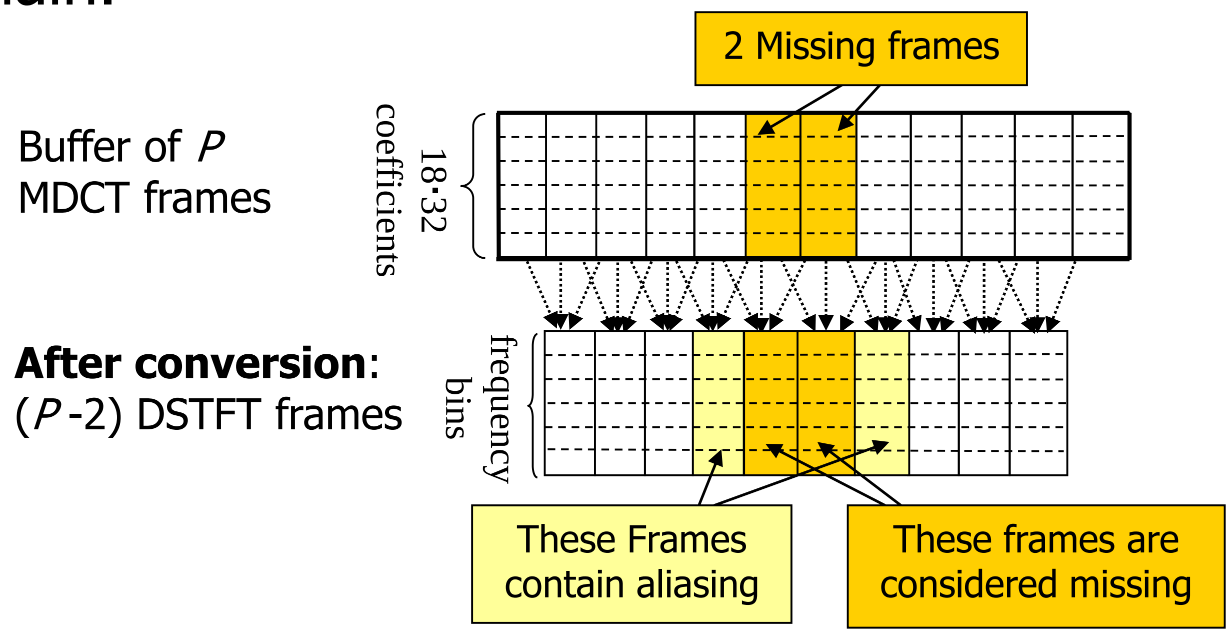
Concealment Algorithm



A block diagram of the reconstruction process

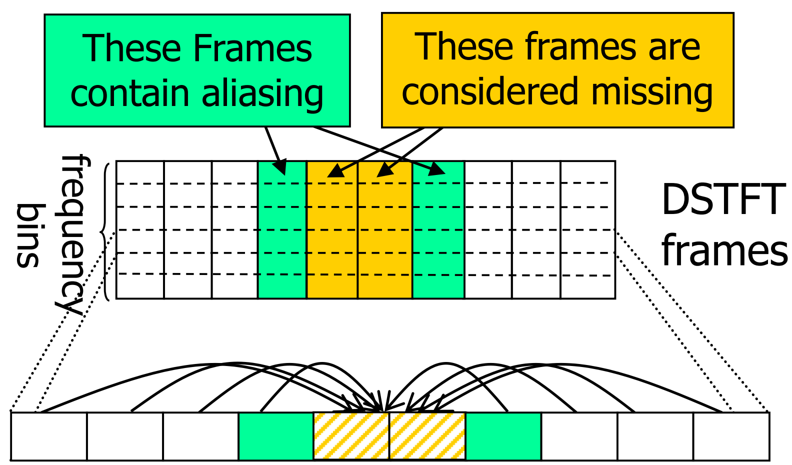
Concealment Algorithm – Example (1)

- Assume that an MP3 packet was lost.
  - 2 MDCT frames are missing in the buffer.
- Stage 1
  Convert the MDCT frames in the buffer to the DSTFT domain.



