Quality Controlled Temporal Video Adaptation

Klaus Leopold

Introduction

- Temporal Scalability $\Leftrightarrow$ Frame Rate Reduction
  - Reduce bandwidth consumption
  - Reduce complexity in decoding
  - Cut out scenes
  - Content priorities
- Can be realized in the compressed domain
  - Can be handled in realtime
  - Good for intermediate network nodes
  - Easy to perform
- Motion in the video does not look smooth
- Video quality suffers if temporal scalability is performed permanently

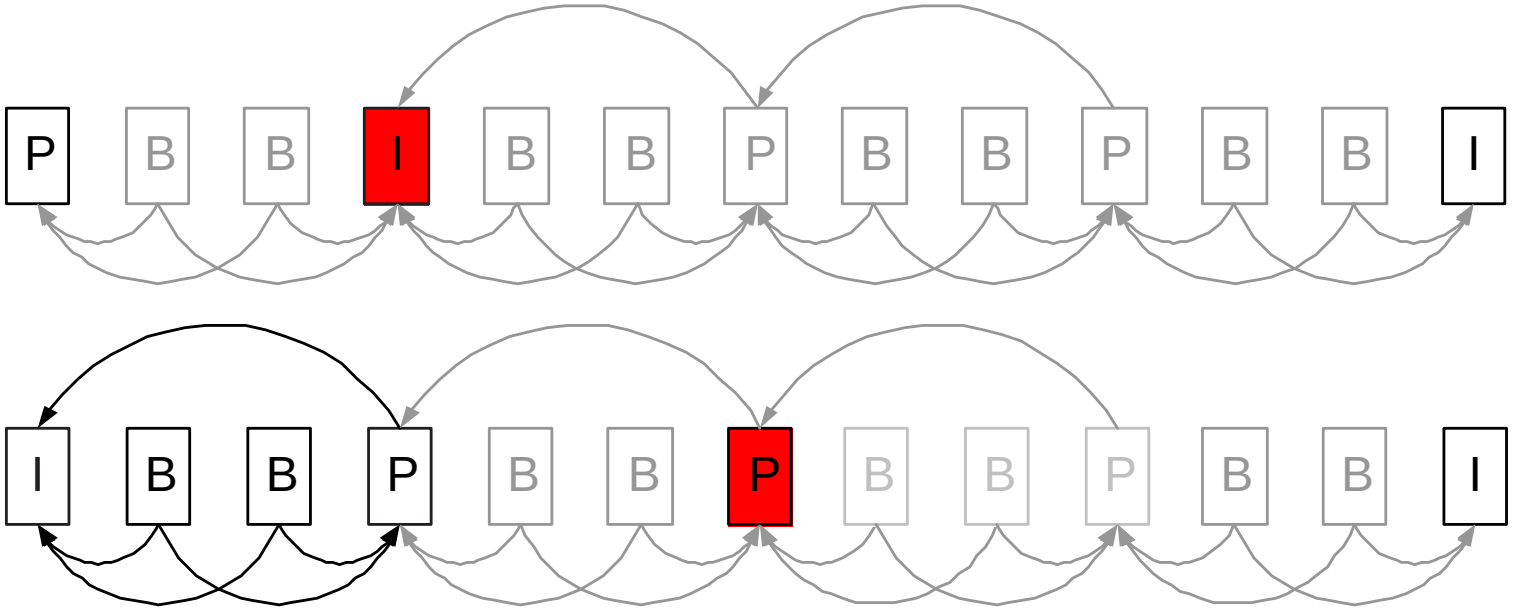
Theoretical Background (1/2)

- Three different frame types in MPEG-2/4:
  - I-Frames , *intra coded pictures*: Independent pictures
  - P-Frames , *predictive coded pictures*: Need previous I or P frame for encoding/decoding
  - B-Frames , *bidirectionally predictive coded pictures*: Need previous and next I or P-Frame for encoding/decoding

