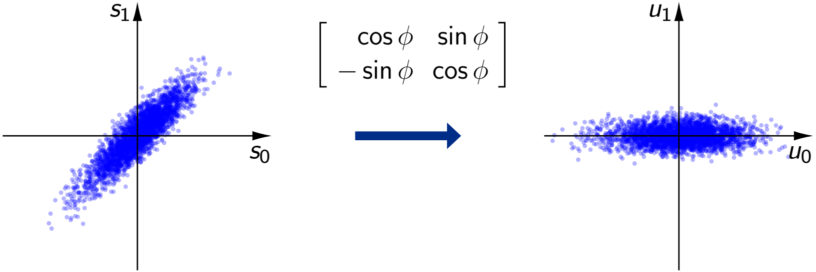
Transform Coding



Last Lectures: Scalar and Vector Quantization

Scalar Quantization

- Simple encoding and decoding procedure
- Uniform reconstruction quantizers (URQ): Particularly simple and still very efficient
- Cannot exploit statistical dependencies (would require very complex entropy coding)

Last Lectures: Scalar and Vector Quantization

Scalar Quantization

- Simple encoding and decoding procedure
- Uniform reconstruction quantizers (URQ): Particularly simple and still very efficient
- Cannot exploit statistical dependencies (would require very complex entropy coding)

Vector Quantization

- High-dimensional vector quantizers can approach rate-distortion bound
- Space-filling gain can only be exploited by vector quantization (1.53 dB for $N \to \infty$)
- Rarely used in practice: High computational complexity and memory requirements

Last Lectures: Scalar and Vector Quantization

Scalar Quantization

- Simple encoding and decoding procedure
- Uniform reconstruction quantizers (URQ): Particularly simple and still very efficient
- Cannot exploit statistical dependencies (would require very complex entropy coding)

Vector Quantization

- High-dimensional vector quantizers can approach rate-distortion bound
- Space-filling gain can only be exploited by vector quantization ($1.53\,\text{dB}$ for $N \to \infty$)
- Rarely used in practice: High computational complexity and memory requirements

Lossy Coding of Sources with Memory

- Most important aspect: Exploit statistical dependencies (memory advantage)
- Need approach that is simpler than vector quantization, but still efficient

Transform Coding: Introduction

Transform Coding

- Simple concept for exploiting linear dependencies between samples
- Low complexity compared to vector quantization (can be interpreted as very simple VQ)
- **Used in virtually all lossy audio, image and video codecs**

Transform Coding: Introduction

Transform Coding

- Simple concept for exploiting linear dependencies between samples
- Low complexity compared to vector quantization (can be interpreted as very simple VQ)
- **Used in virtually all lossy audio, image and video codecs**

Basic Concept

1. Arrange samples into blocks/vectors $\boldsymbol{s}$ of N adjacent samples

Transform Coding: Introduction

Transform Coding

- Simple concept for exploiting linear dependencies between samples
- Low complexity compared to vector quantization (can be interpreted as very simple VQ)
- **Used in virtually all lossy audio, image and video codecs**

Basic Concept

1. Arrange samples into blocks/vectors $\boldsymbol{s}$ of N adjacent samples
2. **Analysis transform**: Mapping to vectors of transform coefficients

$$\boldsymbol{u} = \boldsymbol{A} \cdot \boldsymbol{s}$$

Transform Coding: Introduction

Transform Coding

- Simple concept for exploiting linear dependencies between samples
- Low complexity compared to vector quantization (can be interpreted as very simple VQ)
- **Used in virtually all lossy audio, image and video codecs**

Basic Concept

1. Arrange samples into blocks/vectors $\boldsymbol{s}$ of N adjacent samples
2. **Analysis transform**: Mapping to vectors of transform coefficients

$$\boldsymbol{u} = \boldsymbol{A} \cdot \boldsymbol{s}$$

3. **Scalar quantization** of transform coefficients $\boldsymbol{u} = \{u_k\}$

$$u_k \mapsto u'_k$$

Transform Coding: Introduction

Transform Coding

- Simple concept for exploiting linear dependencies between samples
- Low complexity compared to vector quantization (can be interpreted as very simple VQ)
- **Used in virtually all lossy audio, image and video codecs**

Basic Concept

1. Arrange samples into blocks/vectors $\boldsymbol{s}$ of N adjacent samples
2. **Analysis transform**: Mapping to vectors of transform coefficients

$$\boldsymbol{u} = \boldsymbol{A} \cdot \boldsymbol{s}$$

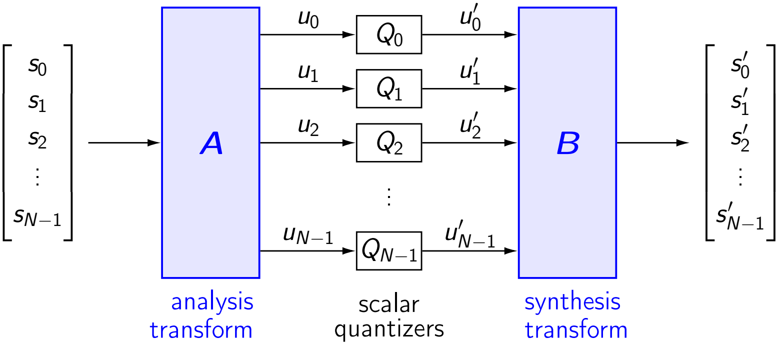
3. **Scalar quantization** of transform coefficients $\boldsymbol{u} = \{u_k\}$

$$u_k \mapsto u'_k$$

4. **Synthesis transform**: Mapping to blocks/vectors of reconstructed samples

$$\boldsymbol{s}' = \boldsymbol{B} \cdot \boldsymbol{u}'$$

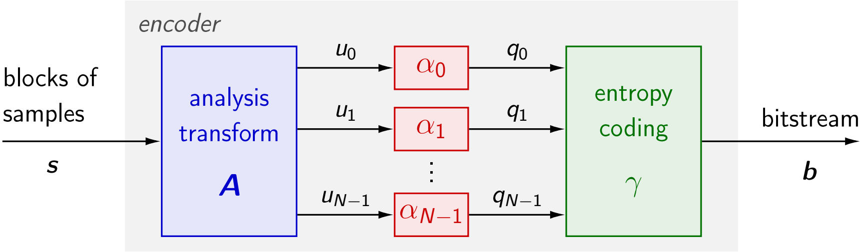
Structure of Transform Coding Systems



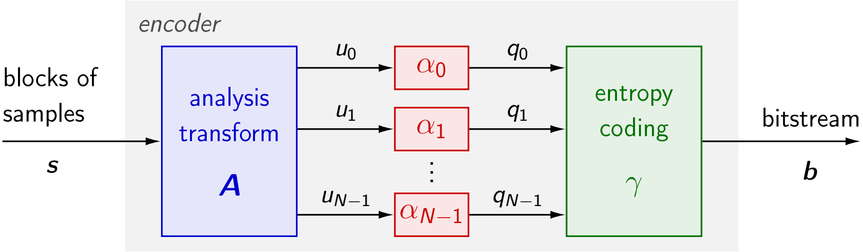
Effect of transform coding:

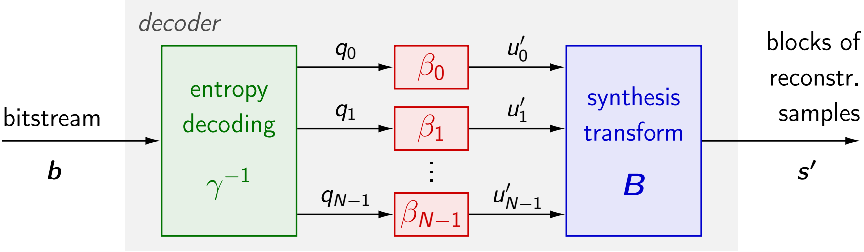
- Remove/reduce dependencies before scalar quantization
- Simple alternative to vector quantization
- Simple and most relevant case: Linear transforms

Transform Encoder and Transform Decoder



Transform Encoder and Transform Decoder





Motivation of Transform Coding

Exploitation of Statistical Dependencies

- Typically, the signal energy is concentrated in a few transform coefficients
- Coding of a few non-zero coefficients and many zero-valued coefficients can be very efficient (e.g., using arithmetic coding, run-level coding, ...)

➡ Scalar quantization is more effective in transform domain

Motivation of Transform Coding

Exploitation of Statistical Dependencies

- Typically, the signal energy is concentrated in a few transform coefficients
- Coding of a few non-zero coefficients and many zero-valued coefficients can be very efficient (e.g., using arithmetic coding, run-level coding, ...)

➡ Scalar quantization is more effective in transform domain

Efficient trade-off between Coding Efficiency and Complexity

- Vector Quantization: Searching through codebook for best matching vector

➡ Transform and scalar quantization: Substantial reduction in complexity

Motivation of Transform Coding

Exploitation of Statistical Dependencies

- Typically, the signal energy is concentrated in a few transform coefficients
- Coding of a few non-zero coefficients and many zero-valued coefficients can be very efficient (e.g., using arithmetic coding, run-level coding, ...)

➡ Scalar quantization is more effective in transform domain

Efficient trade-off between Coding Efficiency and Complexity

- Vector Quantization: Searching through codebook for best matching vector

➡ Transform and scalar quantization: Substantial reduction in complexity

Suitable for Quantization using Perceptual Criteria

- Speech & audio coding: Frequency bands might be used to simulate processing of human ear
- Image & video coding: Quantization in transform domain leads to subjective improvement

➡ Removal of perceptually irrelevant signal components

Linear Block Transforms: Analysis Transform

Linear Block Transform

- Each component of the N-dimensional output vector $\boldsymbol{u}$ represents a linear combination of the N components of the N-dimensional input vector $\boldsymbol{s}$

➡ Can be represented as matrix multiplication

Linear Block Transforms: Analysis Transform

Linear Block Transform

- Each component of the N-dimensional output vector $\boldsymbol{u}$ represents a linear combination of the N components of the N-dimensional input vector $\boldsymbol{s}$

➡ Can be represented as matrix multiplication

Linear Analysis Transform

- Block of samples $\boldsymbol{s}$ is converted into vector of transform coefficients $\boldsymbol{u}$

$$\boldsymbol{u} = \boldsymbol{A} \cdot \boldsymbol{s}$$

- Extended notation

$$\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} & & \\ & \boldsymbol{A} & \\ & & \end{bmatrix} \cdot \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{N-1} \end{bmatrix}$$

Linear Block Transforms: Synthesis Transform

Linear Synthesis Transform

- Vector of reconstructed transform coefficients $\boldsymbol{u}'$ is converted into block of reconstructed samples $\boldsymbol{s}'$

$$\boldsymbol{s}' = \boldsymbol{B} \cdot \boldsymbol{u}'$$

- Extended notation

$$\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ \vdots \\ s'_{N-1} \end{bmatrix} = \begin{bmatrix} & & \\ & \boldsymbol{B} & \\ & & \end{bmatrix} \cdot \begin{bmatrix} u'_0 \\ u'_1 \\ u'_2 \\ \vdots \\ u'_{N-1} \end{bmatrix}$$

Linear Block Transforms: Synthesis Transform

Linear Synthesis Transform

- Vector of reconstructed transform coefficients $\boldsymbol{u}'$ is converted into block of reconstructed samples $\boldsymbol{s}'$

$$\boldsymbol{s}' = \boldsymbol{B} \cdot \boldsymbol{u}'$$

- Extended notation

$$\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ \vdots \\ s'_{N-1} \end{bmatrix} = \begin{bmatrix} & & \\ & \boldsymbol{B} & \\ & & \end{bmatrix} \cdot \begin{bmatrix} u'_0 \\ u'_1 \\ u'_2 \\ \vdots \\ u'_{N-1} \end{bmatrix}$$

➡ Interpretation: Vector of reconstructed samples $\boldsymbol{s}'$ is represented as a linear combination of column vectors $\{\boldsymbol{b}_k\}$ of the synthesis matrix $\boldsymbol{B}$

$$\boldsymbol{s}' = u'_0 \cdot \boldsymbol{b}_0 + u'_1 \cdot \boldsymbol{b}_1 + u'_2 \cdot \boldsymbol{b}_1 + \ldots + u'_{N-1} \cdot \boldsymbol{b}_{N-1}$$

