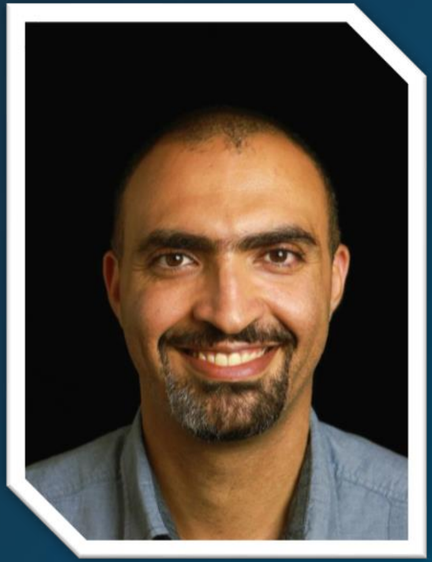
MHV-2023: Amazon Technical Session

Perceptually motivated compression efficiency for live encoders

Ramzi Khsib

Principal software Engineer
AWS Elemental



Ramzi Khsib

Principal Software Engineer
AWS Elemental

Agenda

- Introduction
- Machine learning driven encoder configuration
- Video input pre-filter
- Results
- Conclusions

Improve video compression efficiency

Improve video compression efficiency:

- Novel coding tools
- Optimize coding tools
- Specific scenarios optimization: QVBR, per title, etc.
- Pre-filter input video

Novel coding tools

New generation of codecs challenges:

- New decoder
- Diminishing returns
- Compute cost

Live applications challenges

Compression efficiency in live encoders is challenging:

- Realtime processing
- Limited compute power
- Limited latency

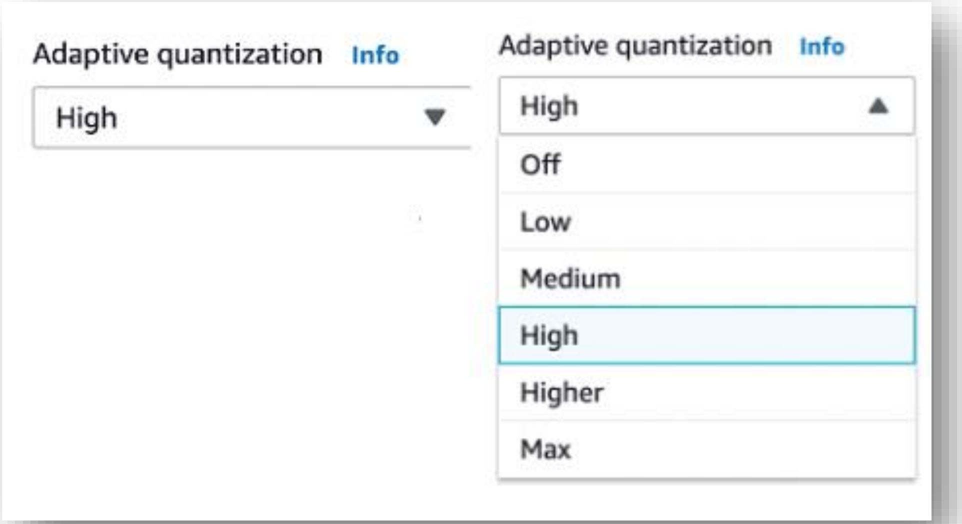
Real Time ML driven dynamic encoder configuration

Video encoder configuration

Configurable with command line options

One among many settings is selected

Varying impact on efficiency



Optimal encoder configuration

- ~20 options to configure
- Highly content dependent
- Search space is **astronomical**

Presets / encoding recipes

Simplify encoder optimization.

Pre-defined sets of configuration options

Optimized for specific content types:

- Film grain presets
- Screen content presets
- High motion presets

Presets in action

Video Asset

Film Grain Tuning

Quality	+++	---	-	+

Shortcomings of presets

- Limited granularity
- High level of expertise.
- Manual tuning

Content adaptive encoder limitations

- **Mainly Rate Control:** Adaptive bitrate (ABR)
- **VOD only**: limited application for live encoding

Need:

Dynamic content adaptivity in live applications

Content adaptive dynamic encoder configuration

Fine granularity encoder settings optimization

Content adaptive

Compression and rate control aware.