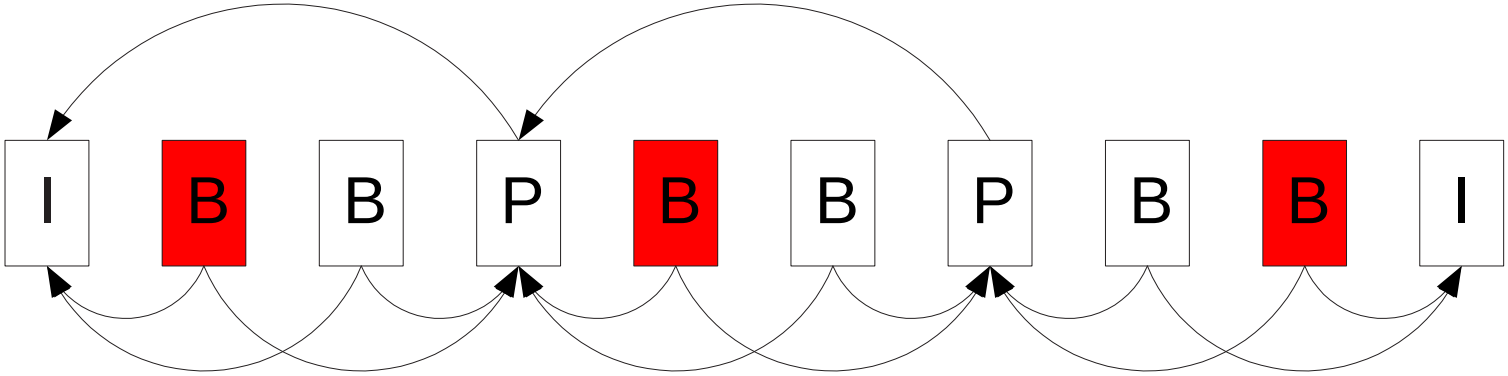
Theoretical Background (1/2)

- Three different frame types in MPEG-2/4:
  - I-Frames , *intra coded pictures*: Independent pictures
  - P-Frames , *predictive coded pictures*: Need previous I or P frame for encoding/decoding
  - B-Frames , *bidirectionally predictive coded pictures*: Need previous and next I or P-Frame for encoding/decoding

- Impact of frame dropping (MPEG-2/4):



Theoretical Background (2/2)



- Just B-Frame dropping makes sense
- Nowadays temporal scalability approaches: Frame dropping regardless of quality
- Aspects to take care of:
  - Quality after dropping
  - Timely distribution
  - Frame sizes
  - Importance

Quality Estimation for GOPs

- PSNR value for two single images where I is the original and L the loss induced image with dimension $X \times Y$

$$psnr = 20 \log_{10} \left(\frac{255}{\sqrt{\frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} (I_{x,y} - L_{x,y})^2}} \right)$$

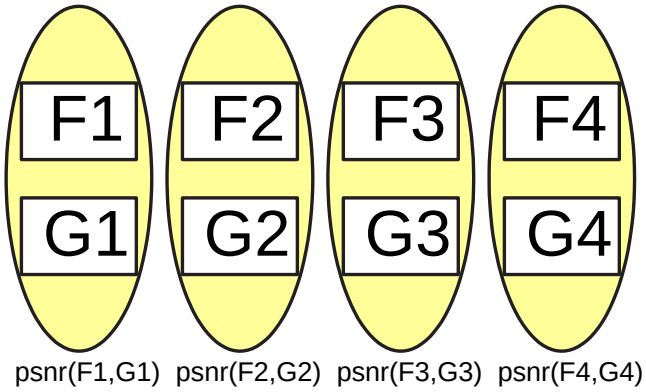
Quality Estimation for GOPs

- PSNR value for two single images where I is the original and L the loss induced image with dimension $X \times Y$

$$psnr = 20 \log_{10} \left(\frac{255}{\sqrt{\frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} (I_{x,y} - L_{x,y})^2}} \right)$$

- For two frame sequences (GOPs, patterns) F and G with n single frames $F = \{F_1, \ldots, F_n\}$ and $G = \{G_1, \ldots, G_n\}$, the quality Q_P is:

$$Q_P = \frac{\sum_{i=1}^{n} psnr(F_i, G_i)}{n}$$

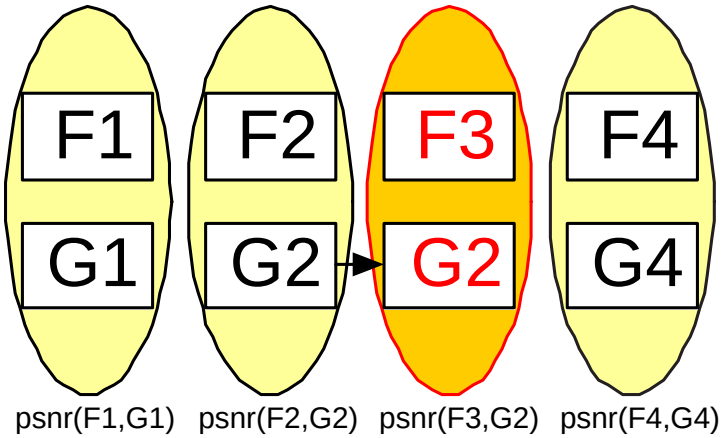


Quality Estimation for GOPs | Single Dropped Frame

Quality of two patterns F and G with n frames, where frame k is missing:

$$Q_P = \frac{\sum_{i=1}^{k-1} psnr(F_i, G_i) + psnr(F_k, G_{k-1}) + \sum_{j=k+1}^{n} psnr(F_j, G_j)}{n}$$

