Graph Transform Coding for AV2

Aleix Seguí Ugalde

Director: Joel Solé Rojals
Tutor: Ferran Marquès Acosta

telecos BCN

Overview

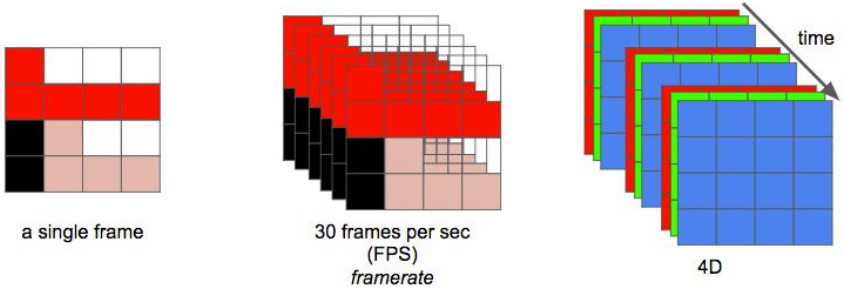
- **Goal:** explore the potential of **Graph Transforms** to improve the video compression for the developing video codec **AV2**.

- The presentation is organized in **4 sections**:
  - Video Coding Background.
  - Unitary Transforms.
  - Irregular Transforms.
  - Implementation for AV2.

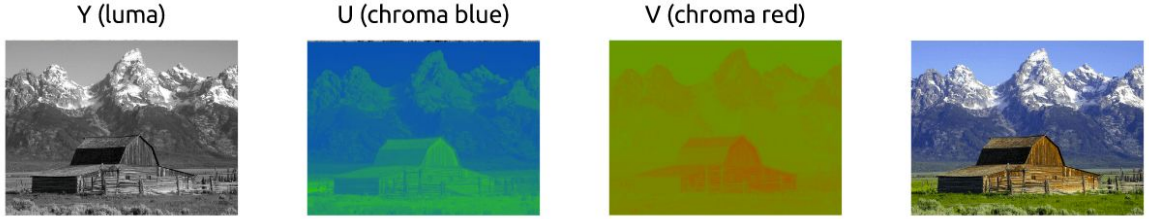
Background

Introduction to digital video compression

- A digital video sequence is a succession of frames in three channels.



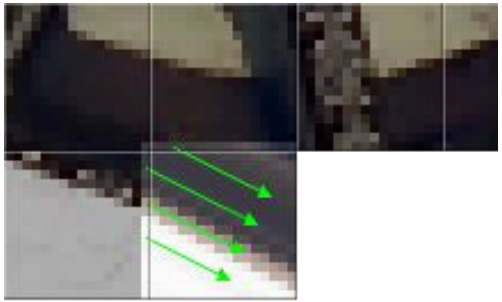
- Raw video data are very large, but very **redundant**.
- **Lossy compression** is a practical solution, which is carried out by **video codecs**.
- The typical color space is **YUV** (luma, chroma blue and chroma red).



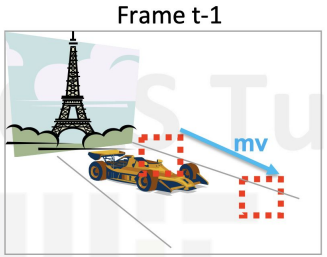
Compression Strategies

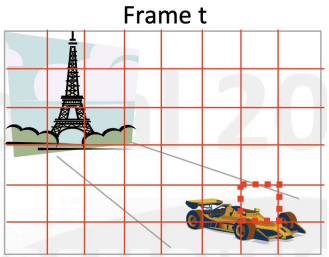
- Video codecs exploit **redundancies** within a video signal to achieve great compression.

Type of redundancy	Compression tool
Spatial	Intra prediction
Temporal	Inter prediction
Perceptual	Quantization
Statistical	Entropy coding



Intra prediction

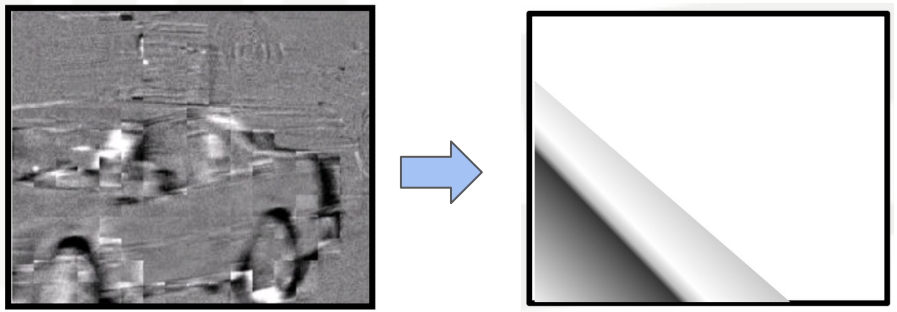




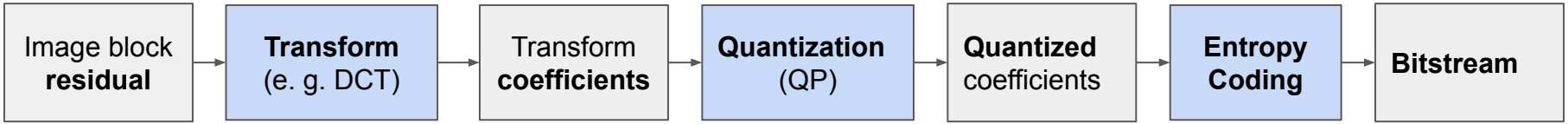
Inter prediction

Transform Coding

- Residual block signals are linearly **transformed** to achieve **decorrelation**.