Dynamic presets in action

Video Asset



Sequence Level Tuning

Tuning Sequence #0

Tuning Sequence #1

Tuning Sequence #2

Tuning Sequence #3

Granularity

GOP #0 Tuning

GOP #1 Tuning

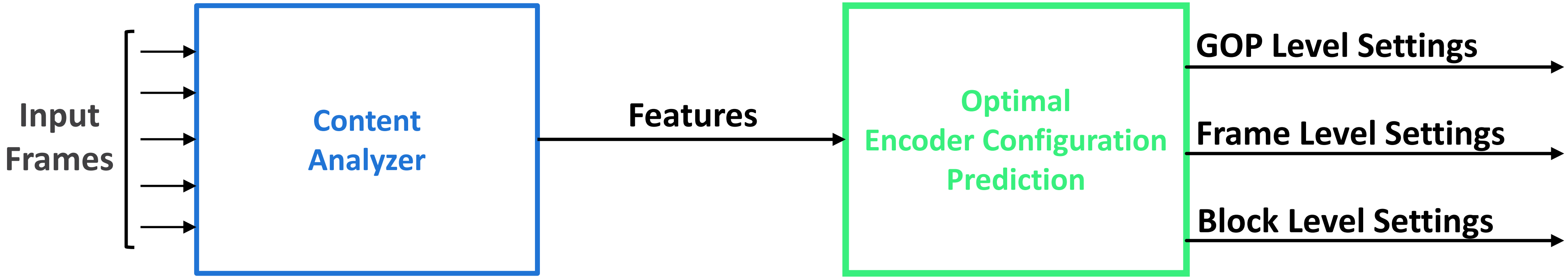
GOP #2 Tuning

Frame

Dynamic encoder configuration design principles

1. Optimize video quality
2. Dynamically adaptive to content
3. High throughput and low latency
4. Scalable and robust

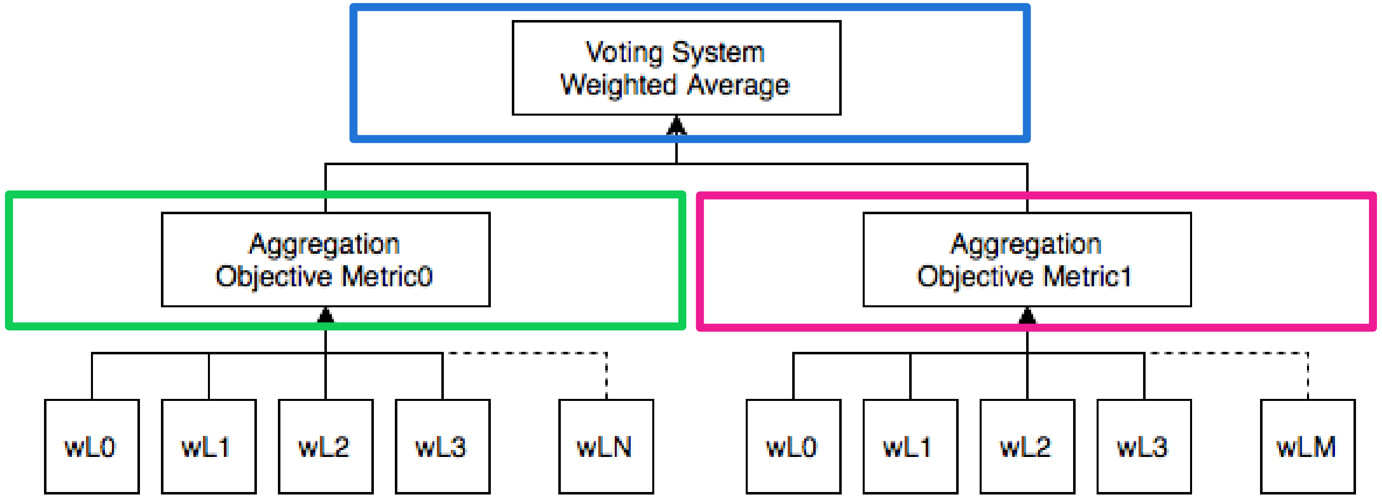
Content Adaptive Dynamic Encoder Configuration design



Content adaptation:

1. **Content Analysis:** Extract features from look ahead buffer
2. **Prediction:** ML driven optimal encoder configuration

Highly performant machine learning models



Meta ensemble :

- Ensemble learns from a selected objective metric
- Voting/blending uses learnt weighted average

State of the Art: Input Video Pre-Filters

Increase compression efficiency by reducing noise

Reduce video complexity:

- **M**otion **C**ompensated **T**emporal **F**ilter
- Filter original pixels

Input video pre-Filter: filter input video prior to compression

Input video pre-filter advantages

1. Codec agonistic
2. No decoder spec change
3. Significant efficiency increase
4. Noisy content: visually appealing

Input video pre-filter in video codecs

Called Temporal Filter in reference to MCTF

Idea not new:

- Proposed in VP8 and VP9
- **HEVC**: HM in version HM16.1
- **AV1**: AOM-AV1 SVT-AV1
- VVC
- AV2