Concealment Algorithm – Example (2)

- Stage 2
  - Reconstruct the lost coefficients by applying a single iteration of GAPES or MAPES.
  - The algorithm is applied separately on each frequency bin, with the DSTFT coefficients along time as its input.

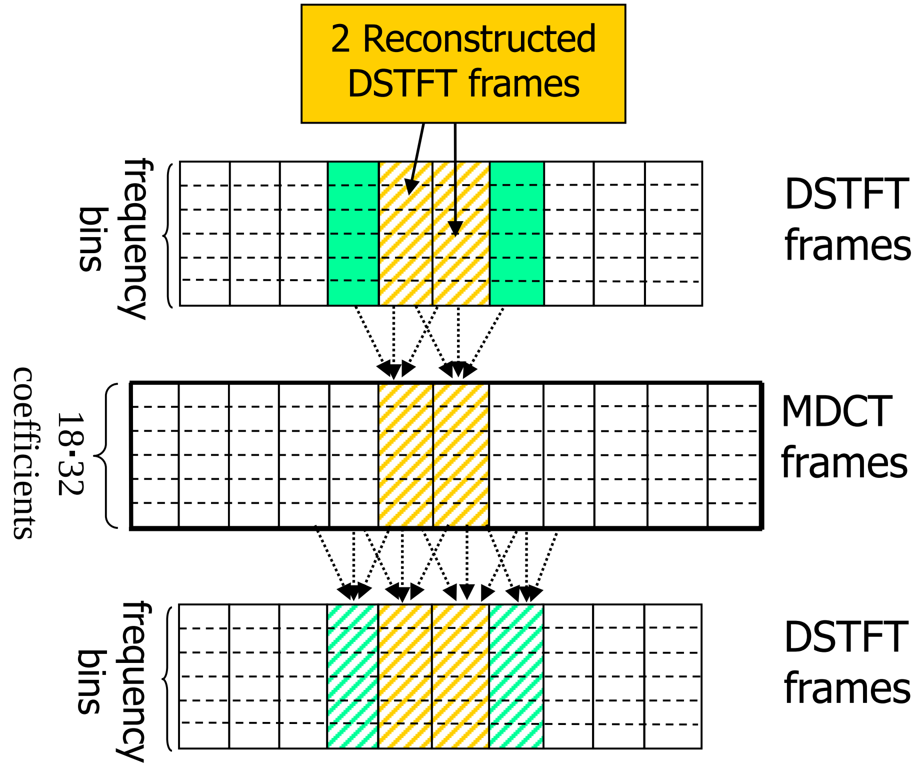


Reconstruct the missing data separately for each bin using GAPES or MAPES.

Concealment Algorithm – Example (3)

- Stage 3
  Convert the estimated DSTFT frames to the MDCT domain:

- Stage 4
  Check stopping criterion:
  - If satisfied – **stop**.
  - If not – Use another iteration.
    (Convert back to the DSTFT domain).



Subjective Tests Results

- The algorithm was evaluated using comparative informal listening tests.
- The listeners had to determine which of the two audio files, each concealed by a different method, sounds better.
- 8-16 inexperienced listeners participated in each test.
- The methods were tested using several music types and at 10% - 30% loss rates at random patterns.

Thanks!!!

To everyone who took part in the listening tests…

Subjective Tests Results 1

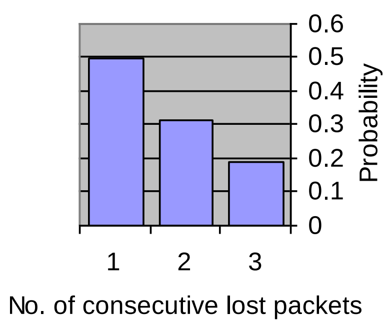
The proposed algorithm is compared to previously reported works:

- Packet Repetition
- Statistical Interpolation

The numbers indicate the distribution of votes in favor of each method.