- Transform coding steps (**encoder**):



- The **decoder** performs the **inverse** steps from the bitstream to the residual signal.

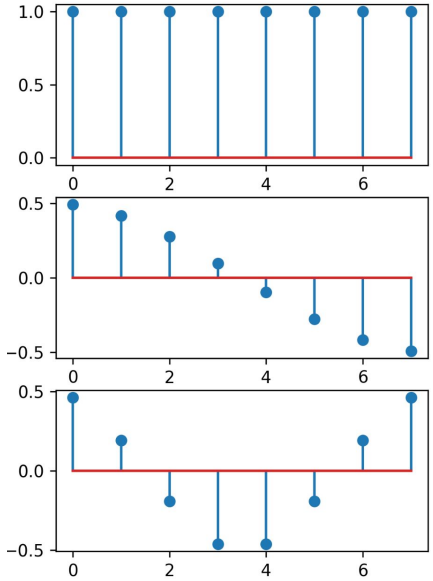
The Discrete Cosine Transform

- The DCT (DCT-2) is widely used in compression.
- It is a core component of **JPEG**.

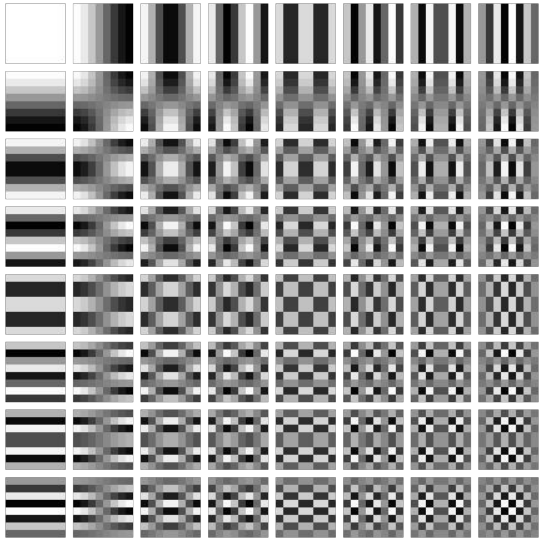
$$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$$

where $\omega_0 = (i == 0)\ ?\ \sqrt{\frac{2}{N}} : 1$

One-dimensional basis coefficients



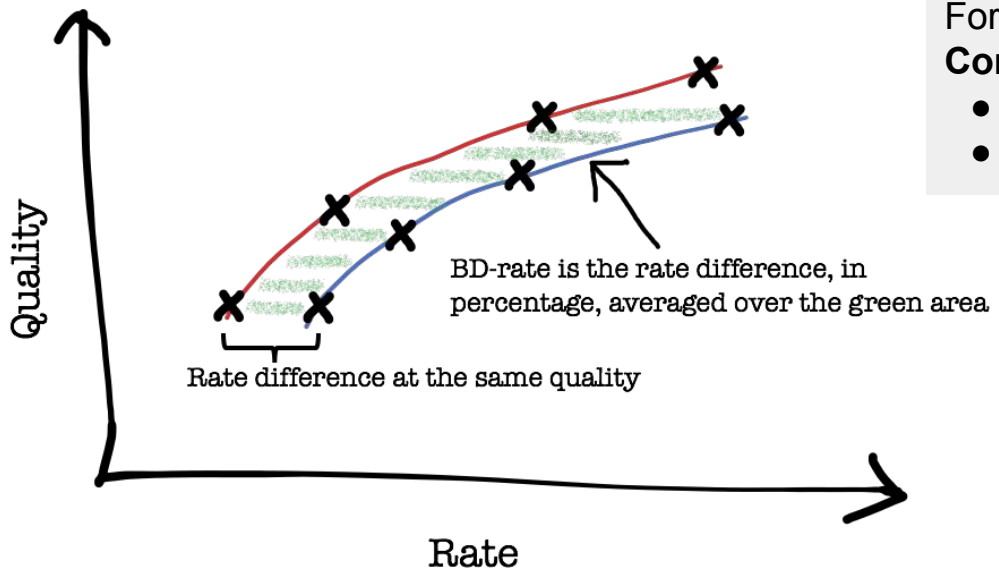
One-dimensional 8-point basis



Two-dimensional 8-point DCT basis

Comparing Video Codecs

- The performance of a compression method is measured by its **rate-distortion curve**.
- To compare two curves, the **BD-rate** is utilized.



For standardization we use the **AV2 Common Test Conditions**:

- Test configurations.
- Video sequences.

Source: [Netflix Tech Blog]

Unitary Transforms

The Graph Fourier Transform

- **Laplacian** Matrix: $L = D - A + S$ = *(degree) - (adjacency) + (self-loops)*

Graph Fourier Transform.
Given the Laplacian L, the GFT is the orthogonal matrix U defined by the eigendecomposition

$$L = U \Lambda U^{\mathsf{T}}$$

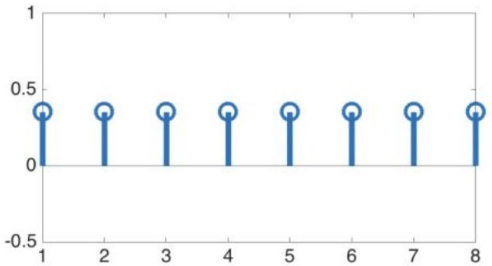
with the unitary matrix condition $UU^{\mathsf{T}} = I$

- Why? **Images** can be interpreted as graphs:
  - One **node** per **pixel**.
  - **Edges** between pixels that are **spatially contiguous**.
  - The graph **signal** is the pixel value or **intensity**.