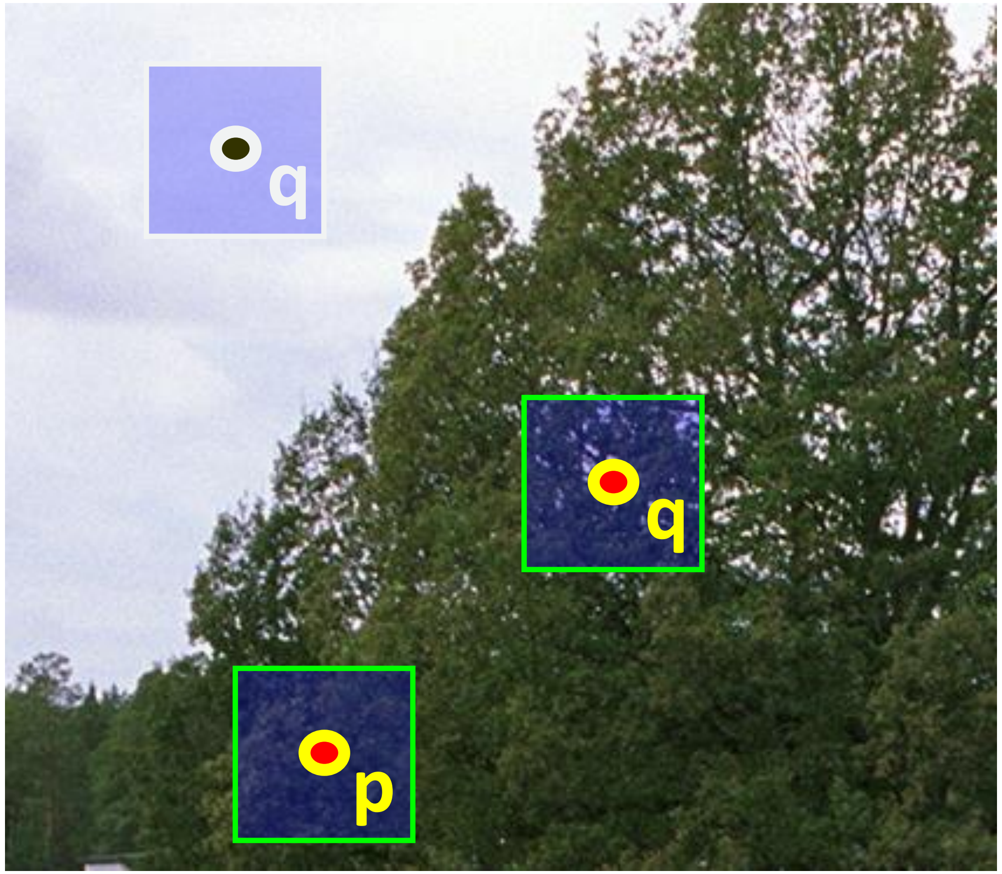
Pixel domain filter

The Bilateral filter:

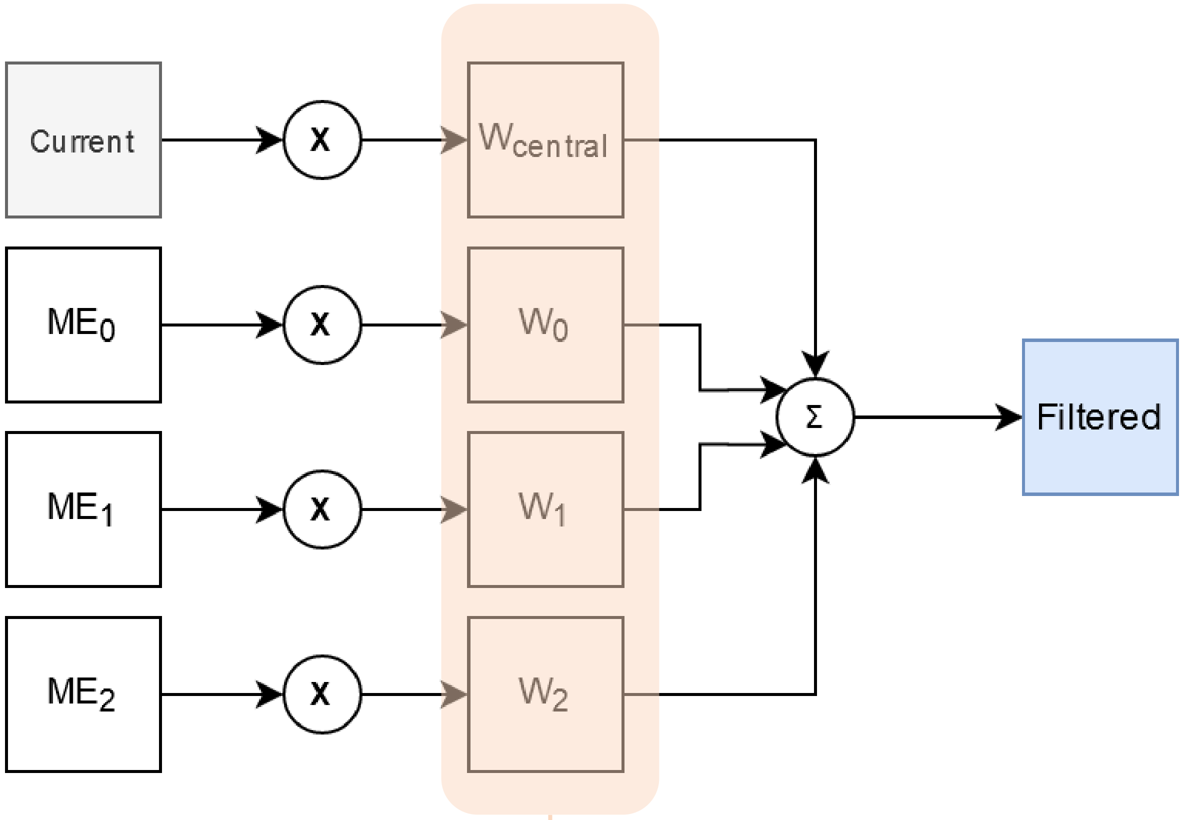
$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_{\mathrm{s}}}(||\mathbf{p} - \mathbf{q}||) \; G_{\sigma_{\mathrm{r}}}\left(||\vec{V}_{\mathbf{p}} - \vec{V}_{\mathbf{q}}||^2\right) \; I_{\mathbf{q}}$$

NLM removes the spatial component

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} \cancel{G_{\sigma_{\mathrm{s}}}(||\mathbf{p} - \mathbf{q}||)} \; G_{\sigma_{\mathrm{r}}}\left(||\vec{V}_{\mathbf{p}} - \vec{V}_{\mathbf{q}}||^2\right) \; I_{\mathbf{q}}$$



HM Temporal Filter: weighted Average



Weighting function

Replace I_o with the new pixel value, I_n

$$I_n = \frac{I_o + \sum_{i=0}^{3} w_r(i,a)\boldsymbol{I_r(i)}}{1 + \sum_{i=0}^{3} w_r(i,a)}$$

$\boldsymbol{w_r(i,a)}$, is defined as follows:

$$\boldsymbol{w_r(i,a)} = s_l s_o(n) s_r(i,a) e^{-\frac{\Delta I(i)^2}{2\sigma_l(QP)^2}}$$

With

$$\boldsymbol{\sigma_l(QP)} = 3 * (QP - 10)$$

$$\Delta I(i) = I_r(i) - I_o$$

LibAOM AV1 temporal filter

Keys differences to HM:

- NLM model
- Block based
- Adaptive noise level

Pixel based filter shortcomings

+ Improved efficiency in noisy content

- Not recommended for pristine content

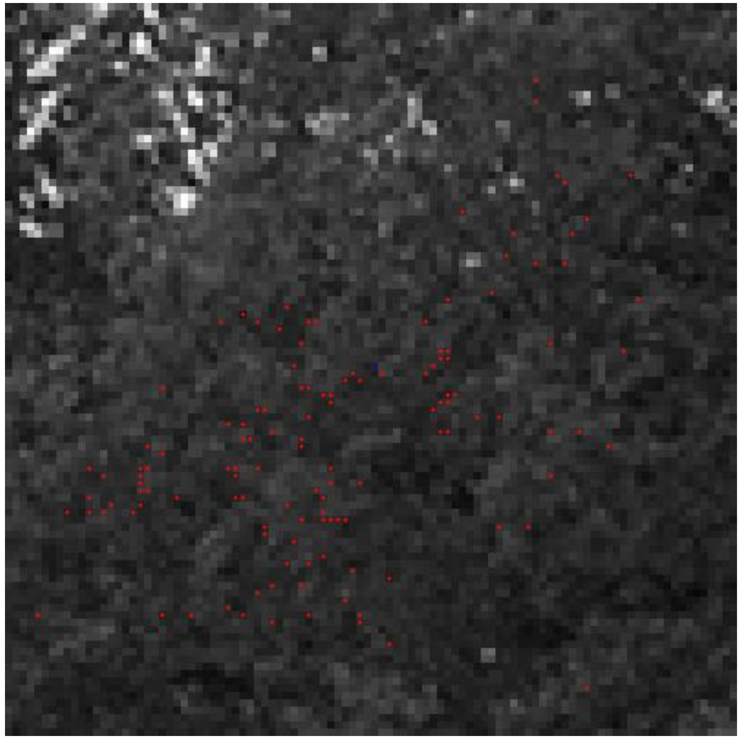
Pixel based filter visual issues

Non optimal filtering:

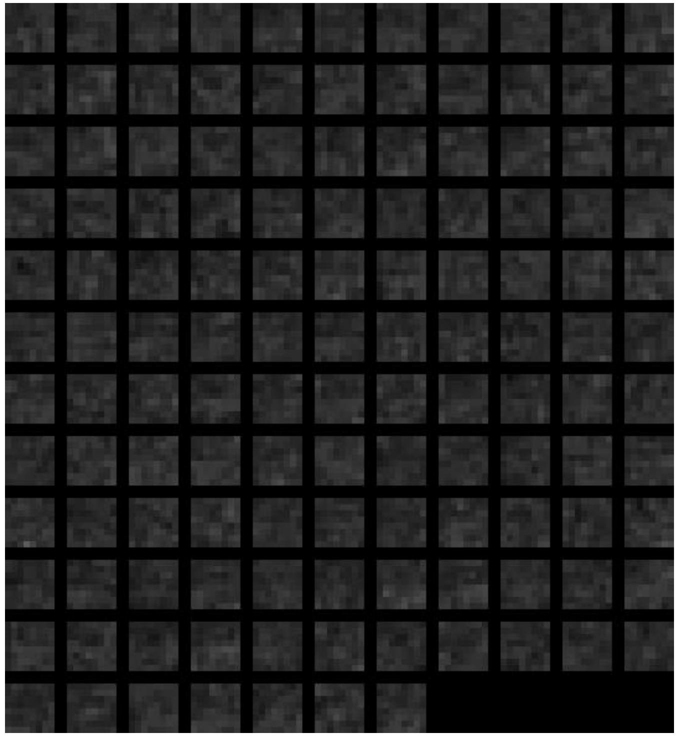
1. Visible sharpness loss
2. Visible blocking
3. Blurriness in texture/film grain