No_F	$Type_F$	No_G	$Type_G$	$psnr(F,G)$
1	I	1	I	35.342
2	B	2	B	33.993
3	B	3	B	33.984
4	B	4	B	34.191
5	B	5	B	34.032
6	P	6	P	35.561
7	B	6	P	14.432
8	B	8	B	34.331
9	B	9	B	34.531
10	B	10	B	34.667
11	B	11	P	34.123
			Overall Q_P	32.835



Quality Estimation for GOPs

Sequence of Dropped Frames

Given two patterns F and G with n frames where m frames are sequentially dropped starting with frame G_k, the quality Q_P is calculated with:

$$\forall\ k, m \in \mathbb{N} : k + m \leq n$$

$$Q_P = \frac{\sum_{i=1}^{k-1} psnr(F_i, G_i) + \sum_{l=k}^{k+m-1} psnr(F_l, G_{k-1}) + \sum_{j=k+m}^{n} psnr(F_j, G_j)}{n}$$

No_F	$Type_F$	No_G	$Type_G$	$psnr(F,G)$
1	I	1	I	35.342
2	B	2	B	33.993
3	B	2	B	23.984
4	B	2	B	19.191
5	B	2	B	14.032
6	P	6	P	35.561
7	B	7	P	34.432
8	B	8	B	34.331
9	B	9	B	34.531
10	B	10	B	34.667
11	B	11	P	34.123
			Overall Q_P	30.562

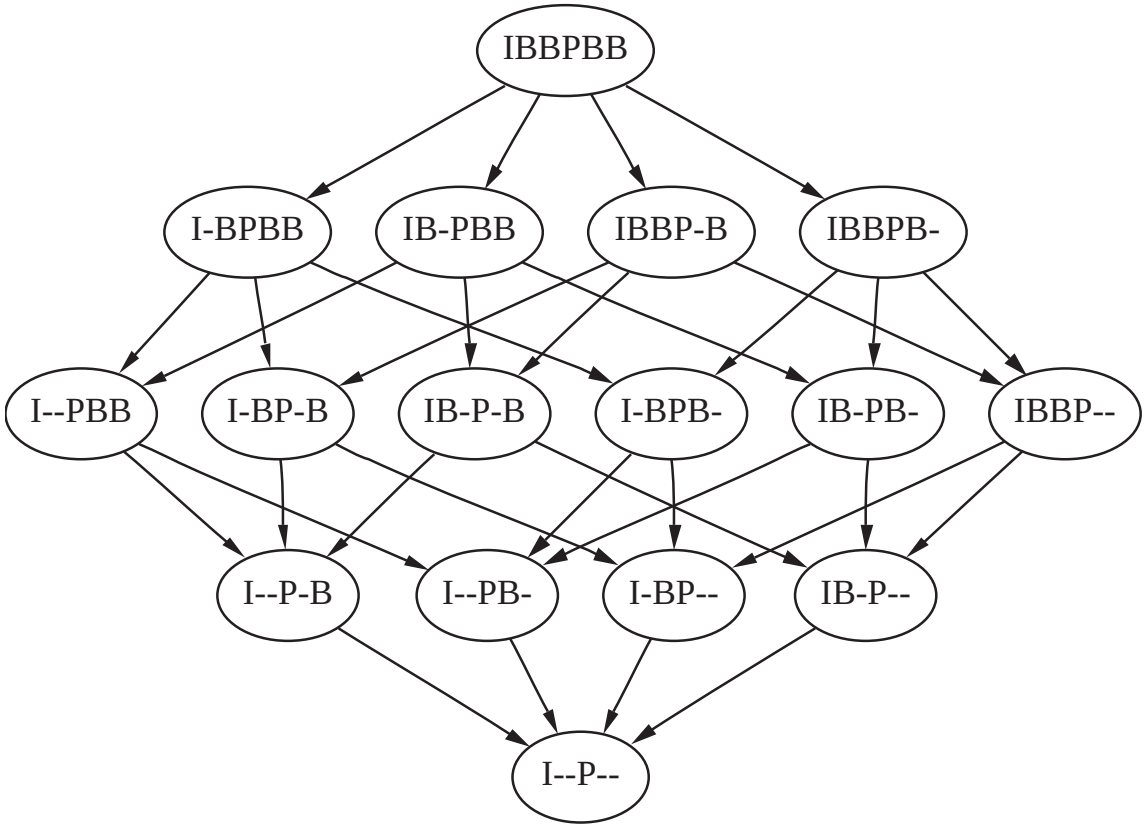
Quality Estimation for GOPs | Arbitrary Frame Sequences