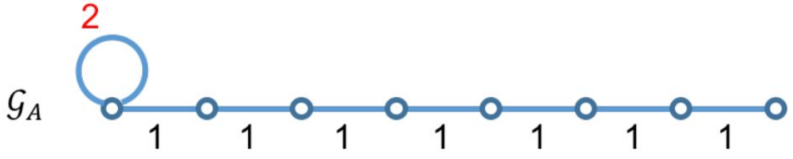
Common Graph Transforms

Discrete Cosine Transform (DCT-2)

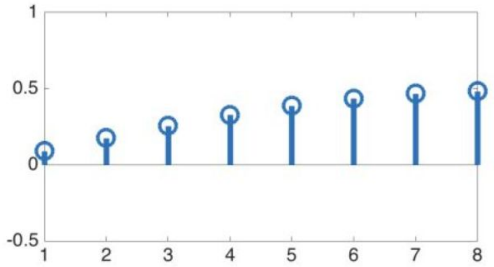
$$\mathbf{L}_D = \begin{pmatrix} 1 & & & & & & & \\ & 2 & & & & & & \\ & & \ddots & & & & & \\ & & & 2 & & & & \\ & & & & 1 & & & \end{pmatrix} - \begin{pmatrix} 0 & 1 & & & & & & \\ 1 & 0 & 1 & & & & & \\ & 1 & \ddots & \ddots & & & & \\ & & \ddots & 0 & 1 & & & \\ & & & 1 & 0 & & & \end{pmatrix} + \begin{pmatrix} 0 & 0 & & & & & & \\ 0 & 0 & 0 & & & & & \\ & 0 & \ddots & \ddots & & & & \\ & & \ddots & 0 & 0 & & & \\ & & & 0 & 0 & & & \end{pmatrix}$$



Asymmetric Discrete Sine Transform (ADST, DST-4)



$$\mathbf{L}_A = \begin{pmatrix} 3 & & & & & & & \\ & 2 & & & & & & \\ & & \ddots & & & & & \\ & & & 2 & & & & \\ & & & & 1 & & & \end{pmatrix} - \begin{pmatrix} 2 & 1 & & & & & & \\ 1 & 0 & 1 & & & & & \\ & 1 & \ddots & \ddots & & & & \\ & & \ddots & 0 & 1 & & & \\ & & & 1 & 0 & & & \end{pmatrix} + \begin{pmatrix} 2 & 0 & & & & & & \\ 0 & 0 & 0 & & & & & \\ & 0 & \ddots & \ddots & & & & \\ & & \ddots & 0 & 0 & & & \\ & & & 0 & 0 & & & \end{pmatrix}$$

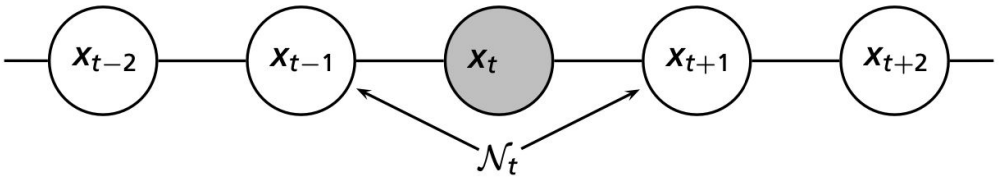


Probabilistic interpretation and Graph Estimation

- Graph ⟷ Gaussian Markov process (GMRF)
- Large **edge weight** ⟷ high **correlation** between pixels.
- Designing graph weights ⟷ parameter estimation for a GMRF.

An **AR-1** process is an example of a simple GMRF

$x_t = ax_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim \mathrm{N}(0, \sigma^2)$ and independent.



The maximum likelihood estimator of the **Graph Laplacian** from a sample covariance matrix ***S***

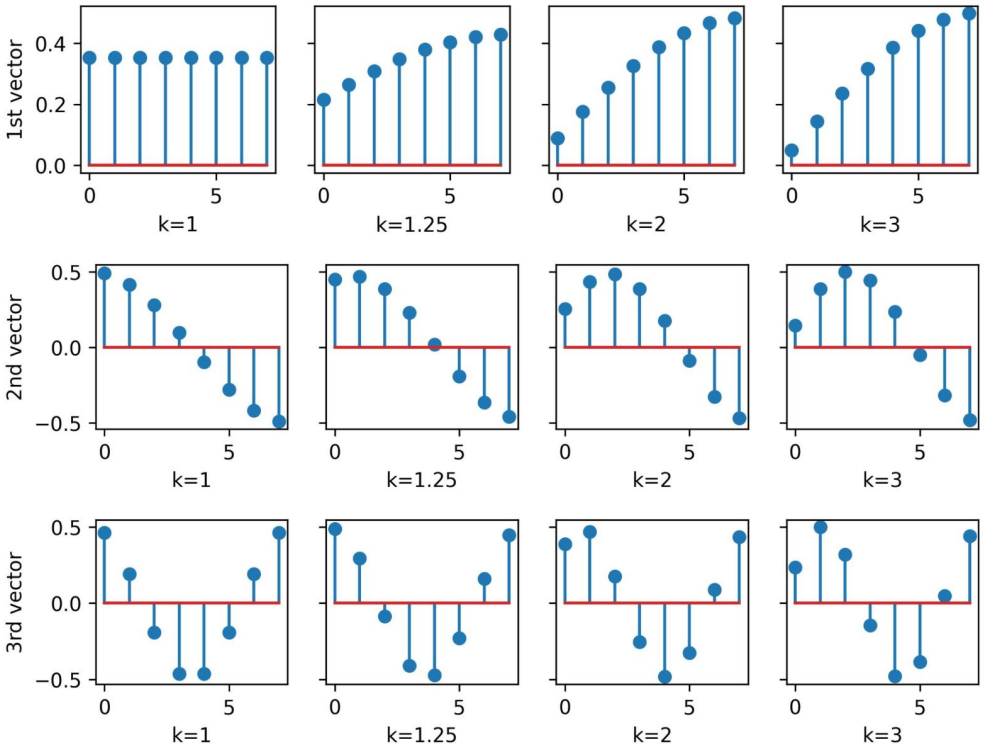
$$\min_{\boldsymbol{\theta}} \quad \mathrm{L}(\boldsymbol{\theta})\mathrm{S} + \log\det \mathrm{L}(\boldsymbol{\theta})$$
$$\text{s.t.} \quad \mathrm{L}(\boldsymbol{\theta})\mathbf{1} \geqslant \mathbf{0}$$