Interpretation of Synthesis Transform

Synthesis Transform

- Reconstructed block of samples $\boldsymbol{s'}$

$$\underbrace{\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ \vdots \\ s'_{N-1} \end{bmatrix}}_{\boldsymbol{s'}} = u'_0 \cdot \underbrace{\begin{bmatrix} b_{00} \\ b_{01} \\ b_{02} \\ \vdots \\ b_{0,N-1} \end{bmatrix}}_{\boldsymbol{b}_0} + u'_1 \cdot \underbrace{\begin{bmatrix} b_{10} \\ b_{11} \\ b_{12} \\ \vdots \\ b_{1,N-1} \end{bmatrix}}_{\boldsymbol{b}_1} + u'_2 \cdot \underbrace{\begin{bmatrix} b_{20} \\ b_{21} \\ b_{22} \\ \vdots \\ b_{2,N-1} \end{bmatrix}}_{\boldsymbol{b}_2} + \cdots$$

Interpretation of Synthesis Transform

Synthesis Transform

- Reconstructed block of samples $\boldsymbol{s'}$

$$\underbrace{\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ \vdots \\ s'_{N-1} \end{bmatrix}}_{\boldsymbol{s'}} = u'_0 \cdot \underbrace{\begin{bmatrix} b_{00} \\ b_{01} \\ b_{02} \\ \vdots \\ b_{0,N-1} \end{bmatrix}}_{\boldsymbol{b}_0} + u'_1 \cdot \underbrace{\begin{bmatrix} b_{10} \\ b_{11} \\ b_{12} \\ \vdots \\ b_{1,N-1} \end{bmatrix}}_{\boldsymbol{b}_1} + u'_2 \cdot \underbrace{\begin{bmatrix} b_{20} \\ b_{21} \\ b_{22} \\ \vdots \\ b_{2,N-1} \end{bmatrix}}_{\boldsymbol{b}_2} + \cdots$$

➡ Reconstructed transform coefficients $\{u'_k\}$ represent weighting factors for basis vectors $\{\boldsymbol{b}_k\}$ (i.e., columns) of synthesis transform matrix $\boldsymbol{B}$

Interpretation of Synthesis Transform

Synthesis Transform

- Reconstructed block of samples $\boldsymbol{s'}$

$$\underbrace{\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ \vdots \\ s'_{N-1} \end{bmatrix}}_{\boldsymbol{s'}} = u'_0 \cdot \underbrace{\begin{bmatrix} b_{00} \\ b_{01} \\ b_{02} \\ \vdots \\ b_{0,N-1} \end{bmatrix}}_{\boldsymbol{b}_0} + u'_1 \cdot \underbrace{\begin{bmatrix} b_{10} \\ b_{11} \\ b_{12} \\ \vdots \\ b_{1,N-1} \end{bmatrix}}_{\boldsymbol{b}_1} + u'_2 \cdot \underbrace{\begin{bmatrix} b_{20} \\ b_{21} \\ b_{22} \\ \vdots \\ b_{2,N-1} \end{bmatrix}}_{\boldsymbol{b}_2} + \cdots$$

➡ Reconstructed transform coefficients $\{u'_k\}$ represent weighting factors for basis vectors $\{\boldsymbol{b}_k\}$ (i.e., columns) of synthesis transform matrix $\boldsymbol{B}$

Analysis Transform for most relevant case $\boldsymbol{A} = \boldsymbol{B}^{-1}$

➡ Decomposition of sample vector $\boldsymbol{s}$ into basis vectors $\{\boldsymbol{b}_k\}$

➡ Transform coefficients u_k represent the corresponding weighting factors

Example for Possible Basis Vectors (of size 4)

$$\boldsymbol{b}_0 = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{b}_1 = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \qquad \boldsymbol{b}_2 = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{b}_3 = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

Example for Possible Basis Vectors (of size 4)

$$\boldsymbol{b}_0 = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{b}_1 = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \qquad \boldsymbol{b}_2 = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{b}_3 = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

➡ Synthesis matrix $\boldsymbol{B}$

$$\boldsymbol{B} = \begin{bmatrix} | & | & | & | \\ \boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{b}_3 \\ | & | & | & | \end{bmatrix}$$

Example for Possible Basis Vectors (of size 4)

$$\boldsymbol{b}_0 = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{b}_1 = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \qquad \boldsymbol{b}_2 = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{b}_3 = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

➡ Synthesis matrix $\boldsymbol{B}$

$$\boldsymbol{B} = \begin{bmatrix} | & | & | & | \\ \boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{b}_3 \\ | & | & | & | \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

Example for Possible Basis Vectors (of size 4)

$$\boldsymbol{b}_0 = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{b}_1 = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \quad \boldsymbol{b}_2 = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}, \quad \boldsymbol{b}_3 = \frac{1}{4}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

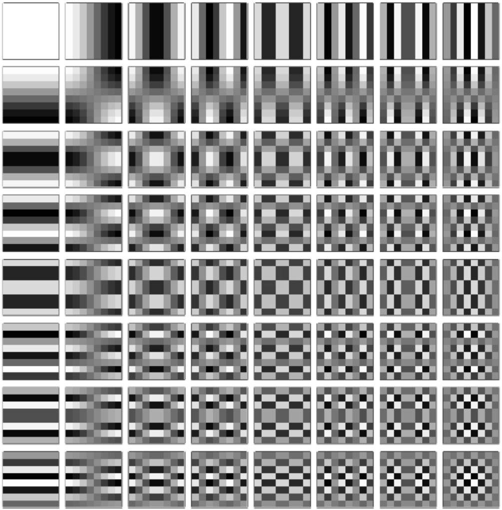
➡ Synthesis matrix $\boldsymbol{B}$

$$\boldsymbol{B} = \begin{bmatrix} | & | & | & | \\ \boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{b}_3 \\ | & | & | & | \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

➡ Associated analysis matrix $\boldsymbol{A}$ (typical choice)

$$\boldsymbol{A} = \boldsymbol{B}^{-1} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

Example: Typical Basis Functions for 8×8 Image Blocks



Perfect Reconstruction Property

Without Quantization

- Transform coefficients are lossless coded: $\boldsymbol{u}' = \boldsymbol{u}$

➡ Optimal synthesis transform: $\boldsymbol{B} = \boldsymbol{A}^{-1}$

Perfect Reconstruction Property

Without Quantization

- Transform coefficients are lossless coded: $\boldsymbol{u}' = \boldsymbol{u}$
- ➡ Optimal synthesis transform: $\boldsymbol{B} = \boldsymbol{A}^{-1}$
- ➡ Reconstructed samples are equal to source samples

$$\boldsymbol{s}' = \boldsymbol{B}\,\boldsymbol{u} = \boldsymbol{B}\,\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{A}^{-1}\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{s}$$

Perfect Reconstruction Property

Without Quantization

- Transform coefficients are lossless coded: $\boldsymbol{u}' = \boldsymbol{u}$

➡ Optimal synthesis transform: $\boldsymbol{B} = \boldsymbol{A}^{-1}$

➡ Reconstructed samples are equal to source samples

$$\boldsymbol{s}' = \boldsymbol{B}\,\boldsymbol{u} = \boldsymbol{B}\,\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{A}^{-1}\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{s}$$

Optimal Synthesis Transform (in presence of quantization)

- Optimality: Minimum MSE distortion among all synthesis transforms for given analysis transform $\boldsymbol{A}$

Perfect Reconstruction Property

Without Quantization

- Transform coefficients are lossless coded: $\boldsymbol{u}' = \boldsymbol{u}$

➡ Optimal synthesis transform: $\boldsymbol{B} = \boldsymbol{A}^{-1}$

➡ Reconstructed samples are equal to source samples

$$\boldsymbol{s}' = \boldsymbol{B}\,\boldsymbol{u} = \boldsymbol{B}\,\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{A}^{-1}\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{s}$$

Optimal Synthesis Transform (in presence of quantization)

- Optimality: Minimum MSE distortion among all synthesis transforms for given analysis transform $\boldsymbol{A}$
- $\boldsymbol{B} = \boldsymbol{A}^{-1}$ is optimal if
  - $\boldsymbol{A}$ is invertible and produces independent transform coefficients
  - all component quantizers are centroid quantizers

Perfect Reconstruction Property

Without Quantization

- Transform coefficients are lossless coded: $\boldsymbol{u}' = \boldsymbol{u}$
- ➡ Optimal synthesis transform: $\boldsymbol{B} = \boldsymbol{A}^{-1}$
- ➡ Reconstructed samples are equal to source samples

$$\boldsymbol{s}' = \boldsymbol{B}\,\boldsymbol{u} = \boldsymbol{B}\,\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{A}^{-1}\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{s}$$

Optimal Synthesis Transform (in presence of quantization)

- Optimality: Minimum MSE distortion among all synthesis transforms for given analysis transform $\boldsymbol{A}$
- $\boldsymbol{B} = \boldsymbol{A}^{-1}$ is optimal if
  - $\boldsymbol{A}$ is invertible and produces independent transform coefficients
  - all component quantizers are centroid quantizers
- If above conditions are not fulfilled, a synthesis transform $\boldsymbol{B} \neq \boldsymbol{A}^{-1}$ may reduce the distortion

Perfect Reconstruction Property

Without Quantization

- Transform coefficients are lossless coded: $\boldsymbol{u}' = \boldsymbol{u}$
- ➡ Optimal synthesis transform: $\boldsymbol{B} = \boldsymbol{A}^{-1}$
- ➡ Reconstructed samples are equal to source samples

$$\boldsymbol{s}' = \boldsymbol{B}\,\boldsymbol{u} = \boldsymbol{B}\,\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{A}^{-1}\boldsymbol{A}\,\boldsymbol{s} = \boldsymbol{s}$$

Optimal Synthesis Transform (in presence of quantization)

- Optimality: Minimum MSE distortion among all synthesis transforms for given analysis transform $\boldsymbol{A}$
- $\boldsymbol{B} = \boldsymbol{A}^{-1}$ is optimal if
  - $\boldsymbol{A}$ is invertible and produces independent transform coefficients
  - all component quantizers are centroid quantizers
- If above conditions are not fulfilled, a synthesis transform $\boldsymbol{B} \neq \boldsymbol{A}^{-1}$ may reduce the distortion

➡ **In Practice: Use linear transforms with $\boldsymbol{B} = \boldsymbol{A}^{-1}$**

Unitary Transforms

Unitary Matrix

- Inverse matrix is equal to its conjugate transpose

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{\dagger} = (\boldsymbol{A}^{*})^{\mathrm{T}}$$

Unitary Transforms

Unitary Matrix

- Inverse matrix is equal to its conjugate transpose

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{\dagger} = \left(\boldsymbol{A}^{*}\right)^{\mathrm{T}}$$

➡ Unitary transforms preserve length of vectors: $\|\boldsymbol{A} \cdot \boldsymbol{s}\|_2 = \|\boldsymbol{s}\|_2$

$$\|\boldsymbol{u}\|_2^2 = \sum_k |u_k|^2 = \sum_k u_k^* \cdot u_k = \left(\boldsymbol{u}^*\right)^{\mathrm{T}} \boldsymbol{u}$$

Unitary Transforms

Unitary Matrix

- Inverse matrix is equal to its conjugate transpose

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{\dagger} = \left(\boldsymbol{A}^{*}\right)^{\mathrm{T}}$$

➡ Unitary transforms preserve length of vectors: $\|\boldsymbol{A} \cdot \boldsymbol{s}\|_2 = \|\boldsymbol{s}\|_2$

$$\begin{aligned} \|\boldsymbol{u}\|_2^2 &= \sum_k |u_k|^2 = \sum_k u_k^* \cdot u_k = \left(\boldsymbol{u}^*\right)^{\mathrm{T}} \boldsymbol{u} \\ &= \boldsymbol{u}^{\dagger} \cdot \boldsymbol{u} = (\boldsymbol{A}\boldsymbol{s})^{\dagger} \cdot (\boldsymbol{A}\boldsymbol{s}) = \boldsymbol{s}^{\dagger} \cdot \boldsymbol{A}^{\dagger} \cdot \boldsymbol{A} \cdot \boldsymbol{s} \end{aligned}$$

Unitary Transforms

Unitary Matrix

- Inverse matrix is equal to its conjugate transpose

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{\dagger} = (\boldsymbol{A}^{*})^{\mathrm{T}}$$