	GAPES vs. Repetition		GAPES vs. SI		GAPES vs. Original	
Loss rate	GAPES	Repetition	GAPES	SI	GAPES	Original
10%	88.75%	11.25%	97.5%	2.5%	18.75%	81.25%
20%	95%	5%	100%	0%		
30%	88.75%	11.25%	97.5%	2.5%		

Loss histogram - 30% random loss rate



Piano Original

No concealment
(30% random loss)

Repetition

SI

Proposed Algorithm
(GAPES)

Subjective Tests Results 2

Comparison of the use of the two interpolation algorithms:

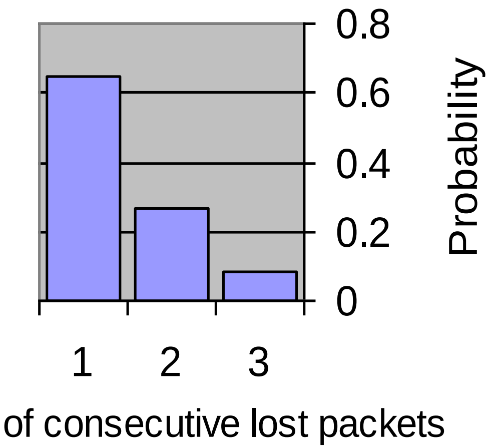
- GAPES
- MAPES

The numbers indicate the distribution

of votes in favor of each method.

Loss rate	GAPES	MAPES
10%	75%	25%
20%	72.5%	27.5%
30%	65%	35%

Loss histogram -
20% random loss rate



Flute Original

MAPES

No concealment
(20% random loss)

GAPES

SI

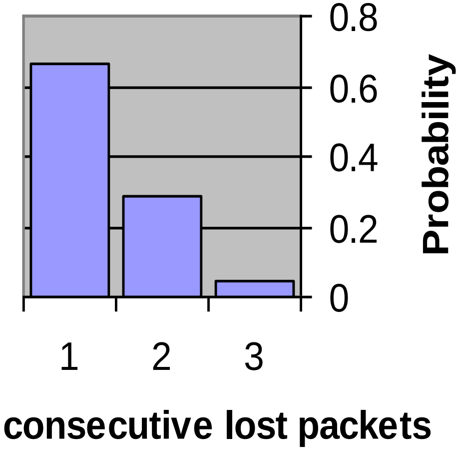
Subjective Tests Results 3

Comparison of the proposed scheme when used in the MDCT domain, versus using it in the DSTFT domain.

The numbers indicate the distribution of votes in favor of each method.

Loss rate	DSTFT	MDCT
10%	70.9%	29.1%
20%	95.9%	4.1%
30%	100%	0%

Loss histogram - 20% random loss rate



Beatles Original

No concealment
(20% random loss)

GAPES (MDCT)

GAPES (DSTFT)

Repetition

Conclusion

- A new algorithm for packet loss concealment.
  - Audio streaming, encoded by MPEG audio coders.
  - Based on applying GAPES or MAPES in the DSTFT domain.
- Comparative informal listening tests were used.
  - The algorithm was tested at 10%-30% loss rates at random loss patterns, using different music types.
  - Performs better than packet repetition and statistical interpolation.
- A direct conversion scheme was introduced: MDCT $\rightleftarrows$ DSTFT.
  - Enables efficient conversion between domains.

Future Directions

- Use psychoacoustic rules in the interpolation process.
  - Set different stopping thresholds to different critical bands, according to SMR.
- Use frequency bins inter-correlation in the interpolation process.
- Add side information at the encoder to be used in the concealment process, such as:
  - The sign of the MDCT coefficients.
  - Polynomial approximation of the MDCT coefficients.
- Make proper adjustments in order to implement the proposed algorithm on a Real-Time platform.
  - Numeric shortcuts (matrix calculations).
  - Use a sub-optimal version of GAPES.