Line Graph Transforms

- We refer to the **Parametric Line Graph** defined by the Graph Laplacian as:

$$L(k,a) = \begin{bmatrix} k\,a & -a & \dots & & 0 \\ -a & 2a & -a & & \\ \vdots & & \ddots & & \\ & & -a & 2a & -a \\ 0 & & \dots & -a & a \end{bmatrix}$$

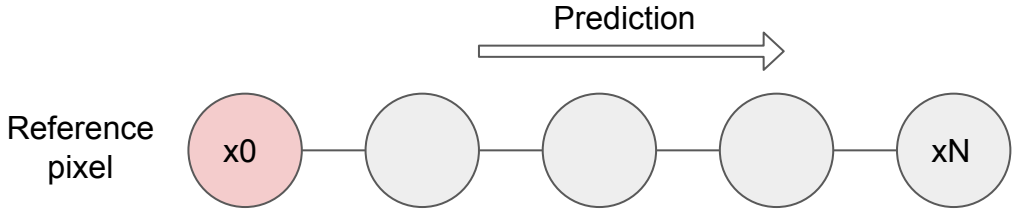
- The value **k** allows to better fit different **sidedness** of the residual signal.



First basis vectors for different **k**.

Graph models for intra-prediction residuals

- **1D** model for intra-prediction, which defines a signal for a **1D transform**.



- Assume an **AR-1** image signal. $x_k = \rho x_{k-1} + e_k$
- Calculate the residual as ***y* = *x* - *x0*** and find the covariance matrix $S = E[yy^T]$
- We find that the **precision matrix** (inverse of covariance) coincides with the Laplacian of the ***Discrete Sine Transform*** (for AR-1 with large autocorrelation).

$$\text{Prec}(0.999) = \begin{bmatrix} 2.0 & -0.999 & 0 & -0.001 \\ -0.999 & 2.0 & -0.999 & -0.001 \\ 0 & -0.999 & 2.0 & -1.0 \\ -0.001 & -0.001 & -1.0 & 1.027 \end{bmatrix}$$

$$L_{DST-7} = \begin{bmatrix} 2 & -1 & \dots & & 0 \\ -1 & 2 & -1 & & \\ \vdots & & \ddots & & \\ & & -1 & 2 & -1 \\ 0 & & \dots & -1 & 1 \end{bmatrix}$$

Model with quantization noise (*lossy model*)

- Consider that the reference pixel has a noise term $\hat{x}_0 = x_0 + \delta$ with variance σ^2
- The effect on the covariance matrix is an added term σ^2 to each of the matrix's entries.
- Consider a Parametric Line Graph, ***L(k)***. We estimate the optimal parameter and find

$$k = 2 - \frac{\sigma^2}{\sigma^2 + 2(1 - r)}$$

Analysis

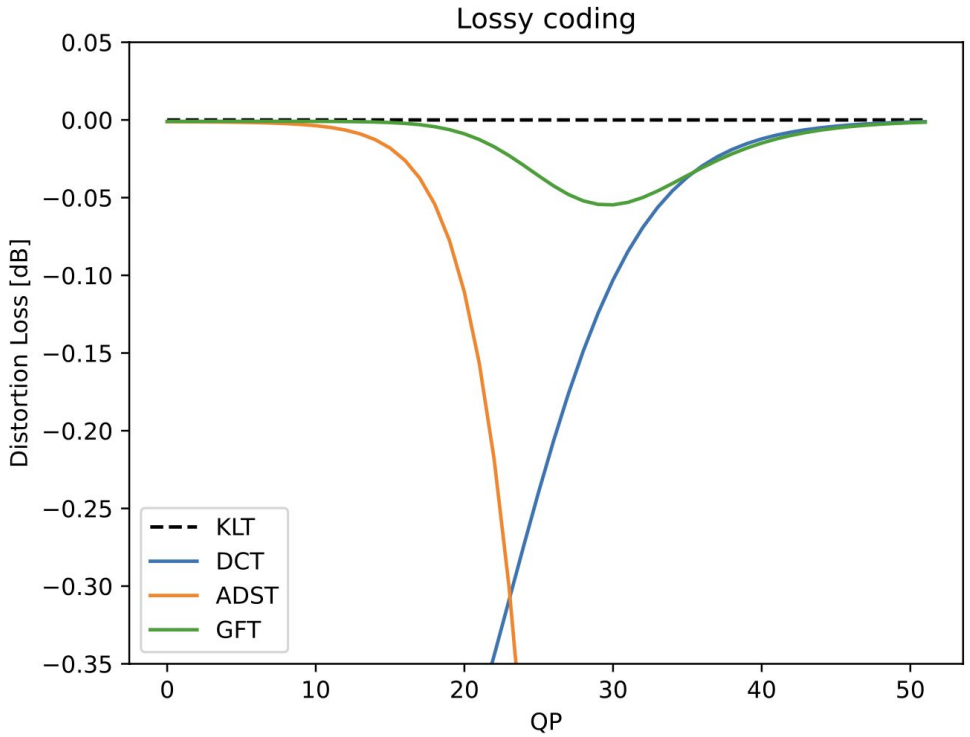
- In a lossless setting (**σ = 0**), the optimal **k = 2**, corresponding to the DST-7.
- When the noise is large (**σ >> 0**), the optimal **k → 1**, corresponding to the DCT.