➡ Unitary transforms preserve length of vectors: $\|\boldsymbol{A} \cdot \boldsymbol{s}\|_2 = \|\boldsymbol{s}\|_2$

$$\begin{aligned} \|\boldsymbol{u}\|_2^2 &= \sum_k |u_k|^2 = \sum_k u_k^* \cdot u_k = (\boldsymbol{u}^*)^{\mathrm{T}} \boldsymbol{u} \\ &= \boldsymbol{u}^{\dagger} \cdot \boldsymbol{u} = (\boldsymbol{A}\boldsymbol{s})^{\dagger} \cdot (\boldsymbol{A}\boldsymbol{s}) = \boldsymbol{s}^{\dagger} \cdot \boldsymbol{A}^{\dagger} \cdot \boldsymbol{A} \cdot \boldsymbol{s} \\ &= \boldsymbol{s}^{\dagger} \cdot \left(\boldsymbol{A}^{-1} \cdot \boldsymbol{A}\right) \cdot \boldsymbol{s} = \boldsymbol{s}^{\dagger} \cdot \boldsymbol{s} \end{aligned}$$

Unitary Transforms

Unitary Matrix

- Inverse matrix is equal to its conjugate transpose

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{\dagger} = \left(\boldsymbol{A}^{*}\right)^{\mathrm{T}}$$

➡ Unitary transforms preserve length of vectors: $\|\boldsymbol{A} \cdot \boldsymbol{s}\|_2 = \|\boldsymbol{s}\|_2$

$$\begin{aligned}
\|\boldsymbol{u}\|_2^2 &= \sum_k |u_k|^2 = \sum_k u_k^* \cdot u_k = \left(\boldsymbol{u}^*\right)^{\mathrm{T}} \boldsymbol{u} \\
&= \boldsymbol{u}^{\dagger} \cdot \boldsymbol{u} = (\boldsymbol{A}\boldsymbol{s})^{\dagger} \cdot (\boldsymbol{A}\boldsymbol{s}) = \boldsymbol{s}^{\dagger} \cdot \boldsymbol{A}^{\dagger} \cdot \boldsymbol{A} \cdot \boldsymbol{s} \\
&= \boldsymbol{s}^{\dagger} \cdot \left(\boldsymbol{A}^{-1} \cdot \boldsymbol{A}\right) \cdot \boldsymbol{s} = \boldsymbol{s}^{\dagger} \cdot \boldsymbol{s} \\
&= \sum_k s_k^* \cdot s_k = \sum_k |s_k|^2
\end{aligned}$$

Unitary Transforms

Unitary Matrix

- Inverse matrix is equal to its conjugate transpose

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{\dagger} = \left(\boldsymbol{A}^{*}\right)^{\mathrm{T}}$$

➡ Unitary transforms preserve length of vectors: $\|\boldsymbol{A}\cdot\boldsymbol{s}\|_2 = \|\boldsymbol{s}\|_2$

$$\begin{aligned}
\|\boldsymbol{u}\|_2^2 &= \sum_k |u_k|^2 = \sum_k u_k^* \cdot u_k = \left(\boldsymbol{u}^*\right)^{\mathrm{T}} \boldsymbol{u} \\
&= \boldsymbol{u}^{\dagger}\cdot\boldsymbol{u} = (\boldsymbol{A}\boldsymbol{s})^{\dagger}\cdot(\boldsymbol{A}\boldsymbol{s}) = \boldsymbol{s}^{\dagger}\cdot\boldsymbol{A}^{\dagger}\cdot\boldsymbol{A}\cdot\boldsymbol{s} \\
&= \boldsymbol{s}^{\dagger}\cdot\left(\boldsymbol{A}^{-1}\cdot\boldsymbol{A}\right)\cdot\boldsymbol{s} = \boldsymbol{s}^{\dagger}\cdot\boldsymbol{s} \\
&= \sum_k s_k^*\cdot s_k = \sum_k |s_k|^2 \\
&= \|\boldsymbol{s}\|_2^2
\end{aligned}$$

Orthogonal Transforms

Orthogonal Matrix

- Special case of unitary matrix: All matrix elements are real values

➡ Inverse matrix is equal to the transpose

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{\mathrm{T}}$$

Orthogonal Transforms

Orthogonal Matrix

- Special case of unitary matrix: All matrix elements are real values

➡ Inverse matrix is equal to the transpose

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{\mathrm{T}}$$

Basis Vectors

- Columns of synthesis matrix $\boldsymbol{B}$

➡ Rows of analysis matrix $\boldsymbol{A} = \boldsymbol{B}^{\mathrm{T}}$

$$\boldsymbol{A} = \begin{bmatrix} \text{——} & \boldsymbol{b}_0 & \text{——} \\ \text{——} & \boldsymbol{b}_1 & \text{——} \\ \text{——} & \boldsymbol{b}_2 & \text{——} \\ & \vdots & \\ \text{——} & \boldsymbol{b}_{N-1} & \text{——} \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} | & | & | & & | \\ \boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_{N-1} \\ | & | & | & & | \end{bmatrix}$$

Orthonormal Basis

Property of Unitary Transforms

- Consider product of analysis and synthesis matrix: $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}^{-1}\boldsymbol{B} = \boldsymbol{B}^{\dagger}\boldsymbol{B}$

$$\begin{bmatrix} \text{---} & \boldsymbol{b}_0^* & \text{---} \\ \text{---} & \boldsymbol{b}_1^* & \text{---} \\ \text{---} & \boldsymbol{b}_2^* & \text{---} \\ & \vdots & \\ \text{---} & \boldsymbol{b}_{N-1}^* & \text{---} \end{bmatrix} \begin{bmatrix} | & | & | & & | \\ \boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_{N-1} \\ | & | & | & & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Orthonormal Basis

Property of Unitary Transforms

- Consider product of analysis and synthesis matrix: $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}^{-1}\boldsymbol{B} = \boldsymbol{B}^{\dagger}\boldsymbol{B}$

$$\begin{bmatrix} — & \boldsymbol{b}_0^* & — \\ — & \boldsymbol{b}_1^* & — \\ — & \boldsymbol{b}_2^* & — \\ & \vdots & \\ — & \boldsymbol{b}_{N-1}^* & — \end{bmatrix} \begin{bmatrix} | & | & | & & | \\ \boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_{N-1} \\ | & | & | & & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

➡ Basis vectors $\boldsymbol{b}_k$ are orthogonal to each other

Orthonormal Basis

Property of Unitary Transforms

- Consider product of analysis and synthesis matrix: $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}^{-1}\boldsymbol{B} = \boldsymbol{B}^{\dagger}\boldsymbol{B}$

$$\begin{bmatrix} \text{---} & \boldsymbol{b}_0^* & \text{---} \\ \text{---} & \boldsymbol{b}_1^* & \text{---} \\ \text{---} & \boldsymbol{b}_2^* & \text{---} \\ & \vdots & \\ \text{---} & \boldsymbol{b}_{N-1}^* & \text{---} \end{bmatrix} \begin{bmatrix} | & | & | & & | \\ \boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_{N-1} \\ | & | & | & & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

➡ Basis vectors $\boldsymbol{b}_k$ are orthogonal to each other

➡ Basis vectors $\boldsymbol{b}_k$ have a length equal to 1

Orthonormal Basis

Property of Unitary Transforms

- Consider product of analysis and synthesis matrix: $\boldsymbol{AB} = \boldsymbol{B}^{-1}\boldsymbol{B} = \boldsymbol{B}^{\dagger}\boldsymbol{B}$

$$\begin{bmatrix} \text{---} & \boldsymbol{b}_0^* & \text{---} \\ \text{---} & \boldsymbol{b}_1^* & \text{---} \\ \text{---} & \boldsymbol{b}_2^* & \text{---} \\ & \vdots & \\ \text{---} & \boldsymbol{b}_{N-1}^* & \text{---} \end{bmatrix} \begin{bmatrix} | & | & | & & | \\ \boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_{N-1} \\ | & | & | & & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

➡ Basis vectors $\boldsymbol{b}_k$ are orthogonal to each other

➡ Basis vectors $\boldsymbol{b}_k$ have a length equal to 1

➡ **Basis vectors of unitary matrices form an orthonormal basis**

Orthonormal Basis

Property of Unitary Transforms

- Consider product of analysis and synthesis matrix: $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}^{-1}\boldsymbol{B} = \boldsymbol{B}^{\dagger}\boldsymbol{B}$

$$\begin{bmatrix} \text{---} & \boldsymbol{b}_0^* & \text{---} \\ \text{---} & \boldsymbol{b}_1^* & \text{---} \\ \text{---} & \boldsymbol{b}_2^* & \text{---} \\ & \vdots & \\ \text{---} & \boldsymbol{b}_{N-1}^* & \text{---} \end{bmatrix} \begin{bmatrix} | & | & | & & | \\ \boldsymbol{b}_0 & \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_{N-1} \\ | & | & | & & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

➡ Basis vectors $\boldsymbol{b}_k$ are orthogonal to each other

➡ Basis vectors $\boldsymbol{b}_k$ have a length equal to 1

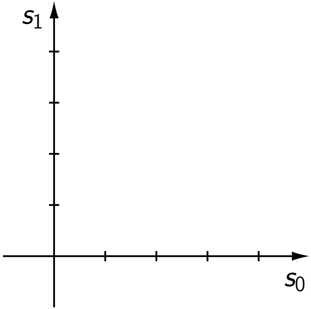
➡ **Basis vectors of unitary matrices form an orthonormal basis**

Geometric Interpretation

- Rotation (and possible reflection) of coordinate system

Example of Orthogonal Transform for $N = 2$

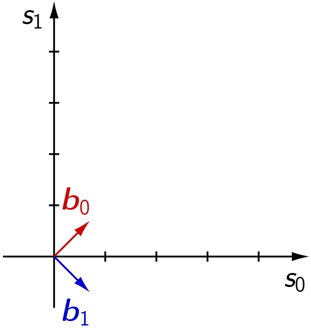
- Vector of two samples $\boldsymbol{s} = (s_0, s_1)^{\mathrm{T}}$



Example of Orthogonal Transform for $N = 2$

- Vector of two samples $\boldsymbol{s} = (s_0, s_1)^{\mathrm{T}}$
- Synthesis transform matrix

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_0 & \boldsymbol{b}_1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



Example of Orthogonal Transform for $N = 2$

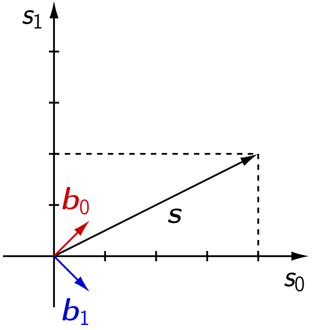
- Vector of two samples $\boldsymbol{s} = (s_0, s_1)^{\mathrm{T}}$
- Synthesis transform matrix

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_0 & \boldsymbol{b}_1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- Representation of signal vector

$$\boldsymbol{s} = u_0 \cdot \boldsymbol{b}_0 + u_1 \cdot \boldsymbol{b}_1$$

$$\begin{bmatrix} 4 \\ 2 \end{bmatrix} = u_0 \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + u_1 \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



Example of Orthogonal Transform for $N = 2$

- Vector of two samples $\boldsymbol{s} = (s_0, s_1)^{\mathrm{T}}$
- Synthesis transform matrix

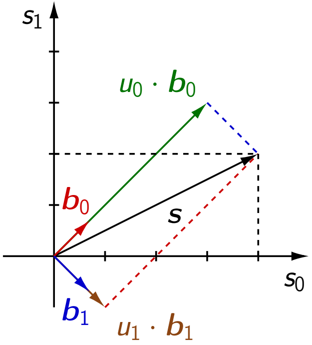
$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_0 & \boldsymbol{b}_1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- Representation of signal vector

$$\boldsymbol{s} = u_0 \cdot \boldsymbol{b}_0 + u_1 \cdot \boldsymbol{b}_1$$

$$\begin{bmatrix} 4 \\ 2 \end{bmatrix} = u_0 \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + u_1 \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} 4 \\ 2 \end{bmatrix} = 3 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 1 \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



Example of Orthogonal Transform for $N = 2$

- Vector of two samples $\boldsymbol{s} = (s_0, s_1)^{\mathrm{T}}$
- Synthesis transform matrix

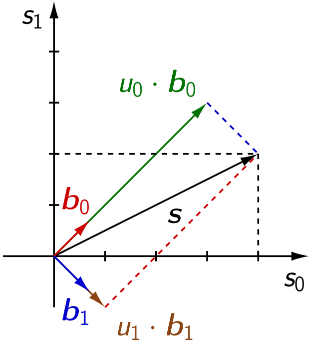
$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_0 & \boldsymbol{b}_1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- Representation of signal vector

$$\boldsymbol{s} = u_0 \cdot \boldsymbol{b}_0 + u_1 \cdot \boldsymbol{b}_1$$

$$\begin{bmatrix} 4 \\ 2 \end{bmatrix} = u_0 \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + u_1 \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} 4 \\ 2 \end{bmatrix} = 3 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 1 \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