Transform domain collaborative filtering

Original

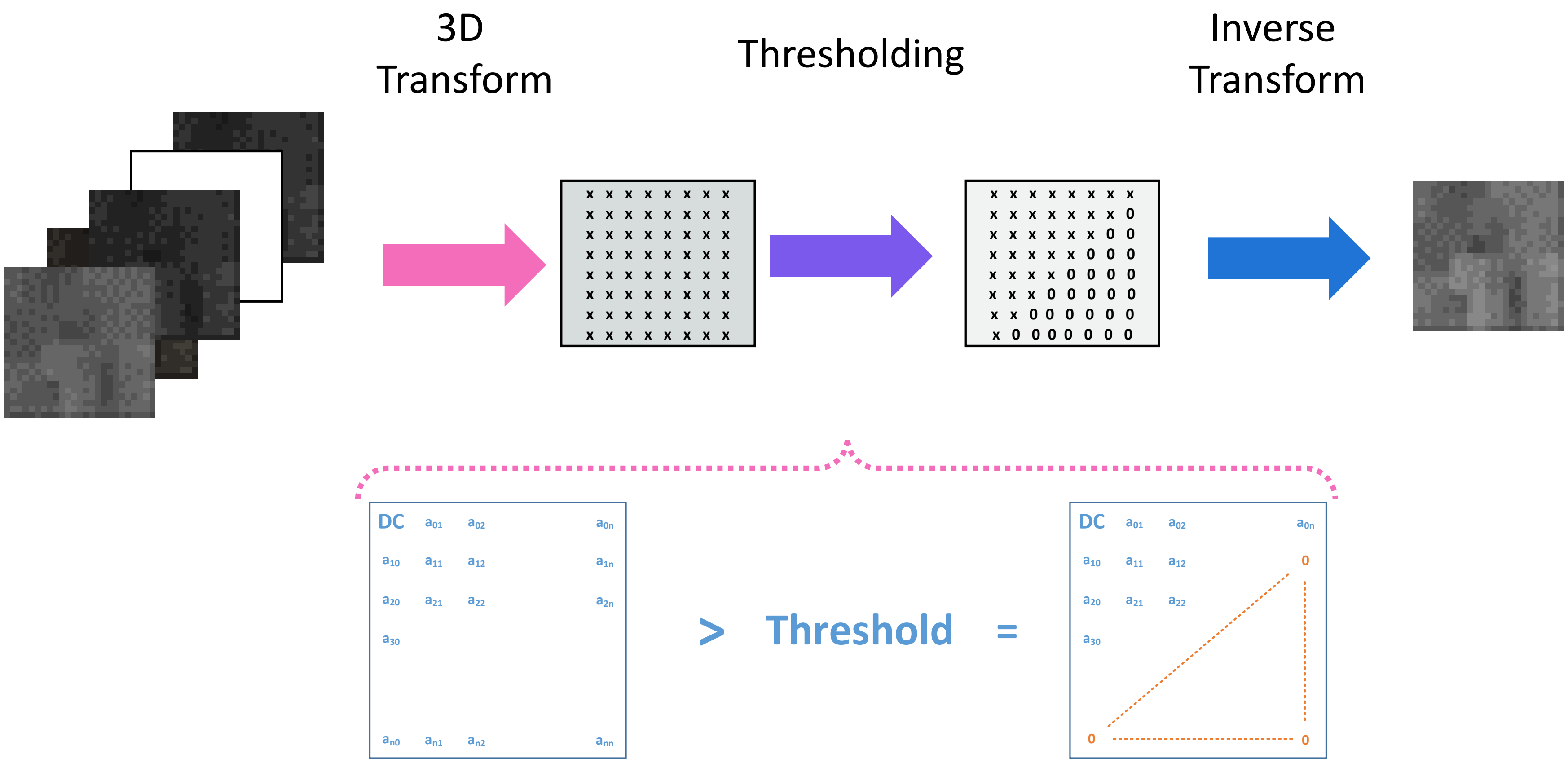


Patch Locations



Patches

Transform domain collaborative filtering



Pixel domain filters

Pros:

- Great bandwidth reduction

Cons:

- Quality depends on motion estimation
- Texture softness
- Compute cost

Transform domain filters

Pros:

- Better texture preservation

Cons:

- Hyperparameter tuning
- Compute

Deep learning based filtering

Pros:

- Greater filtering capabilities

Cons:

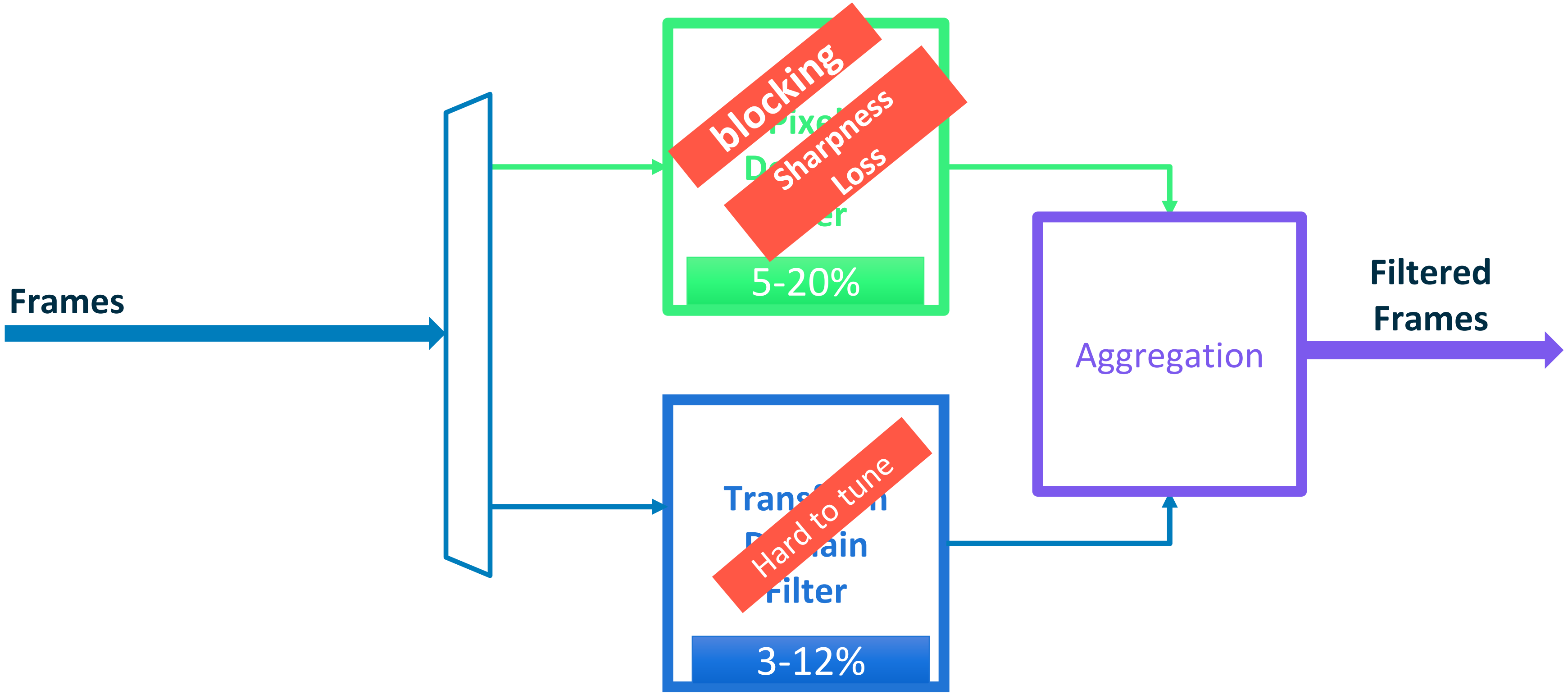
- Compute cost
- Higher latency
- Reduced throughput