Quantitative Analysis of the *lossy model*

- The **Karhunen-Loeve Transform** (KLT) is the best decorrelating transform for any given signal.
- We compare the **distortion loss** with respect to the optimal KLT.

Results

- For **low** noise, the **ADST** is close to the optimal.
- The **DCT** is optimal for **high** noise.
- The optimal **GFT** is able to better decorrelate the signal for **intermediate** noise.



Irregular Transforms

The Q-product and the Irregular-aware GFT

- We introduce a generalized inner product by means of a ***diagonal matrix Q***.

$$\langle x, y \rangle_Q = y^\top Q x$$ with norm $$\|x\|_Q^2 = x^\top Q x$$

- Each node has a diagonal element assigned, q_i which is interpreted as the **node weight**.

The ***(L,Q)-GFT*** is defined by the generalized eigenvalue problem

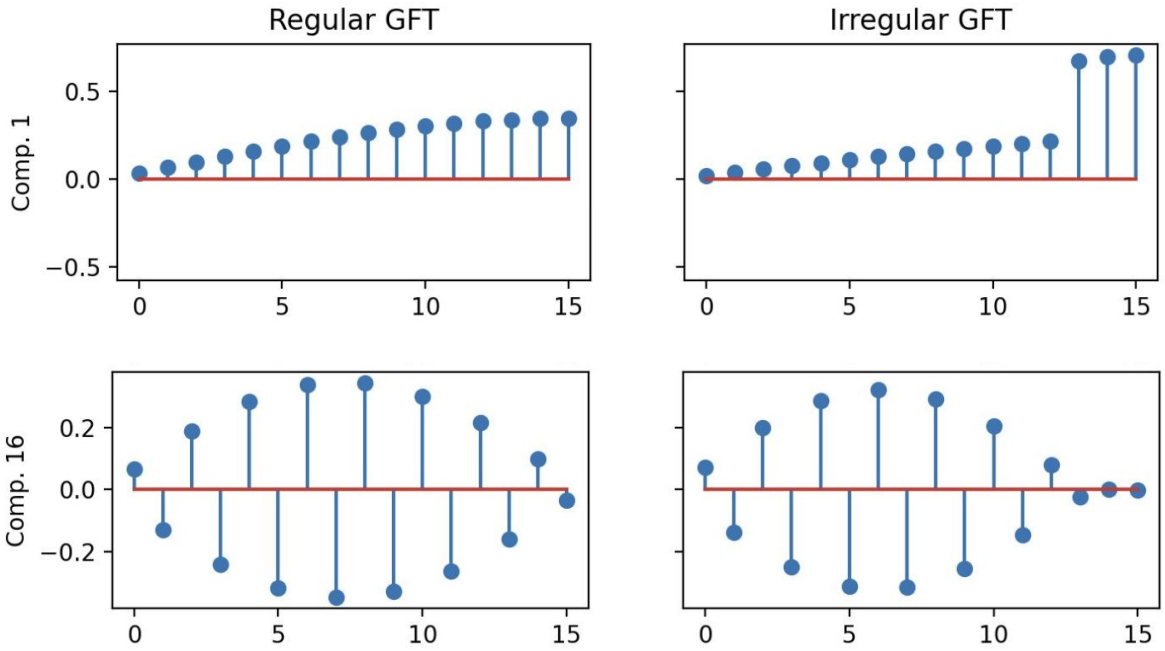
$$L u_i = \lambda_i Q u_i$$

with the ***Q-orthogonality*** condition $U^\top Q U = I$

- A ***Parseval's identity*** is satisfied with respect to the ***Q-norm.*** $\|\hat{x}\|_I^2 = \|x\|_Q^2$

Examining the effect of the *Q*-product

- We set the weight of the **last three** elements to be **x3** as large.
- The information of **important** pixels is captured in the **first** basis components.



Transform coding with (*L*,*Q*)-GFT

- Assume that the frequency coefficients are quantized using a **uniform quantizer**

$$\epsilon_f(i) \sim \text{Unif}(-\Delta/2, \Delta/2)$$

- If a ***(L,Q)-GFT*** is used, the resulting variance in the pixel domain is

$$\sigma^2_{\Delta,i} = \frac{\Delta^2}{12 q_i}$$

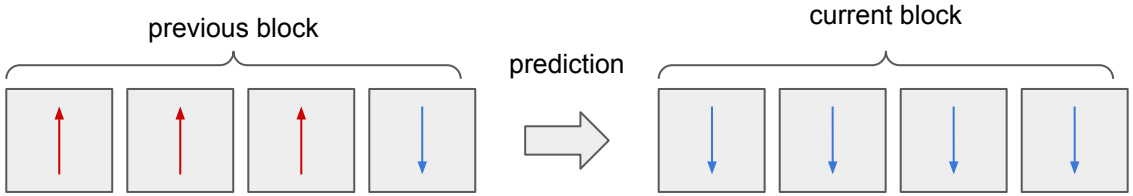
which can be interpreted as **tuning** the quantization step **per pixel**.

- For practical purposes, we need to impose a normalization of the weights

$$\sum_{i=1}^{N} q_i \leqslant N$$

Reducing the noise in the last pixel

- In intra-predicted blocks, **last pixel** is used as **reference** for the next block.
- The idea is to weight the last pixel favorably to **reduce the distortion of the next block**.