➡ Forward transform: Project signal vector onto basis vectors

$$u_0 = \boldsymbol{b}_0^{\mathrm{T}} \cdot \boldsymbol{s} = 3\sqrt{2} \qquad \text{and} \qquad u_1 = \boldsymbol{b}_1^{\mathrm{T}} \cdot \boldsymbol{s} = \sqrt{2}$$

Unitary Transforms: MSE Distortion

Conservation of MSE distortion

- Remember: Conservation of signal energy / vector length

$$\|\boldsymbol{u}\|_2^2 = \|\boldsymbol{A} \cdot \boldsymbol{s}\|_2^2 = \|\boldsymbol{s}\|_2^2$$

Unitary Transforms: MSE Distortion

Conservation of MSE distortion

- Remember: Conservation of signal energy / vector length

$$\left\|\boldsymbol{u}\right\|_2^2 = \left\|\boldsymbol{A}\cdot\boldsymbol{s}\right\|_2^2 = \left\|\boldsymbol{s}\right\|_2^2$$

➡ Consequence for MSE distortion

$$\begin{aligned} d_N(\boldsymbol{u},\boldsymbol{u}') &= \frac{1}{N}\left\|\boldsymbol{u}-\boldsymbol{u}'\right\|_2^2 \\ &= \frac{1}{N}\left\|\boldsymbol{A}\boldsymbol{s}-\boldsymbol{B}^{-1}\boldsymbol{s}'\right\|_2^2 = \frac{1}{N}\left\|\boldsymbol{A}\left(\boldsymbol{s}-\boldsymbol{s}'\right)\right\|_2^2 \\ &= \frac{1}{N}\left\|\boldsymbol{s}-\boldsymbol{s}'\right\|_2^2 = d_N(\boldsymbol{s},\boldsymbol{s}') \end{aligned}$$

Unitary Transforms: MSE Distortion

Conservation of MSE distortion

- Remember: Conservation of signal energy / vector length

$$\|\boldsymbol{u}\|_2^2 = \|\boldsymbol{A}\cdot\boldsymbol{s}\|_2^2 = \|\boldsymbol{s}\|_2^2$$

➡ Consequence for MSE distortion

$$\begin{aligned} d_N(\boldsymbol{u},\boldsymbol{u}') &= \frac{1}{N}\|\boldsymbol{u}-\boldsymbol{u}'\|_2^2 \\ &= \frac{1}{N}\left\|\boldsymbol{A}\boldsymbol{s}-\boldsymbol{B}^{-1}\boldsymbol{s}'\right\|_2^2 = \frac{1}{N}\left\|\boldsymbol{A}\left(\boldsymbol{s}-\boldsymbol{s}'\right)\right\|_2^2 \\ &= \frac{1}{N}\|\boldsymbol{s}-\boldsymbol{s}'\|_2^2 = d_N(\boldsymbol{s},\boldsymbol{s}') \end{aligned}$$

Main Reason for using Unitary Transforms

➡ Minimization of MSE distortion $d_N(\boldsymbol{u},\boldsymbol{u}')$ in transform domain also minimizes MSE distortion $d_N(\boldsymbol{s},\boldsymbol{s}')$ in original signal space

Unitary Transforms: MSE Distortion

Conservation of MSE distortion

- Remember: Conservation of signal energy / vector length

$$\left\|\boldsymbol{u}\right\|_2^2 = \left\|\boldsymbol{A}\cdot\boldsymbol{s}\right\|_2^2 = \left\|\boldsymbol{s}\right\|_2^2$$

➡ Consequence for MSE distortion

$$\begin{aligned} d_N(\boldsymbol{u},\boldsymbol{u}') &= \frac{1}{N}\left\|\boldsymbol{u}-\boldsymbol{u}'\right\|_2^2 \\ &= \frac{1}{N}\left\|\boldsymbol{A}\boldsymbol{s}-\boldsymbol{B}^{-1}\boldsymbol{s}'\right\|_2^2 = \frac{1}{N}\left\|\boldsymbol{A}\left(\boldsymbol{s}-\boldsymbol{s}'\right)\right\|_2^2 \\ &= \frac{1}{N}\left\|\boldsymbol{s}-\boldsymbol{s}'\right\|_2^2 = d_N(\boldsymbol{s},\boldsymbol{s}') \end{aligned}$$

Main Reason for using Unitary Transforms

➡ Minimization of MSE distortion $d_N(\boldsymbol{u},\boldsymbol{u}')$ in transform domain also minimizes MSE distortion $d_N(\boldsymbol{s},\boldsymbol{s}')$ in original signal space

➡ **Enables independent scalar quantization of transform coefficients**

Unitary Transforms: Covariance Matrix

Covariance of Transform Coefficients

- Covariance matrix of transform coefficients (general case: complex values)

$$
\begin{aligned}
\boldsymbol{C}_{UU} &= \mathrm{E}\Big\{ (\boldsymbol{U} - \mathrm{E}\{\boldsymbol{U}\})\,(\boldsymbol{U} - \mathrm{E}\{\boldsymbol{U}\})^{\dagger} \Big\} \\
&= \mathrm{E}\Big\{ \boldsymbol{A}\,(\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})\,(\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})^{\dagger}\,\boldsymbol{A}^{\dagger} \Big\} \\
&= \boldsymbol{A} \cdot \mathrm{E}\Big\{ (\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})\,(\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})^{\dagger} \Big\} \cdot \boldsymbol{A}^{\dagger} \\
&= \boldsymbol{A} \cdot \boldsymbol{C}_{SS} \cdot \boldsymbol{A}^{\dagger} \\
&= \boldsymbol{A} \cdot \boldsymbol{C}_{SS} \cdot \boldsymbol{A}^{-1}
\end{aligned}
$$

Unitary Transforms: Covariance Matrix

Covariance of Transform Coefficients

- Covariance matrix of transform coefficients (general case: complex values)

$$
\begin{aligned}
\boldsymbol{C}_{UU} &= \mathrm{E}\Big\{ (\boldsymbol{U} - \mathrm{E}\{\boldsymbol{U}\})\,(\boldsymbol{U} - \mathrm{E}\{\boldsymbol{U}\})^{\dagger} \Big\} \\
&= \mathrm{E}\Big\{ \boldsymbol{A}\,(\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})\,(\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})^{\dagger}\,\boldsymbol{A}^{\dagger} \Big\} \\
&= \boldsymbol{A} \cdot \mathrm{E}\Big\{ (\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})\,(\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})^{\dagger} \Big\} \cdot \boldsymbol{A}^{\dagger} \\
&= \boldsymbol{A} \cdot \boldsymbol{C}_{SS} \cdot \boldsymbol{A}^{\dagger} \\
&= \boldsymbol{A} \cdot \boldsymbol{C}_{SS} \cdot \boldsymbol{A}^{-1}
\end{aligned}
$$

➡ Transform matrix $\boldsymbol{A}$ can be chosen in a way that (linear) statistical dependencies are reduced

Unitary Transforms: Covariance Matrix

Covariance of Transform Coefficients

- Covariance matrix of transform coefficients (general case: complex values)

$$\begin{aligned}
\boldsymbol{C}_{UU} &= \mathrm{E}\Big\{ (\boldsymbol{U} - \mathrm{E}\{\boldsymbol{U}\})\,(\boldsymbol{U} - \mathrm{E}\{\boldsymbol{U}\})^{\dagger} \Big\} \\
&= \mathrm{E}\Big\{ \boldsymbol{A}\,(\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})\,(\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})^{\dagger}\,\boldsymbol{A}^{\dagger} \Big\} \\
&= \boldsymbol{A} \cdot \mathrm{E}\Big\{ (\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})\,(\boldsymbol{S} - \mathrm{E}\{\boldsymbol{S}\})^{\dagger} \Big\} \cdot \boldsymbol{A}^{\dagger} \\
&= \boldsymbol{A} \cdot \boldsymbol{C}_{SS} \cdot \boldsymbol{A}^{\dagger} \\
&= \boldsymbol{A} \cdot \boldsymbol{C}_{SS} \cdot \boldsymbol{A}^{-1}
\end{aligned}$$

➡ Transform matrix $\boldsymbol{A}$ can be chosen in a way that (linear) statistical dependencies are reduced

➡ **Possible to increase efficiency of scalar quantization**
(if source contains linear statistical dependencies)

Unitary Transforms: Variances

Variances of Transform Coefficients

- Sum of variances: Trace of autocovariance matrix

$$\boldsymbol{C}_{UU} = \begin{bmatrix} \sigma_0^2 & x & x & \cdots & x \\ x & \sigma_1^2 & x & \cdots & x \\ x & x & \sigma_2^2 & \cdots & x \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x & x & x & \cdots & \sigma_{N-1}^2 \end{bmatrix} = \boldsymbol{A} \cdot \boldsymbol{C}_{SS} \cdot \boldsymbol{A}^{-1}$$

Unitary Transforms: Variances

Variances of Transform Coefficients

- Sum of variances: Trace of autocovariance matrix

$$\boldsymbol{C}_{UU} = \begin{bmatrix} \sigma_0^2 & x & x & \cdots & x \\ x & \sigma_1^2 & x & \cdots & x \\ x & x & \sigma_2^2 & \cdots & x \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x & x & x & \cdots & \sigma_{N-1}^2 \end{bmatrix} = \boldsymbol{A} \cdot \boldsymbol{C}_{SS} \cdot \boldsymbol{A}^{-1}$$

- Trace of a matrix is similarity-invariant

$$\mathrm{tr}(\boldsymbol{X}) = \mathrm{tr}(\boldsymbol{Q}\,\boldsymbol{X}\,\boldsymbol{Q}^{-1})$$

Unitary Transforms: Variances

Variances of Transform Coefficients

- Sum of variances: Trace of autocovariance matrix

$$\boldsymbol{C}_{UU} = \begin{bmatrix} \sigma_0^2 & x & x & \cdots & x \\ x & \sigma_1^2 & x & \cdots & x \\ x & x & \sigma_2^2 & \cdots & x \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x & x & x & \cdots & \sigma_{N-1}^2 \end{bmatrix} = \boldsymbol{A} \cdot \boldsymbol{C}_{SS} \cdot \boldsymbol{A}^{-1}$$

- Trace of a matrix is similarity-invariant

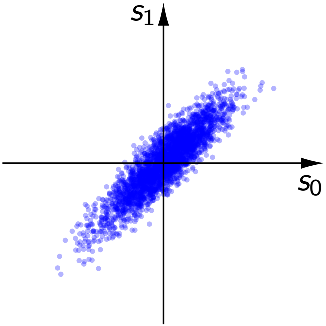
$$\mathrm{tr}(\boldsymbol{X}) = \mathrm{tr}(\boldsymbol{Q}\,\boldsymbol{X}\,\boldsymbol{Q}^{-1})$$

➡ **The arithmetic mean of the transform coefficient variances is equal to source variance**

$$\frac{1}{N} \sum_{k=0}^{N-1} \sigma_k^2 = \sigma_S^2$$

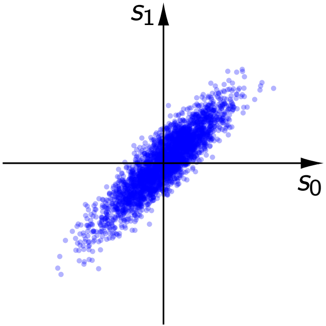
Effect of Orthogonal Transform for Correlated Sources

- 2d signal vectors of Gauss-Markov source with $\varrho = 0.9$



Effect of Orthogonal Transform for Correlated Sources

- 2d signal vectors of Gauss-Markov source with $\varrho = 0.9$

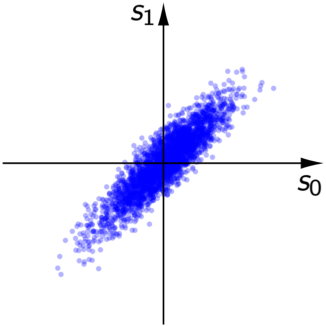


$$\boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

rotation by
$\phi = -45^\circ$

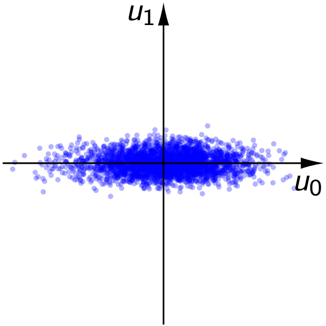
Effect of Orthogonal Transform for Correlated Sources