Proposed Filter

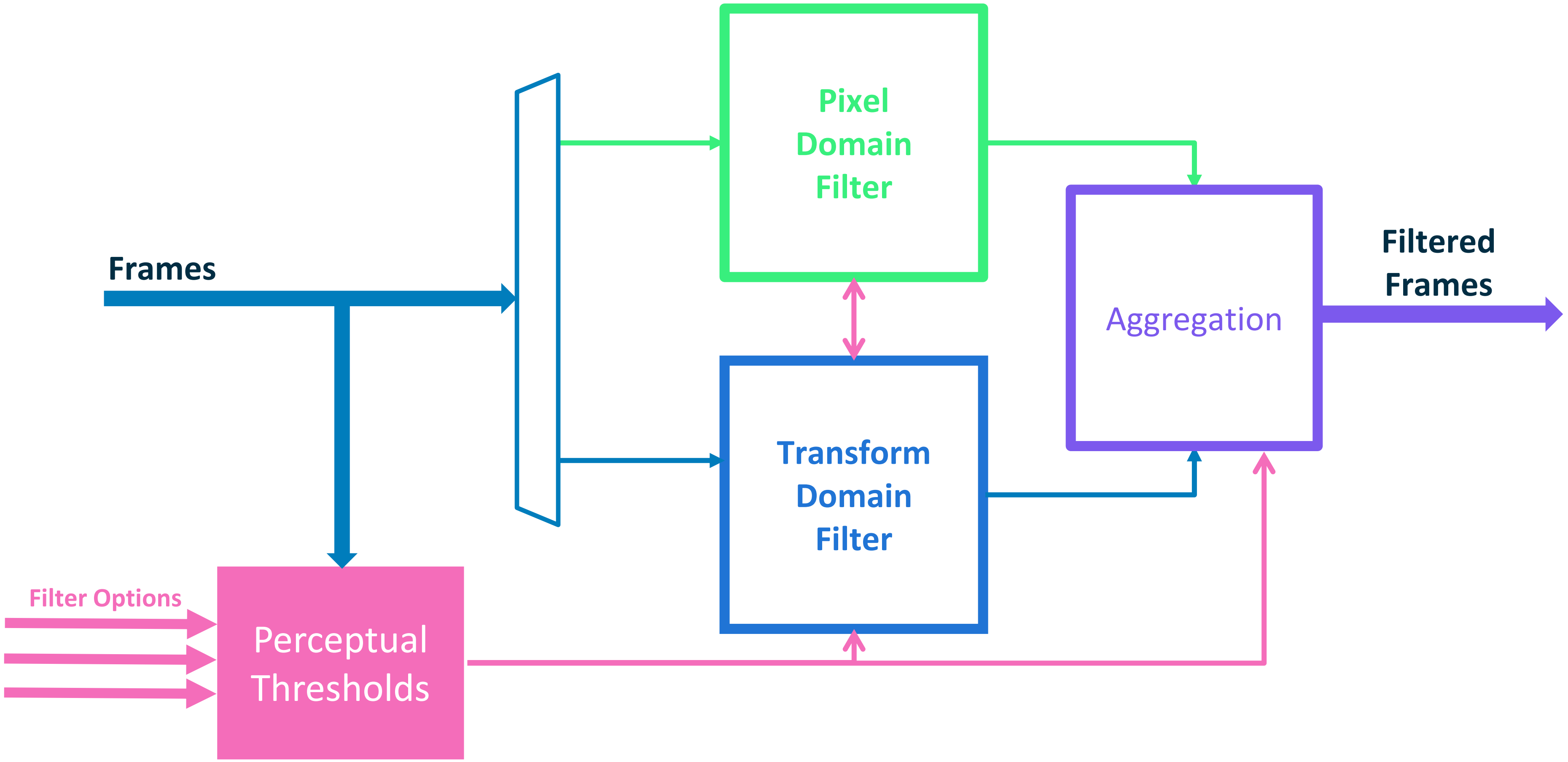
Filter design principles

- Dual domain filter
- Suitable for live encoders
- Preserve textures
- Reduce bandwidth
- Rate control aware