- From Parseval's identity, we can derive an optimization problem by minimizing the **total energy of the signal**

$$\sum_{i=1}^{N} \epsilon_t^2(i) = \sum_{i=1}^{N} q_i \left(\sigma_i^2 + \frac{\sigma_\Delta^2}{q_N} \right)$$

The resulting weights satisfy

$$q_N = \sqrt{1 + \frac{\sigma_\Delta^2}{\sigma_s^2} N}\, q_j$$

Results of a practical implementation

- The **PSNR** BD-Rate results are shown (**negative** means **compression gain**).
- The implementation has an **interaction** with another compression tool, the **Secondary Transform**.

Class	BD-rate Y	BD-rate U	BD-rate V
A1_4K	0.06	-0.09	-0.05
A2_2K	0.08	-0.42	0.26
A3_720p	0.08	0.44	0.27
B1_SYN	0.04	0.65	0.86
overall	0.07	0.15	0.30

Sec txfm ON

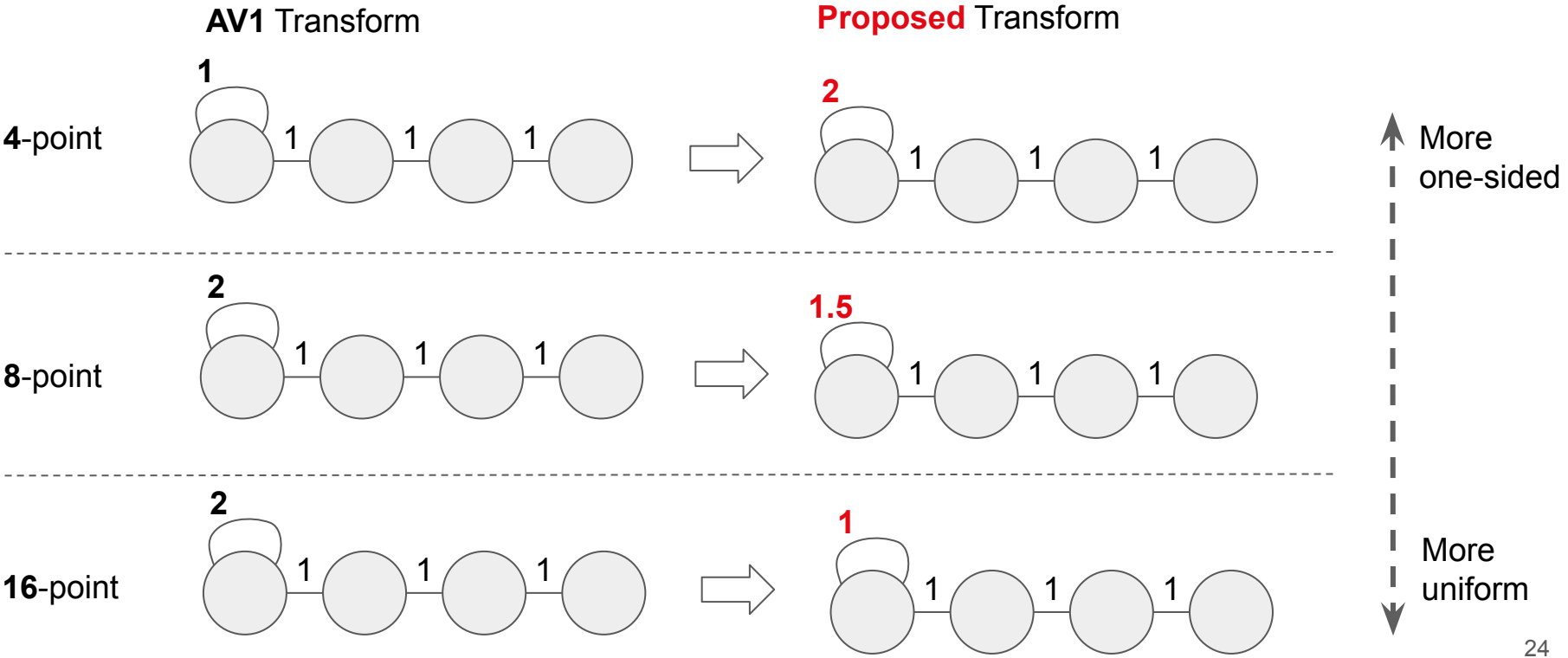
Class	BD-rate Y	BD-rate U	BD-rate V
A1_4K	-0.12	-0.17	-0.09
A2_2K	-0.10	1.30	1.13
A3_720p	-0.15	1.64	0.24
B1_SYN	-0.14	1.10	0.02
overall	-0.13	0.93	0.29