- 2d signal vectors of Gauss-Markov source with $\varrho = 0.9$



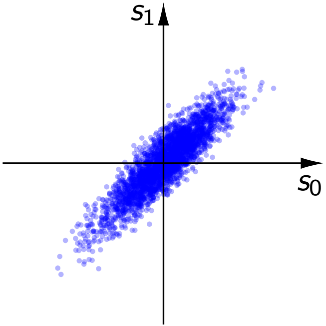
$$\boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

rotation by
$\phi = -45^\circ$



Effect of Orthogonal Transform for Correlated Sources

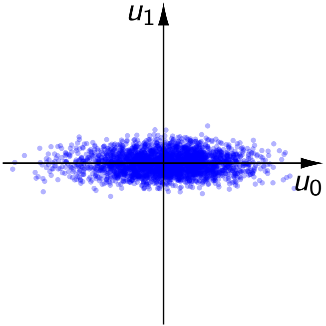
- 2d signal vectors of Gauss-Markov source with $\varrho = 0.9$



$$\boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

rotation by

$\phi = -45°$

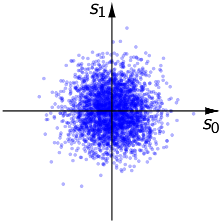


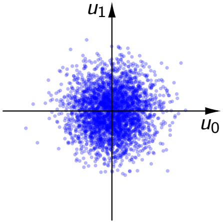
➡ Uneven distribution of transform coefficient variances: $\sigma_0^2 > \sigma_1^2$

➡ Most signal energy is concentrated in first transform coefficient

Gauss-Markov Examples for $N = 2$

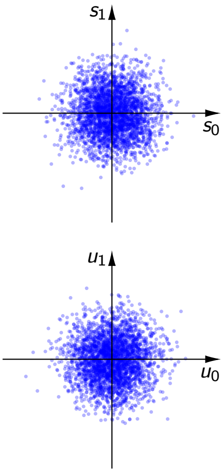
$\varrho = 0.00$



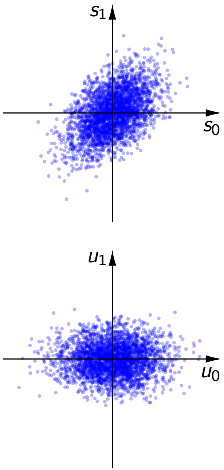


Gauss-Markov Examples for $N = 2$

$\varrho = 0.00$

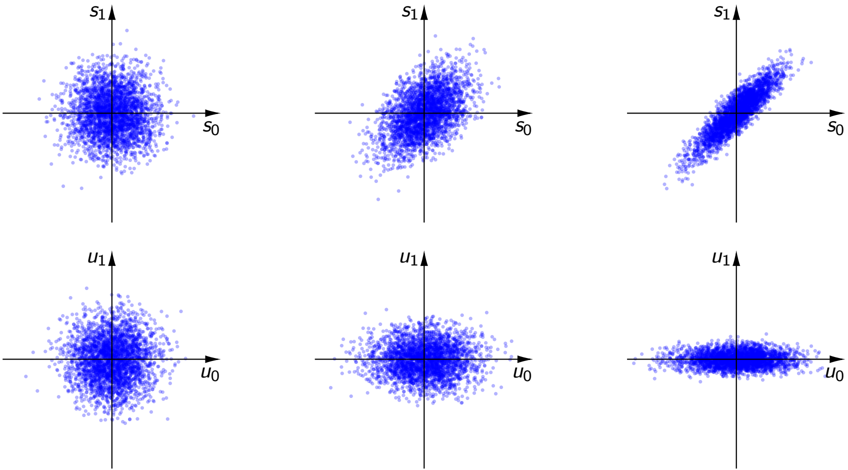


$\varrho = 0.50$

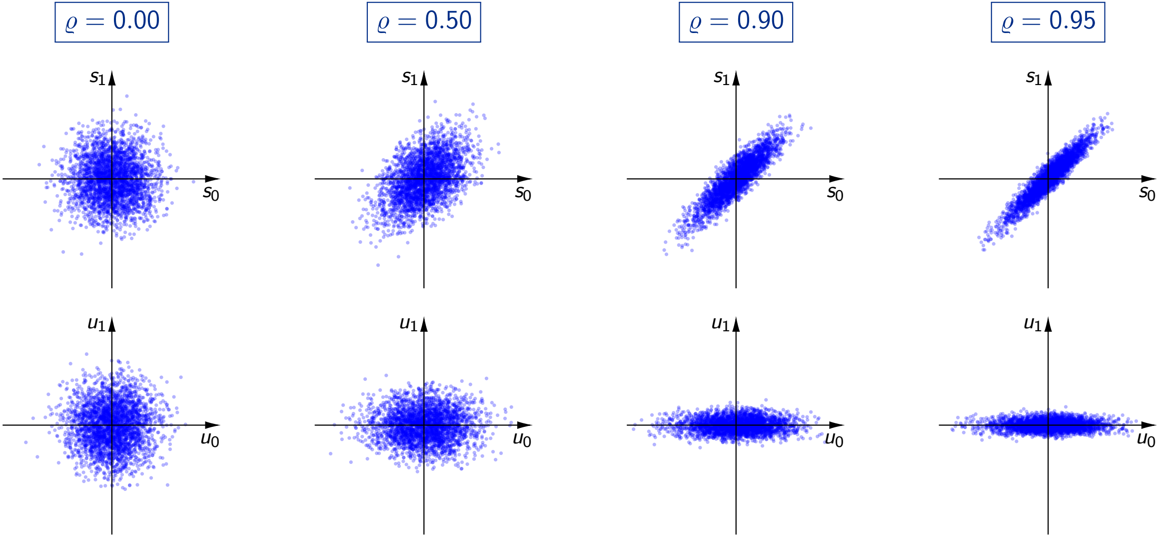


Gauss-Markov Examples for $N = 2$

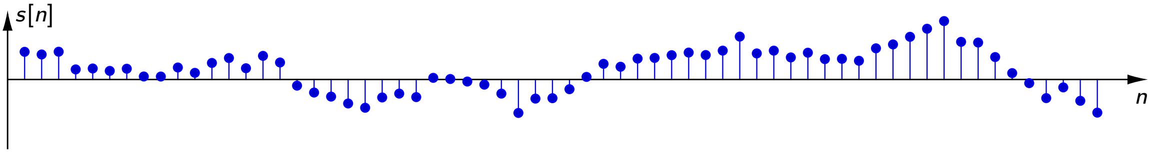
$\varrho = 0.00$ | $\varrho = 0.50$ | $\varrho = 0.90$



Gauss-Markov Examples for $N = 2$

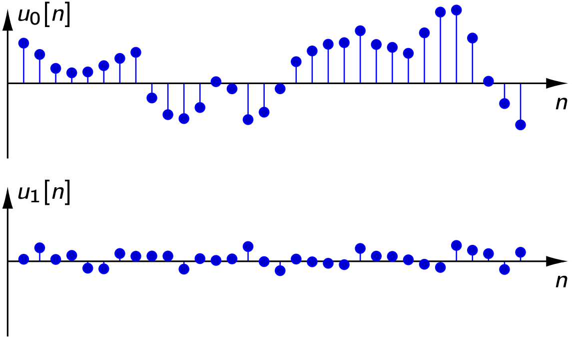


Example for Waveforms: Gauss-Markov with $\varrho = 0.95$



$$\begin{pmatrix} u_0[n] \\ u_1[n] \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} s[2n] \\ s[2n+1] \end{pmatrix}$$

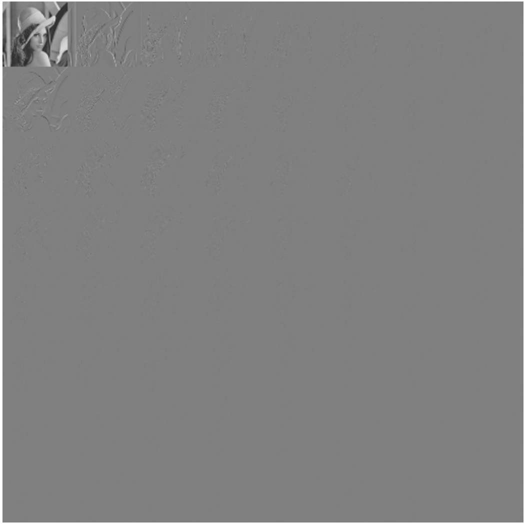
most signal energy is concentrated in u_0



Example for Images: 2×2 Block Transform (sorted Coefficients)

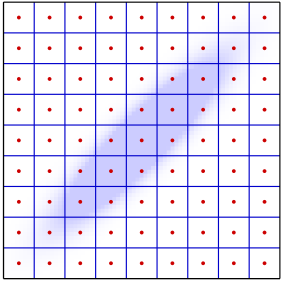
Example for Images: 4×4 Block Transform (sorted Coefficients)

Example for Images: 8×8 Block Transform (sorted Coefficients)



Transform Coding as Constrained Vector Quantizer

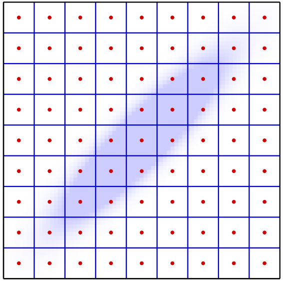
scalar quantization



quantization cells

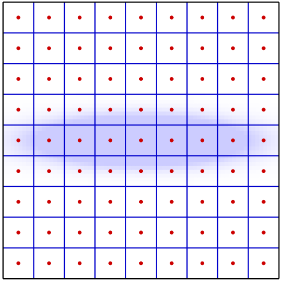
Transform Coding as Constrained Vector Quantizer

scalar quantization



quantization cells

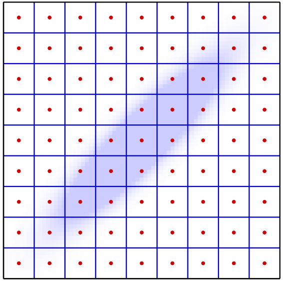
transform coding



quantization cells
in transform domain

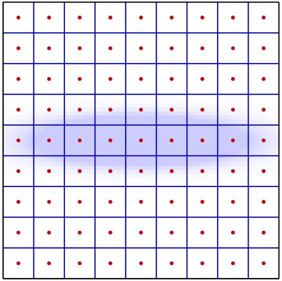
Transform Coding as Constrained Vector Quantizer

scalar quantization



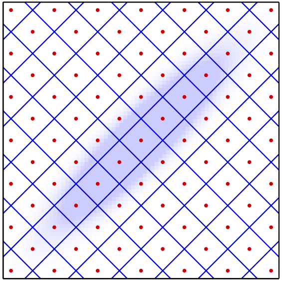
quantization cells

transform coding



quantization cells
in transform domain

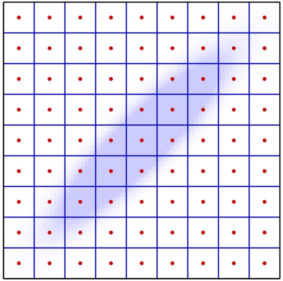
transform coding



quantization cells
in signal space

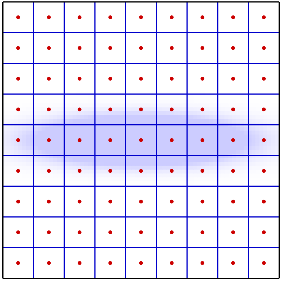
Transform Coding as Constrained Vector Quantizer

scalar quantization



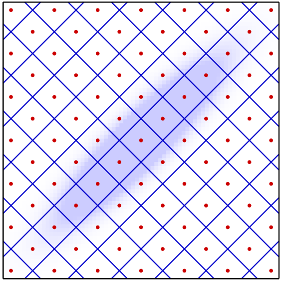
quantization cells

transform coding



quantization cells
in transform domain

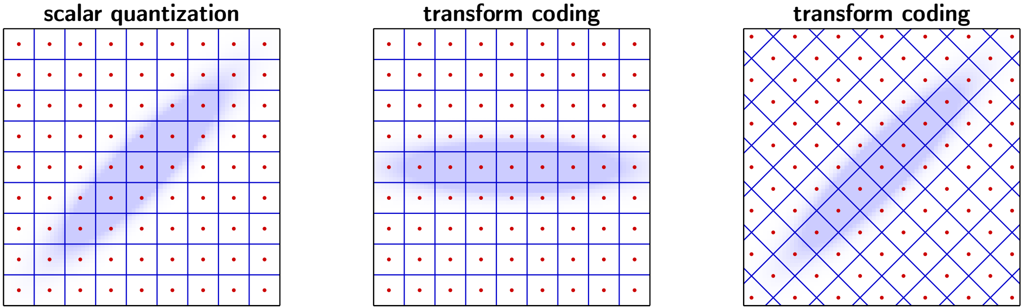
transform coding



quantization cells
in signal space

- Quantization cells are:
  - hyper-rectangles as in conventional scalar quantization
  - but rotated and aligned with the transform basis vectors