Dual domain filter architecture



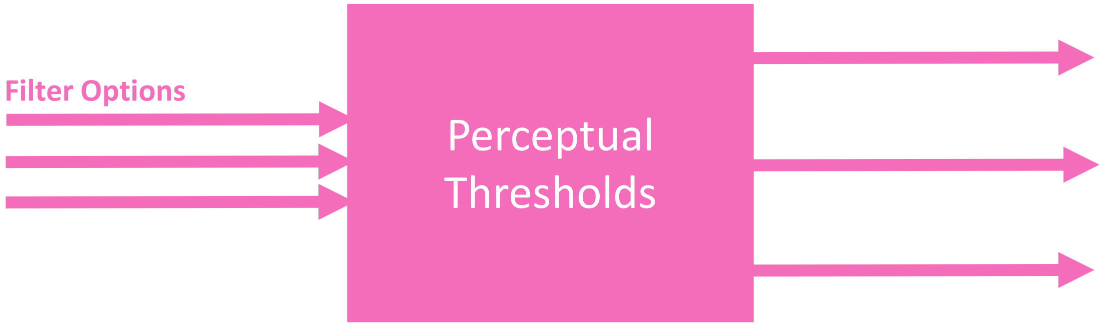
Dual domain filter architecture



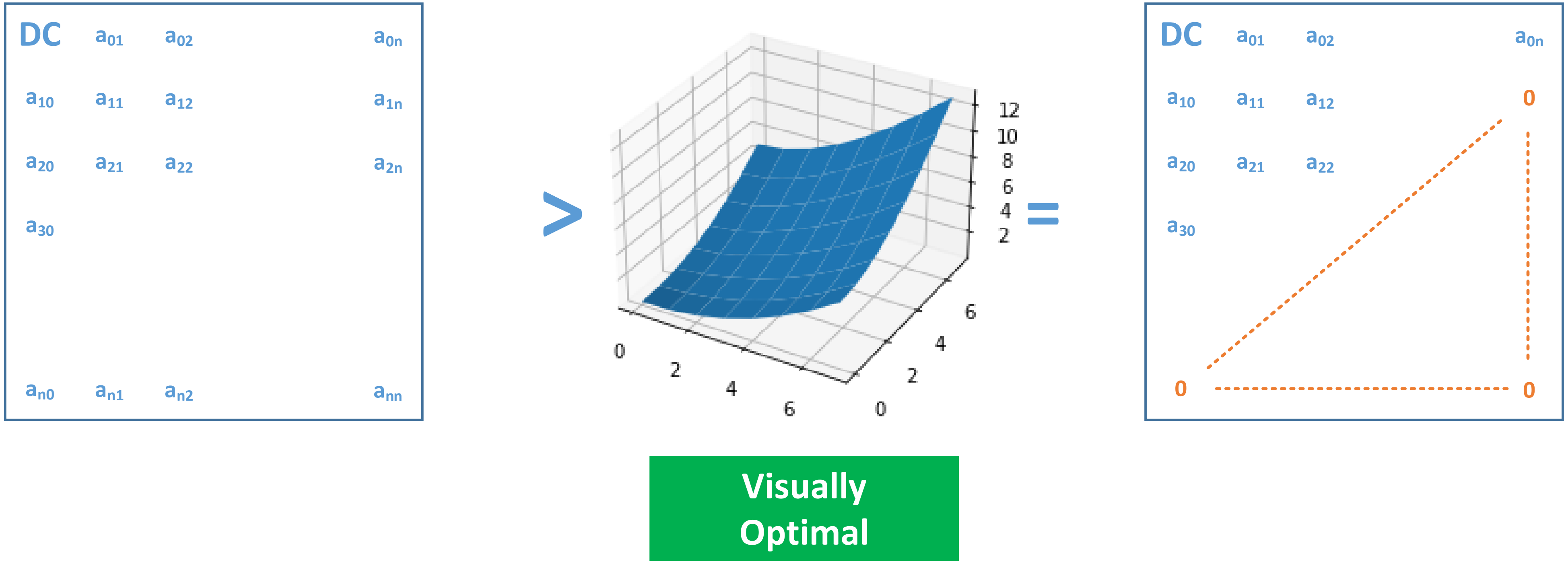
Perceptual thresholds module

Perceptual thresholds module computes:

- Filtering strengths
- Perceptual thresholds
- Aggregation weights



Perceptually guided thresholds

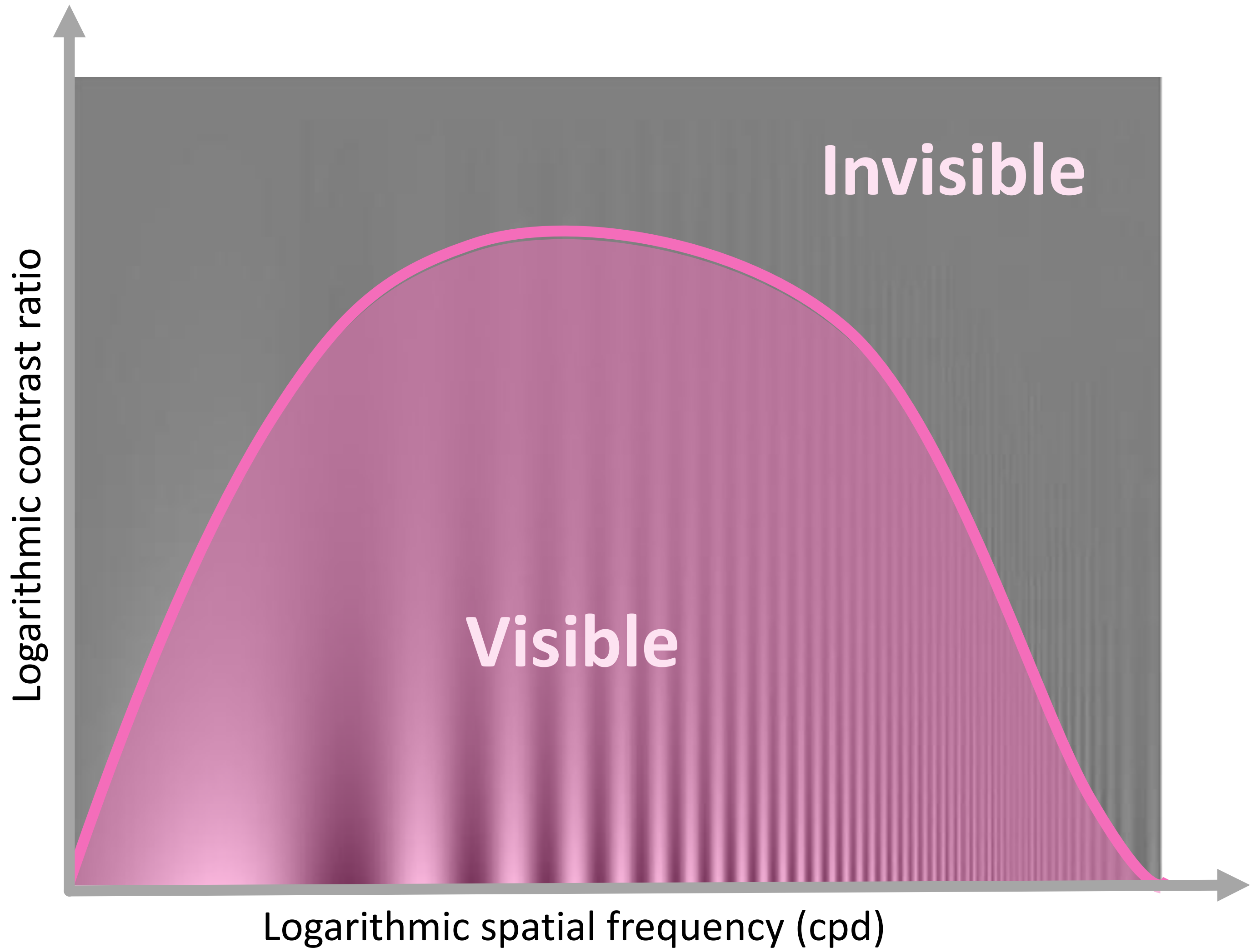


HVS motivated optimal filtering

Human vision system (HVS) models:

- Contrast sensitivity function
- Texture masking
- Luma adaptation
- Motion aliasing
- Artifacts visibility with quantization

Contrast sensitivity function



Texture masking

Quantization aware masking

Optimal filtering:

- In-loop filter
- Block based filter
- Rate control aware: multi models

Perceptual thresholds

Combine HVS models and block based filtering:

$$\text{PerceptualThresholds} = \boxed{f_{QP}}(\text{CSF}, \text{CM}, \text{LA}, \text{MOTION})$$

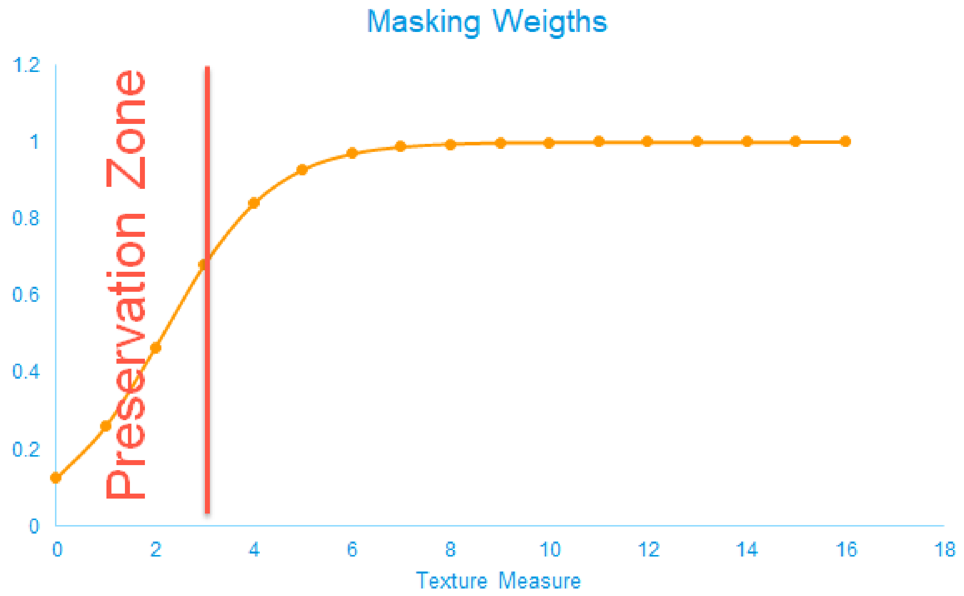
QP Based Multi-Model

Aggregation

Aggregate spatial / transform domain:

Block based sigmoid scaling:

$$Weight = \frac{c}{1 + b \times e^{(-Text \times a)}}$$



Results

Content adaptive dynamic encoder configuration

Computation

- Computation load: **< 2%** AVC encoding time
- Deployed in live application

Content adaptive dynamic encoder configuration (2)

Dynamic content:

- VMAF: **7% to 21%** BD-rate improvement

- Subjective Comparison: video experts (golden eyes)
  - Range **6% to 8%** with up 15% bandwidth reduction

Content adaptive dynamic encoder configuration (3)

OFF

Content Adaptive Dynamic Encoder Configuration

Input pre-filter

Computation

- AVC: **~18%** encoding time
- HEVC: **~13%** encoding time

VMAF: **8% to 15%** BD-rate improvement*

Subjective comparison (golden eyes) *

- Range **8% to 12%**

* Results are highly content dependent

Input pre-filter subjective results

High fidelity filtering: film grain, texture, edges etc.

Suitable for:

- Pristine
- Lightly distorted content
- Heavily distorted content

Conclusions

Conclusions

Suitable for live applications:

- Dynamic encoder configuration
- Input pre-filter

Substantial compression efficiency

Reduced latency, low compute cost

Thank you!

Ramzi Khsib

Khsramzi@amazon.com

ramzi@ieee.org