Sec txfm OFF

An Implementation for AV2

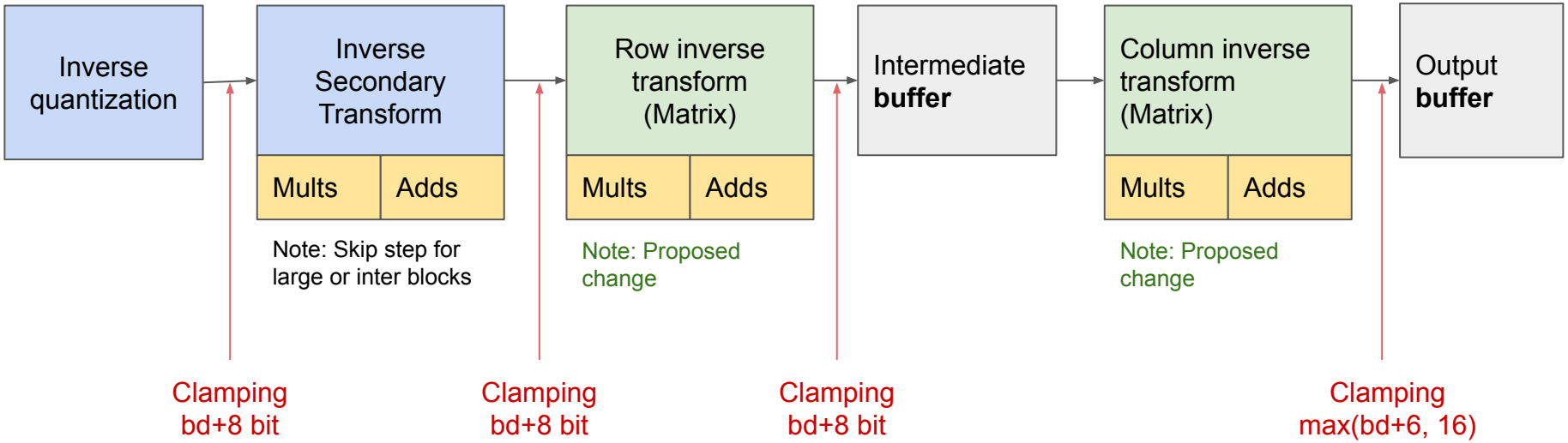
Description of the contribution

- A **replacement** of the **ADST bases** is proposed. The **graphs** are shown.



The Transform Pipeline

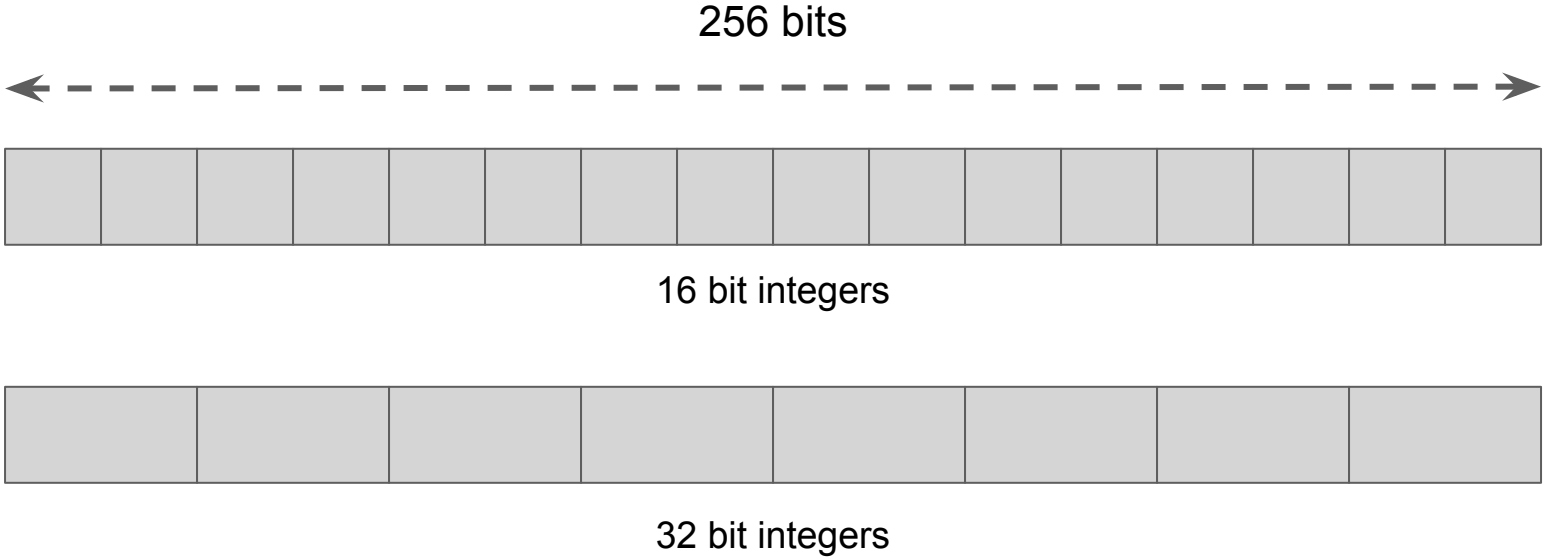
- The inverse transform pipeline is described.



Mults	Multiplier size: **(bd+8) x 8** bit
Adds	Addition size: **(bd+20)** bit

Improving the performance using SIMD

- **Single-instruction, multiple-data (SIMD)** can be used to speed-up in a normal CPU.



x16 parallel operations

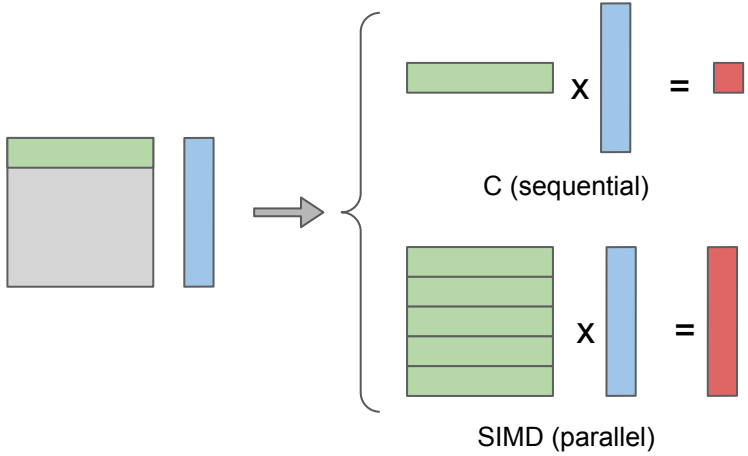
x8 parallel operations

Also, **faster** multiplications:

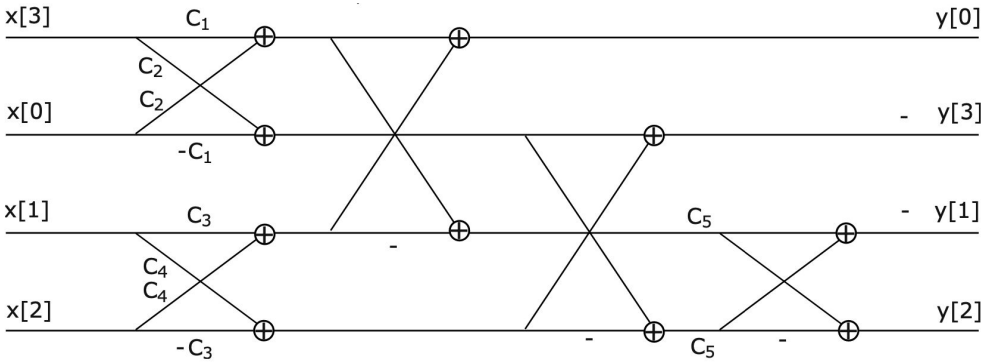
- 16 bit mult in **5 cycles**.
- 32 bit mult in **10 cycles**.

Algorithmic Optimizations

- Parallel matrix multiplication



- **Butterfly** calculations (similar to a ***Fast Fourier Transform***)



Results: Compression efficiency

	Algorithm	Compression Gain	Decoder Speed
4-point	Butterfly	-0.08%	100%
8-point	Matrix Multiplication	-0.03%	100%
16-point	Matrix Multiplication	-0.06%	100%
Total	-	**-0.16%**	**100%**

- The Compression Gain is expressed in terms of PSNR BD-Rate on all-intra configuration.
- A noticeable **compression gain** is achieved at **no decoder speed cost**.

Conclusions

- Described the potential of **Line GFTs** to generalize the DCT and ADST.
- Found a **theoretical model** for the **optimal parametric line graph** for video coding.
- Developed a model for video compression using **irregular transforms** and proved the potential to **improve compression**.
- Designed and developed an **efficient implementation for AV2**.
- The work is accepted as a ***Conditional Candidate*** tool.

Other contributions at Netflix:

- Efficient implementation of an **intra prediction mode** for AV2.
- Explored an improvement of **run-length entropy coding** for AV2.

Extra Material

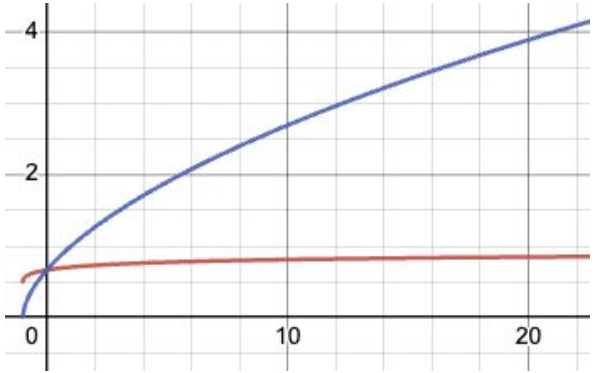
Analysis of the optimal weights

- The weights depend on a **signal-to-noise ratio.**

$$q_N = \sqrt{1 + \frac{\sigma_\Delta^2}{\sigma_s^2} N}\, q_j$$

- Pixels **1 to N-1** have the **same weight**.
- Pixel **N** has $\sqrt{N}$ times **more weight**.

Pixel weight



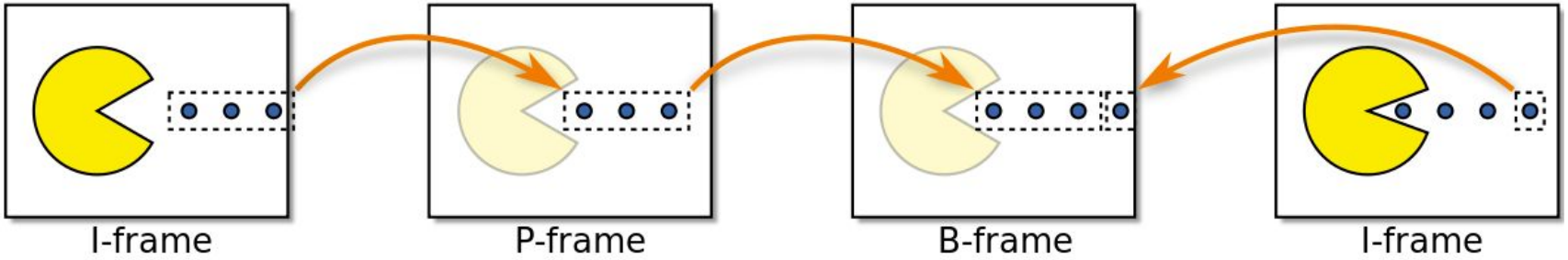
Transform size (N)

- We can quantify the improvement of the optimal weights with $E_{diff} = E_I - E_Q$

which save energy $E_{diff} = \sigma_s^2 \frac{N-1}{N} \left(\sqrt{1 + \frac{\sigma_\Delta^2}{\sigma_s^2} N} - 1 \right)^2$

Frame Types

I-frame: Intra-predicted	**P-frame:** Predicted frame	**B-frame:** Bi-predictive
Self-contained	Reference ***previous*** frames	Reference ***previous and future*** frames
High bit-rate cost	Cheaper rate cost	Cheapest rate cost



- A video sequence is build by **combining all types** of frames.