Transform Coding as Constrained Vector Quantizer



quantization cells | quantization cells in transform domain | quantization cells in signal space

- Quantization cells are:
  - hyper-rectangles as in conventional scalar quantization
  - but rotated and aligned with the transform basis vectors

➡ On average: Value of second quantization index is reduced (for correlated sources)

➡ Indicates improved coding efficiency for correlated sources (exploits memory advantage)

Bit Allocation for Transform Coefficients

- Given: Orthogonal transform with $\boldsymbol{A}$ and $\boldsymbol{B} = \boldsymbol{A}^{\mathrm{T}}$

Bit Allocation for Transform Coefficients

- Given: Orthogonal transform with $\boldsymbol{A}$ and $\boldsymbol{B} = \boldsymbol{A}^{\mathrm{T}}$
- Operational distortion-rate function of scalar quantizers (general form)

$$D_k(R_k) = \sigma_k^2 \cdot g_k(R_k)$$

Bit Allocation for Transform Coefficients

- Given: Orthogonal transform with $\boldsymbol{A}$ and $\boldsymbol{B} = \boldsymbol{A}^{\mathrm{T}}$
- Operational distortion-rate function of scalar quantizers (general form)

$$D_k(R_k) = \sigma_k^2 \cdot g_k(R_k)$$

- Overall MSE distortion D and bit rate R (transform size N)

$$D = \frac{1}{N} \sum_{k=0}^{N-1} D_k(R_k) \qquad \text{and} \qquad R = \frac{1}{N} \sum_{k=0}^{N-1} R_k$$

Bit Allocation for Transform Coefficients

- Given: Orthogonal transform with $\boldsymbol{A}$ and $\boldsymbol{B} = \boldsymbol{A}^{\mathrm{T}}$
- Operational distortion-rate function of scalar quantizers (general form)

$$D_k(R_k) = \sigma_k^2 \cdot g_k(R_k)$$

- Overall MSE distortion D and bit rate R (transform size N)

$$D = \frac{1}{N} \sum_{k=0}^{N-1} D_k(R_k) \qquad \text{and} \qquad R = \frac{1}{N} \sum_{k=0}^{N-1} R_k$$

Bit allocation

- Overall rate-distortion performance $D(R)$ depends on bit distribution among transform coefficents $R \mapsto \{R_0, R_1, \cdots\}$

Bit Allocation for Transform Coefficients

- Given: Orthogonal transform with $\boldsymbol{A}$ and $\boldsymbol{B} = \boldsymbol{A}^{\mathrm{T}}$
- Operational distortion-rate function of scalar quantizers (general form)

$$D_k(R_k) = \sigma_k^2 \cdot g_k(R_k)$$

- Overall MSE distortion D and bit rate R (transform size N)

$$D = \frac{1}{N} \sum_{k=0}^{N-1} D_k(R_k) \qquad \text{and} \qquad R = \frac{1}{N} \sum_{k=0}^{N-1} R_k$$

Bit allocation

- Overall rate-distortion performance $D(R)$ depends on bit distribution among transform coefficents $R \mapsto \{R_0, R_1, \cdots\}$

➡ Optimal bit allocation: Solution of optimization problem

$$\min \; D(R_0, R_1, \cdots) \qquad \text{subject to} \qquad \frac{1}{N} \sum_k R_k = R$$

Bit Allocation for Transform Coefficients

- Constrained optimization problem

$$\min_{R_0, R_1, \cdots} \; D(R) = \frac{1}{N} \sum_{k=0}^{N-1} D_k(R_k) \qquad \text{subject to} \qquad \frac{1}{N} \sum_{k=0}^{N-1} R_k = R$$

with $D_k(R_k)$ being the operational distortion-rate functions the scalar component quantizers

Bit Allocation for Transform Coefficients

- Constrained optimization problem

$$\min_{R_0, R_1, \cdots} \; D(R) = \frac{1}{N} \sum_{k=0}^{N-1} D_k(R_k) \qquad \text{subject to} \qquad \frac{1}{N} \sum_{k=0}^{N-1} R_k = R$$

with $D_k(R_k)$ being the operational distortion-rate functions the scalar component quantizers

➡ Reformulate as unconstrained minimization problem using the technique of Lagrange multipliers (minimize $D + \lambda R$)

$$\min_{R_0, R_1, \cdots} \; \left(\frac{1}{N} \sum_{k=0}^{N-1} D_k(R_k) \right) + \lambda \cdot \left(\frac{1}{N} \sum_{k=0}^{N-1} R_k \right)$$

Bit Allocation for Transform Coefficients

- Constrained optimization problem

$$\min_{R_0, R_1, \cdots} \; D(R) = \frac{1}{N} \sum_{k=0}^{N-1} D_k(R_k) \qquad \text{subject to} \qquad \frac{1}{N} \sum_{k=0}^{N-1} R_k = R$$

with $D_k(R_k)$ being the operational distortion-rate functions the scalar component quantizers

➡ Reformulate as unconstrained minimization problem using the technique of Lagrange multipliers (minimize $D + \lambda R$)

$$\min_{R_0, R_1, \cdots} \; \left(\frac{1}{N} \sum_{k=0}^{N-1} D_k(R_k) \right) + \lambda \cdot \left(\frac{1}{N} \sum_{k=0}^{N-1} R_k \right)$$

➡ Set derivatives with respect to R_k equal to 0

$$\frac{\partial}{\partial R_k} \left(D + \lambda R \right) \overset{!}{=} 0$$

Optimal Bit Allocation: Pareto Condition

- Minimize Lagrangian cost function $D + \lambda R$

$$\frac{\partial}{\partial R_k}\left(\frac{1}{N}\sum_{i=0}^{N-1} D_i(R_i) + \frac{\lambda}{N}\sum_{i=0}^{N-1} R_i \right) \stackrel{!}{=} 0$$

$$\frac{1}{N} \cdot \frac{\partial}{\partial R_k} D_k(R_k) + \frac{\lambda}{N} \stackrel{!}{=} 0$$

Optimal Bit Allocation: Pareto Condition

- Minimize Lagrangian cost function $D + \lambda R$

$$\frac{\partial}{\partial R_k} \left(\frac{1}{N} \sum_{i=0}^{N-1} D_i(R_i) + \frac{\lambda}{N} \sum_{i=0}^{N-1} R_i \right) \stackrel{!}{=} 0$$

$$\frac{1}{N} \cdot \frac{\partial}{\partial R_k} D_k(R_k) + \frac{\lambda}{N} \stackrel{!}{=} 0$$

➡ Solution: **Pareto condition**

$$\boxed{\frac{\partial D_k(R_k)}{\partial R_k} = -\lambda = \text{const}}$$

Optimal Bit Allocation: Pareto Condition

- Minimize Lagrangian cost function $D + \lambda R$

$$\frac{\partial}{\partial R_k}\left(\frac{1}{N}\sum_{i=0}^{N-1} D_i(R_i) + \frac{\lambda}{N}\sum_{i=0}^{N-1} R_i \right) \overset{!}{=} 0$$

$$\frac{1}{N} \cdot \frac{\partial}{\partial R_k} D_k(R_k) + \frac{\lambda}{N} \overset{!}{=} 0$$

➡ Solution: **Pareto condition**

$$\boxed{\frac{\partial D_k(R_k)}{\partial R_k} = -\lambda = \text{const}}$$

➡ All component quantizers have to be operated at the same slope of their operational distortion-rate function

Optimal Bit Allocation: Pareto Condition

- Minimize Lagrangian cost function $D + \lambda R$

$$\frac{\partial}{\partial R_k}\left(\frac{1}{N} \sum_{i=0}^{N-1} D_i(R_i) + \frac{\lambda}{N} \sum_{i=0}^{N-1} R_i \right) \stackrel{!}{=} 0$$

$$\frac{1}{N} \cdot \frac{\partial}{\partial R_k} D_k(R_k) + \frac{\lambda}{N} \stackrel{!}{=} 0$$

➡ Solution: **Pareto condition**

$$\boxed{\frac{\partial D_k(R_k)}{\partial R_k} = -\lambda = \text{const}}$$

➡ All component quantizers have to be operated at the same slope of their operational distortion-rate function

➡ Interpretation: Move bits from coefficients with small distortion reduction per bit to coefficients with larger distortion reduction per bit

High-Rate Approximation: Bit Allocation

High Rates

- All component quantizers are operated at high component rates R_k

High-Rate Approximation: Bit Allocation

High Rates

- All component quantizers are operated at high component rates R_k
- High-rate approximation of distortion-rate function for component quantizers

$$D_k(R_k) = \varepsilon_k^2 \cdot \sigma_k^2 \cdot 2^{-2R_k}$$

where ε_k^2 depends on transform coefficient distribution and quantizer

High-Rate Approximation: Bit Allocation

High Rates

- All component quantizers are operated at high component rates R_k
- High-rate approximation of distortion-rate function for component quantizers

$$D_k(R_k) = \varepsilon_k^2 \cdot \sigma_k^2 \cdot 2^{-2R_k}$$

where ε_k^2 depends on transform coefficient distribution and quantizer

Optimal Bit Allocation at High Rates

- Pareto condition

$$\frac{\partial}{\partial R_k} D_k(R_k) = -2\,\ln 2\,\varepsilon_k^2\,\sigma_k^2\,2^{-2R_k} = -2\,\ln 2\,D_k(R_k) = -\lambda = \text{const}$$

High-Rate Approximation: Bit Allocation

High Rates

- All component quantizers are operated at high component rates R_k
- High-rate approximation of distortion-rate function for component quantizers

$$D_k(R_k) = \varepsilon_k^2 \cdot \sigma_k^2 \cdot 2^{-2R_k}$$

where ε_k^2 depends on transform coefficient distribution and quantizer

Optimal Bit Allocation at High Rates

- Pareto condition

$$\frac{\partial}{\partial R_k} D_k(R_k) = -2\,\ln 2\,\varepsilon_k^2\,\sigma_k^2\,2^{-2R_k} = -2\,\ln 2\,D_k(R_k) = -\lambda = \text{const}$$

➡ **All component quantizers are operated at the same distortion**

$$D_k(R_k) = D$$

High-Rate Approximation: Bit Allocation

Optimal Bit Allocation

- All component quantizers are operated at the same distortion

$$D_k(R_k) = \varepsilon_k^2 \cdot \sigma_k^2 \cdot 2^{-2R_k} = D$$

High-Rate Approximation: Bit Allocation

Optimal Bit Allocation

- All component quantizers are operated at the same distortion

$$D_k(R_k) = \varepsilon_k^2 \cdot \sigma_k^2 \cdot 2^{-2R_k} = D$$

➡ Bit allocation rule

$$R_k(D) = \frac{1}{2} \log_2\left(\frac{\varepsilon_k^2 \, \sigma_k^2}{D} \right)$$

High-Rate Approximation: Bit Allocation

Optimal Bit Allocation

- All component quantizers are operated at the same distortion

$$D_k(R_k) = \varepsilon_k^2 \cdot \sigma_k^2 \cdot 2^{-2R_k} = D$$

➞ Bit allocation rule

$$R_k(D) = \frac{1}{2} \log_2\left(\frac{\varepsilon_k^2 \, \sigma_k^2}{D}\right)$$

Overall Operational Rate-Distortion Function

- Use result of optimal bit allocation

$$R(D) = \frac{1}{N} \sum_{k=0}^{N-1} R_k(D) = \frac{1}{2N} \sum_{k=0}^{N-1} \log_2\left(\frac{\varepsilon_k^2 \, \sigma_k^2}{D}\right)$$

High-Rate Approximation: Distortion-Rate Function

- Operational Rate-Distortion Function

$$R(D) = \frac{1}{2N} \sum_{k=0}^{N-1} \log_2\left(\frac{\varepsilon_k^2 \, \sigma_k^2}{D} \right) = \frac{1}{2} \log_2\left(\frac{1}{D} \left(\prod_{k=0}^{N-1} \varepsilon_k^2 \right)^{\frac{1}{N}} \left(\prod_{k=0}^{N-1} \sigma_k^2 \right)^{\frac{1}{N}} \right)$$

High-Rate Approximation: Distortion-Rate Function

- Operational Rate-Distortion Function

$$R(D) = \frac{1}{2N} \sum_{k=0}^{N-1} \log_2\left(\frac{\varepsilon_k^2\, \sigma_k^2}{D} \right) = \frac{1}{2} \log_2\left(\frac{1}{D} \left(\prod_{k=0}^{N-1} \varepsilon_k^2 \right)^{\frac{1}{N}} \left(\prod_{k=0}^{N-1} \sigma_k^2 \right)^{\frac{1}{N}} \right)$$

- Define geometric means

$$\tilde{\sigma}^2 = \left(\prod_{k=0}^{N-1} \sigma_k^2 \right)^{\frac{1}{N}} \qquad \text{and} \qquad \tilde{\varepsilon}^2 = \left(\prod_{k=0}^{N-1} \varepsilon_k^2 \right)^{\frac{1}{N}}$$

High-Rate Approximation: Distortion-Rate Function

- Operational Rate-Distortion Function

$$R(D) = \frac{1}{2N} \sum_{k=0}^{N-1} \log_2\left(\frac{\varepsilon_k^2 \, \sigma_k^2}{D} \right) = \frac{1}{2} \log_2\left(\frac{1}{D} \left(\prod_{k=0}^{N-1} \varepsilon_k^2 \right)^{\frac{1}{N}} \left(\prod_{k=0}^{N-1} \sigma_k^2 \right)^{\frac{1}{N}} \right)$$

- Define geometric means

$$\tilde{\sigma}^2 = \left(\prod_{k=0}^{N-1} \sigma_k^2 \right)^{\frac{1}{N}} \qquad \text{and} \qquad \tilde{\varepsilon}^2 = \left(\prod_{k=0}^{N-1} \varepsilon_k^2 \right)^{\frac{1}{N}}$$

➡ **High-rate rate-distortion / distortion-rate function** (for optimal bit allocation)

$$\boxed{R(D) = \frac{1}{2} \log_2\left(\frac{\tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2}{D} \right)} \qquad \text{and} \qquad \boxed{D(R) = \tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2 \cdot 2^{-2R}}$$

High-Rate Approximation for Gaussian Sources

Transform Coding for Gaussian Sources

- Any linear combination of Gaussian random variables is also a Gaussian random variable

➡ All transform coefficients represent Gaussian random variables

High-Rate Approximation for Gaussian Sources

Transform Coding for Gaussian Sources

- Any linear combination of Gaussian random variables is also a Gaussian random variable

➡ All transform coefficients represent Gaussian random variables

Transform Coding for Gaussian Sources using Optimal Scalar Quantizers

- High-rate distortion-rate function of entropy-constrained scalar quantizers

$$D_k(R_k) = \frac{\pi e}{6} \cdot \sigma_k^2 \cdot 2^{-2R_k}$$

High-Rate Approximation for Gaussian Sources

Transform Coding for Gaussian Sources

- Any linear combination of Gaussian random variables is also a Gaussian random variable

➡ All transform coefficients represent Gaussian random variables

Transform Coding for Gaussian Sources using Optimal Scalar Quantizers

- High-rate distortion-rate function of entropy-constrained scalar quantizers

$$D_k(R_k) = \frac{\pi e}{6} \cdot \sigma_k^2 \cdot 2^{-2R_k}$$

➡ Overall high-rate distortion-rate function for Gaussian sources

$$D_G(R) = \frac{\pi e}{6} \cdot \tilde{\sigma}^2 \cdot 2^{-2R}$$

High-Rate Approximation for Gaussian Sources

Transform Coding for Gaussian Sources

- Any linear combination of Gaussian random variables is also a Gaussian random variable

➡ All transform coefficients represent Gaussian random variables

Transform Coding for Gaussian Sources using Optimal Scalar Quantizers

- High-rate distortion-rate function of entropy-constrained scalar quantizers

$$D_k(R_k) = \frac{\pi e}{6} \cdot \sigma_k^2 \cdot 2^{-2R_k}$$

➡ Overall high-rate distortion-rate function for Gaussian sources

$$D_G(R) = \frac{\pi e}{6} \cdot \tilde{\sigma}^2 \cdot 2^{-2R}$$

➡ Improvement relative to scalar quantization for uneven distribution of transform coefficient variances

Transform Coding Gain at High Rates

Transform Coding Gain

- Ratio of distortion for scalar quantization and transform coding

Transform Coding Gain at High Rates

Transform Coding Gain

- Ratio of distortion for scalar quantization and transform coding
- Transform coding gain at high rates

$$G_T = \frac{D_{SQ}(R)}{D_{TQ}(R)} = \frac{\varepsilon_S^2 \cdot \sigma_S^2 \cdot 2^{-2R}}{\tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2 \cdot 2^{-2R}} = \frac{\varepsilon_S^2 \cdot \sigma_S^2}{\tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2}$$

Transform Coding Gain at High Rates

Transform Coding Gain

- Ratio of distortion for scalar quantization and transform coding
- Transform coding gain at high rates

$$G_T = \frac{D_{SQ}(R)}{D_{TQ}(R)} = \frac{\varepsilon_S^2 \cdot \sigma_S^2 \cdot 2^{-2R}}{\tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2 \cdot 2^{-2R}} = \frac{\varepsilon_S^2 \cdot \sigma_S^2}{\tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2}$$

Transform Coding Gain for Gaussian Sources

- High-rate transform coding gain for Gaussian sources

$$G_T = \frac{\sigma_S^2}{\tilde{\sigma}^2} = \frac{\frac{1}{N} \sum_{k=0}^{N-1} \sigma_k^2}{\sqrt[N]{\prod_{k=0}^{N-1} \sigma_k^2}}$$

Transform Coding Gain at High Rates

Transform Coding Gain

- Ratio of distortion for scalar quantization and transform coding
- Transform coding gain at high rates

$$G_T = \frac{D_{SQ}(R)}{D_{TQ}(R)} = \frac{\varepsilon_S^2 \cdot \sigma_S^2 \cdot 2^{-2R}}{\tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2 \cdot 2^{-2R}} = \frac{\varepsilon_S^2 \cdot \sigma_S^2}{\tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2}$$

Transform Coding Gain for Gaussian Sources

- High-rate transform coding gain for Gaussian sources

$$G_T = \frac{\sigma_S^2}{\tilde{\sigma}^2} = \frac{\frac{1}{N}\sum_{k=0}^{N-1}\sigma_k^2}{\sqrt[N]{\prod_{k=0}^{N-1}\sigma_k^2}}$$

➡ Ratio of arithmetic and geometric mean of the transform coefficient variances

Transform Coding Gain at High Rates

Transform Coding Gain

- Ratio of distortion for scalar quantization and transform coding
- Transform coding gain at high rates

$$G_T = \frac{D_{SQ}(R)}{D_{TQ}(R)} = \frac{\varepsilon_S^2 \cdot \sigma_S^2 \cdot 2^{-2R}}{\tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2 \cdot 2^{-2R}} = \frac{\varepsilon_S^2 \cdot \sigma_S^2}{\tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2}$$

Transform Coding Gain for Gaussian Sources

- High-rate transform coding gain for Gaussian sources

$$G_T = \frac{\sigma_S^2}{\tilde{\sigma}^2} = \frac{\frac{1}{N}\sum_{k=0}^{N-1} \sigma_k^2}{\sqrt[N]{\prod_{k=0}^{N-1} \sigma_k^2}}$$

➡ Ratio of arithmetic and geometric mean of the transform coefficient variances

➡ Transform coding gain is maximized if the geometric mean $\tilde{\sigma}^2$ of variances is minimized

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Input vector and transform matrix

$$\boldsymbol{s} = \begin{bmatrix} s_0 \\ s_1 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Input vector and transform matrix

$$\boldsymbol{s} = \begin{bmatrix} s_0 \\ s_1 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

➡ Transformation

$$\boldsymbol{u} = \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \boldsymbol{A} \cdot \boldsymbol{s} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \end{bmatrix}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Input vector and transform matrix

$$\boldsymbol{s} = \begin{bmatrix} s_0 \\ s_1 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

➡ Transformation

$$\boldsymbol{u} = \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \boldsymbol{A} \cdot \boldsymbol{s} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \end{bmatrix}$$

➡ Transform coefficients

$$u_0 = \frac{1}{\sqrt{2}}(s_0 + s_1) \qquad \text{and} \qquad u_0 = \frac{1}{\sqrt{2}}(s_0 - s_1)$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Input vector and transform matrix

$$\boldsymbol{s} = \begin{bmatrix} s_0 \\ s_1 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

➡ Transformation

$$\boldsymbol{u} = \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \boldsymbol{A} \cdot \boldsymbol{s} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \end{bmatrix}$$

➡ Transform coefficients

$$u_0 = \frac{1}{\sqrt{2}}(s_0 + s_1) \qquad \text{and} \qquad u_0 = \frac{1}{\sqrt{2}}(s_0 - s_1)$$

- Inverse transformation

$$\boldsymbol{B} = \boldsymbol{A}^{-1} = \boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Transform coefficients

$$u_0 = \frac{1}{\sqrt{2}}(s_0 + s_1) \qquad \text{and} \qquad u_0 = \frac{1}{\sqrt{2}}(s_0 - s_1)$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Transform coefficients

$$u_0 = \frac{1}{\sqrt{2}}(s_0 + s_1) \qquad \text{and} \qquad u_0 = \frac{1}{\sqrt{2}}(s_0 - s_1)$$

- Variance of transform coefficients

$$\begin{aligned} \sigma_0^2 &= \mathrm{E}\{\, U_0^2 \,\} = \frac{1}{2}\mathrm{E}\{\, (S_0 + S_1)^2 \,\} = \frac{1}{2}\Big(\mathrm{E}\{\, S_0^2 \,\} + \mathrm{E}\{\, S_1^2 \,\} + 2\mathrm{E}\{\, S_0 S_1 \,\}\Big) \\ &= \frac{1}{2}\left(\sigma_S^2 + \sigma_S^2 + 2\sigma_S^2 \varrho\right) = \sigma_S^2\,(1 + \varrho) \end{aligned}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Transform coefficients

$$u_0 = \frac{1}{\sqrt{2}}(s_0 + s_1) \qquad \text{and} \qquad u_0 = \frac{1}{\sqrt{2}}(s_0 - s_1)$$

- Variance of transform coefficients

$$\begin{aligned}\sigma_0^2 &= \mathrm{E}\{\, U_0^2 \,\} = \frac{1}{2}\mathrm{E}\{\, (S_0 + S_1)^2 \,\} = \frac{1}{2}\Big(\mathrm{E}\{\, S_0^2 \,\} + \mathrm{E}\{\, S_1^2 \,\} + 2\mathrm{E}\{\, S_0 S_1 \,\}\Big) \\ &= \frac{1}{2}\left(\sigma_S^2 + \sigma_S^2 + 2\sigma_S^2\varrho\right) = \sigma_S^2\,(1 + \varrho)\end{aligned}$$

$$\sigma_1^2 = \mathrm{E}\{\, U_1^2 \,\} = \sigma_S^2\,(1 - \varrho)$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Transform coefficients

$$u_0 = \frac{1}{\sqrt{2}}(s_0 + s_1) \qquad \text{and} \qquad u_0 = \frac{1}{\sqrt{2}}(s_0 - s_1)$$

- Variance of transform coefficients

$$\begin{aligned} \sigma_0^2 &= \mathrm{E}\{\, U_0^2 \,\} = \frac{1}{2}\mathrm{E}\{\, (S_0 + S_1)^2 \,\} = \frac{1}{2}\Big(\mathrm{E}\{\, S_0^2 \,\} + \mathrm{E}\{\, S_1^2 \,\} + 2\mathrm{E}\{\, S_0 S_1 \,\}\Big) \\ &= \frac{1}{2}\left(\sigma_S^2 + \sigma_S^2 + 2\sigma_S^2 \varrho\right) = \sigma_S^2\,(1 + \varrho) \end{aligned}$$

$$\sigma_1^2 = \mathrm{E}\{\, U_1^2 \,\} = \sigma_S^2\,(1 - \varrho)$$

- Cross-correlation of transform coefficients

$$E\{U_0 U_1\} = \frac{1}{2}\mathrm{E}\{\, (S_0 + S_1)\,(S_0 - S_1) \,\} = \frac{1}{2}\mathrm{E}\{\, S_0^2 - S_1^2 \,\} = \sigma_S^2 - \sigma_S^2 = 0$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- High rate distortion-rate functions of component quantizers

$$D_0(R_0) \;=\; \varepsilon^2\,\sigma_0^2\,2^{-2R_0} = \varepsilon^2\,\sigma_S^2\,(1+\varrho)\,2^{-2R_0}$$

$$D_1(R_1) \;=\; \varepsilon^2\,\sigma_1^2\,2^{-2R_1} = \varepsilon^2\,\sigma_S^2\,(1-\varrho)\,2^{-2R_1}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- High rate distortion-rate functions of component quantizers

$$D_0(R_0) = \varepsilon^2\,\sigma_0^2\,2^{-2R_0} = \varepsilon^2\,\sigma_S^2\,(1+\varrho)\,2^{-2R_0}$$
$$D_1(R_1) = \varepsilon^2\,\sigma_1^2\,2^{-2R_1} = \varepsilon^2\,\sigma_S^2\,(1-\varrho)\,2^{-2R_1}$$

- Optimal bit allocation: Pareto condition at high rates $D_0(R_0) = D_1(R_1)$

$$\varepsilon^2\,\sigma_S^2\,(1+\varrho)\,2^{-2R_0} = \varepsilon^2\,\sigma_S^2\,(1-\varrho)\,2^{-2R_1}$$
$$\log_2(1+\varrho) - 2R_0 = \log_2(1-\varrho) - 2R_1$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- High rate distortion-rate functions of component quantizers

$$D_0(R_0) = \varepsilon^2 \sigma_0^2 \, 2^{-2R_0} = \varepsilon^2 \sigma_S^2 (1+\varrho) \, 2^{-2R_0}$$
$$D_1(R_1) = \varepsilon^2 \sigma_1^2 \, 2^{-2R_1} = \varepsilon^2 \sigma_S^2 (1-\varrho) \, 2^{-2R_1}$$

- Optimal bit allocation: Pareto condition at high rates $D_0(R_0) = D_1(R_1)$

$$\varepsilon^2 \sigma_S^2 (1+\varrho) \, 2^{-2R_0} = \varepsilon^2 \sigma_S^2 (1-\varrho) \, 2^{-2R_1}$$
$$\log_2(1+\varrho) - 2R_0 = \log_2(1-\varrho) - 2R_1$$

➡ Using $R = \frac{1}{2}(R_0 + R_1)$ ➡ $R_1 = 2R - R_0$

$$\log_2(1+\varrho) - 2R_0 = \log_2(1-\varrho) - 4R + 2R_0$$
$$4R_0 = 4R + \log_2(1+\varrho) - \log_2(1-\varrho)$$
$$R_0 = R + \frac{1}{4} \log_2\left(\frac{1+\varrho}{1-\varrho}\right)$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Optimal bit allocation

$$R_0 = R + \frac{1}{4}\log_2\left(\frac{1+\varrho}{1-\varrho}\right) \qquad \text{and} \qquad R_1 = R - \frac{1}{4}\log_2\left(\frac{1+\varrho}{1-\varrho}\right)$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Optimal bit allocation

$$R_0 = R + \frac{1}{4}\log_2\left(\frac{1+\varrho}{1-\varrho}\right) \qquad \text{and} \qquad R_1 = R - \frac{1}{4}\log_2\left(\frac{1+\varrho}{1-\varrho}\right)$$

- Resulting component distortions

$$\begin{aligned} D_0(R) &= \varepsilon^2\,\sigma_s^2\,(1+\varrho)\,2^{-2R-\frac{1}{2}\log_2\left(\frac{1+\varrho}{1-\varrho}\right)} \\ &= \varepsilon^2\,\sigma_s^2\,(1+\varrho)\,2^{-2R}\sqrt{\frac{1-\varrho}{1+\varrho}} = \varepsilon^2\,\sigma_s^2\,\sqrt{1-\varrho^2}\,2^{-2R} \end{aligned}$$

$$\begin{aligned} D_1(R) &= \varepsilon^2\,\sigma_s^2\,(1-\varrho)\,2^{-2R+\frac{1}{2}\log_2\left(\frac{1+\varrho}{1-\varrho}\right)} \\ &= \varepsilon^2\,\sigma_s^2\,(1-\varrho)\,2^{-2R}\sqrt{\frac{1+\varrho}{1-\varrho}} = \varepsilon^2\,\sigma_s^2\,\sqrt{1-\varrho^2}\,2^{-2R} \end{aligned}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Component distortions

$$D_0(R) = D_1(R) = \varepsilon^2 \, \sigma_s^2 \, \sqrt{1 - \varrho^2} \, 2^{-2R}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Component distortions

$$D_0(R) = D_1(R) = \varepsilon^2\, \sigma_s^2\, \sqrt{1-\varrho^2}\, 2^{-2R}$$

- Distortion rate function

$$D(R) = \frac{1}{2}\big(D_0(R) + D_1(R)\big) = \varepsilon^2\, \sigma_s^2\, \sqrt{1-\varrho^2}\, 2^{-2R}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Component distortions

$$D_0(R) = D_1(R) = \varepsilon^2 \, \sigma_s^2 \, \sqrt{1 - \varrho^2} \, 2^{-2R}$$

- Distortion rate function

$$D(R) = \frac{1}{2}\big(D_0(R) + D_1(R)\big) = \varepsilon^2 \, \sigma_s^2 \, \sqrt{1 - \varrho^2} \, 2^{-2R}$$

- Geometric mean of variances

$$\tilde{\sigma}^2 \; = \; \sqrt{\sigma_0^2 \cdot \sigma_1^2} = \sigma_S^2 \cdot \sqrt{(1 + \varrho)(1 - \varrho)} = \sigma_S^2 \cdot \sqrt{1 - \varrho^2}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Component distortions

$$D_0(R) = D_1(R) = \varepsilon^2\, \sigma_s^2\, \sqrt{1-\varrho^2}\, 2^{-2R}$$

- Distortion rate function

$$D(R) = \frac{1}{2}\big(D_0(R) + D_1(R)\big) = \varepsilon^2\, \sigma_s^2\, \sqrt{1-\varrho^2}\, 2^{-2R}$$

- Geometric mean of variances

$$\tilde{\sigma}^2 \;=\; \sqrt{\sigma_0^2 \cdot \sigma_1^2} = \sigma_S^2 \cdot \sqrt{(1+\varrho)(1-\varrho)} = \sigma_S^2 \cdot \sqrt{1-\varrho^2}$$

➡ Yields same expression for distortion rate function

$$D(R) = \varepsilon^2\, \tilde{\sigma}^2\, 2^{-2R} = \varepsilon^2\, \sigma_S^2\, \sqrt{1-\varrho^2}\, 2^{-2R}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

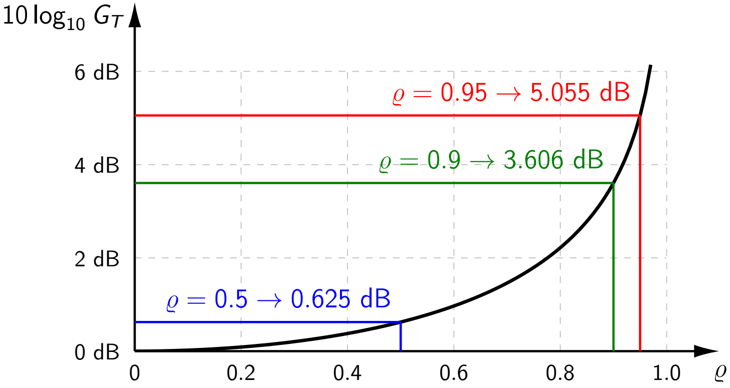
- Transform coding gain for $N = 2$

$$G_T = \frac{\varepsilon^2\,\sigma_S^2\,2^{-2R}}{\varepsilon^2\,\sigma_S^2\,\sqrt{1-\varrho^2}\,2^{-2R}} = \frac{1}{\sqrt{1-\varrho^2}}$$

Example: Transform Coding with $N = 2$ for Zero-Mean Gaussian

- Transform coding gain for $N = 2$

$$G_T = \frac{\varepsilon^2 \, \sigma_S^2 \, 2^{-2R}}{\varepsilon^2 \, \sigma_S^2 \, \sqrt{1-\varrho^2} \, 2^{-2R}} = \frac{1}{\sqrt{1-\varrho^2}}$$



Summary of Lecture

Transform Coding

- Linear unitary/orthogonal transform of block/vector of N consecutive samples
- Scalar quantization of resulting transform coefficients
- Inverse linear transform of reconstructed transform coefficients

Orthogonal Block Transforms

- Inverse transform matrix = Transpose of forward transform matrix

➡ Coordinate axes remain orthogonal to each other (independent quantization)

➡ MSE distortion: Same in transform domain and signal space

Bit Allocation

- Optimal bit allocation: Pareto condition (same slope for all $D_k(R_K)$)
- For high rates: Optimum bit allocation yields equal component distortions $D_k = D$

Summary of Lecture

High-Rate Approximations

- Distortion-rate function of transform coding

$$D(R) = \tilde{\varepsilon}^2 \cdot \tilde{\sigma}^2 \cdot 2^{-2R}$$

- Transform coding gain for Gaussian sources

$$G_T = \frac{\bar{\sigma}^2}{\tilde{\sigma}^2} = \frac{\text{arithmetic mean of variances}}{\text{geometric mean of variances}}$$

➞ Goal of transform: Compaction of signal energy in few transform coefficients

Open Questions

- What is the optimal transform for a given sources ?
- Practical aspects of transform coding

Exercise 1: Orthogonal Transforms of Size $N = 2$ (part I)

If we neglect possible reflections of coordinate axes, all orthogonal transforms for 2-d vectors can be specified by

$$\boldsymbol{A} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

where α is an arbitrary rotation angle.

Consider a zero-mean Gaussian process with variance σ_S^2 and the first-order correlation coefficient ϱ.

(a) Calculate the variances σ_0^2 and σ_1^2 of the resulting transform coefficients as function of ϱ and α.

(b) Calculate the covariance σ_{01}^2 between the resulting transform coefficients as function of ϱ and α.

(c) Consider an even rate distribution $R_0 = R_1 = R$ and determine the associated high-rate distortion-rate function. Does transform coding improve the coding efficiency relative to scalar quantization for this case?

Exercise 1: Orthogonal Transforms of Size $N = 2$ (part II)

(d) Given is the overall rate $R = (R_0 + R_1)/2$. Determine the rate distribution (R_0, R_1) for which the overall distortion $D = (D_0 + D_1)/2$ is minimized (assume that the high rate approximation for scalar quantization of the transform coefficients is valid).

(e) Determine the overall distortion-rate function for optimal rate allocation (and high rates).

(f) Determine the high-rate transform coding gain, which is given by

$$G_T = \frac{D_{\text{scalar quantization}}(R)}{D_{\text{transform coding}}(R)}$$

(g) For what rotation angles is the high-rate transform coding gain maximized (or the distortion minimized)?

Does the optimal rotation angle depend on the correlation coefficient ϱ?

Exercise 2: Implement a PSNR Tool for PPM Images

Implement a tool for measuring PSNRs between two PPM images

- Input to the tool shall be two images in PPM format (original and reconstructed)
- The tool should output the following four Peak-Signal-to-Noise Ratios (PSNR measures)
  - ➡ PSNR of red component, PSNR of green component, PSNR of blue component
  - ➡ Average of the red, green, and blue PSNR

Test the tool by

- Coding one of our test images with JPEG (e.g., using “convert test.ppm test.jpg”)
- Reconstructing the JPEG-coded image into the ppm format (e.g., using “convert test.jpg rec.ppm”)
- Measuring the PSNRs between the original and reconstructed image using the implemented tool

The PSNR for a color component $c[x, y]$ and its reconstruction $c'[x, y]$ is defined as follows

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{255^2}{\text{MSE}}\right) \qquad \text{with} \qquad \text{MSE} = \frac{1}{\text{width} \cdot \text{height}} \sum_{x,y} \big(c'[x, y] - c[x, y]\big)^2$$