No_F	$Type_F$	No_G	$Type_G$	$psnr(F,G)$
1	I	1	I	35.342
2	B	2	B	33.993
3	B	2	B	23.984
4	B	2	B	19.191
5	B	2	B	14.032
6	P	6	P	35.561
7	B	6	P	14.432
8	B	8	B	34.331
9	B	9	B	34.531
10	B	9	B	14.667
11	B	11	P	34.123
			Overall Q_P	26.7443

Modification Lattice

- A single frame sequence has a lot of different dropping patterns

- Master Pattern is the original frame sequence
- Modifications are frame dropping sequences of the master pattern
- Modifications with the same number of dropped frames are labeled as Layers



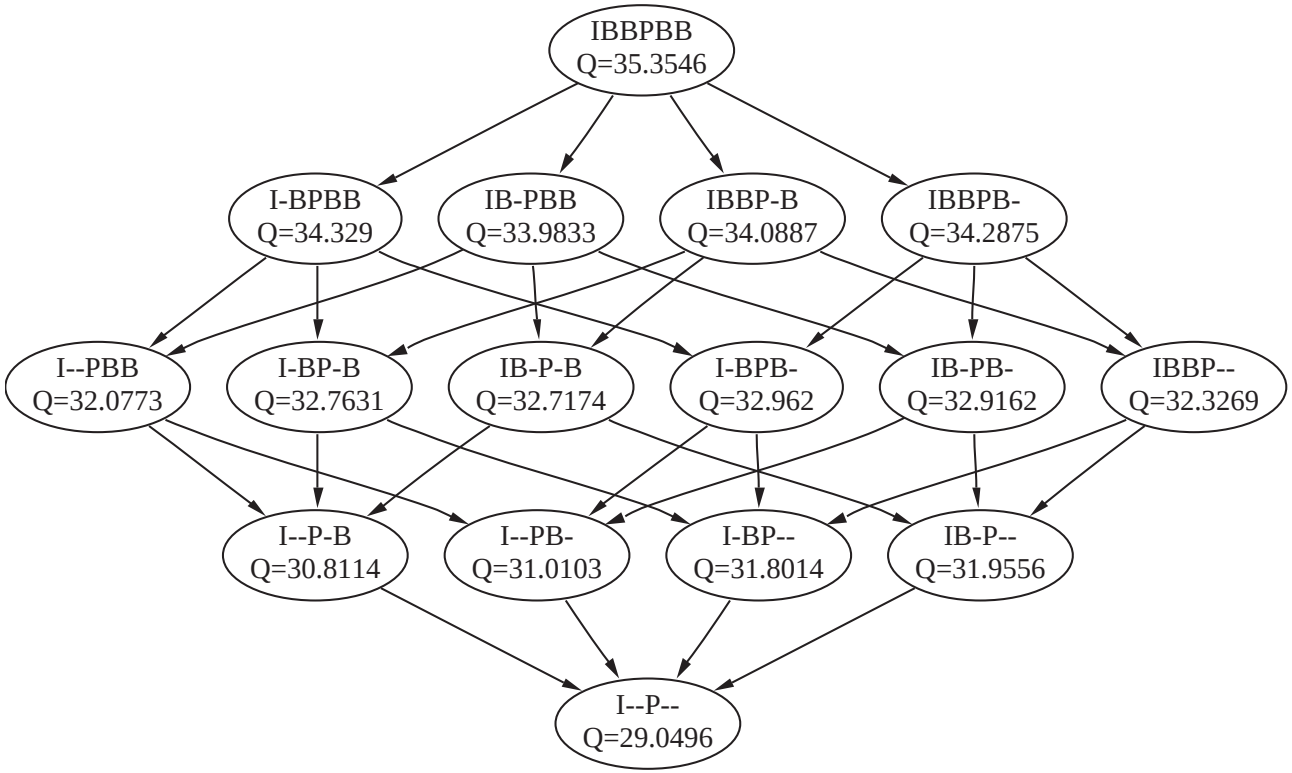
- The Modification with *no* B-Frames is the Base Layer

Modification Lattice

- The formula for arbritrary frame sequences is applied for every modification and its **original** sequence
- Quality measure is assigned to every node in the lattice

Modification Lattice

- The formula for arbitrary frame sequences is applied for every modification and its **original** sequence
- Quality measure is assigned to every node in the lattice



Modification Lattice | BFE - Best First Expansion Heuristic (1/2)

- Building the full lattice needs a lot of computational power and thus, time consuming
- BFE heuristic does not fully expand the lattice
- Only the qualitative best patterns are expanded further

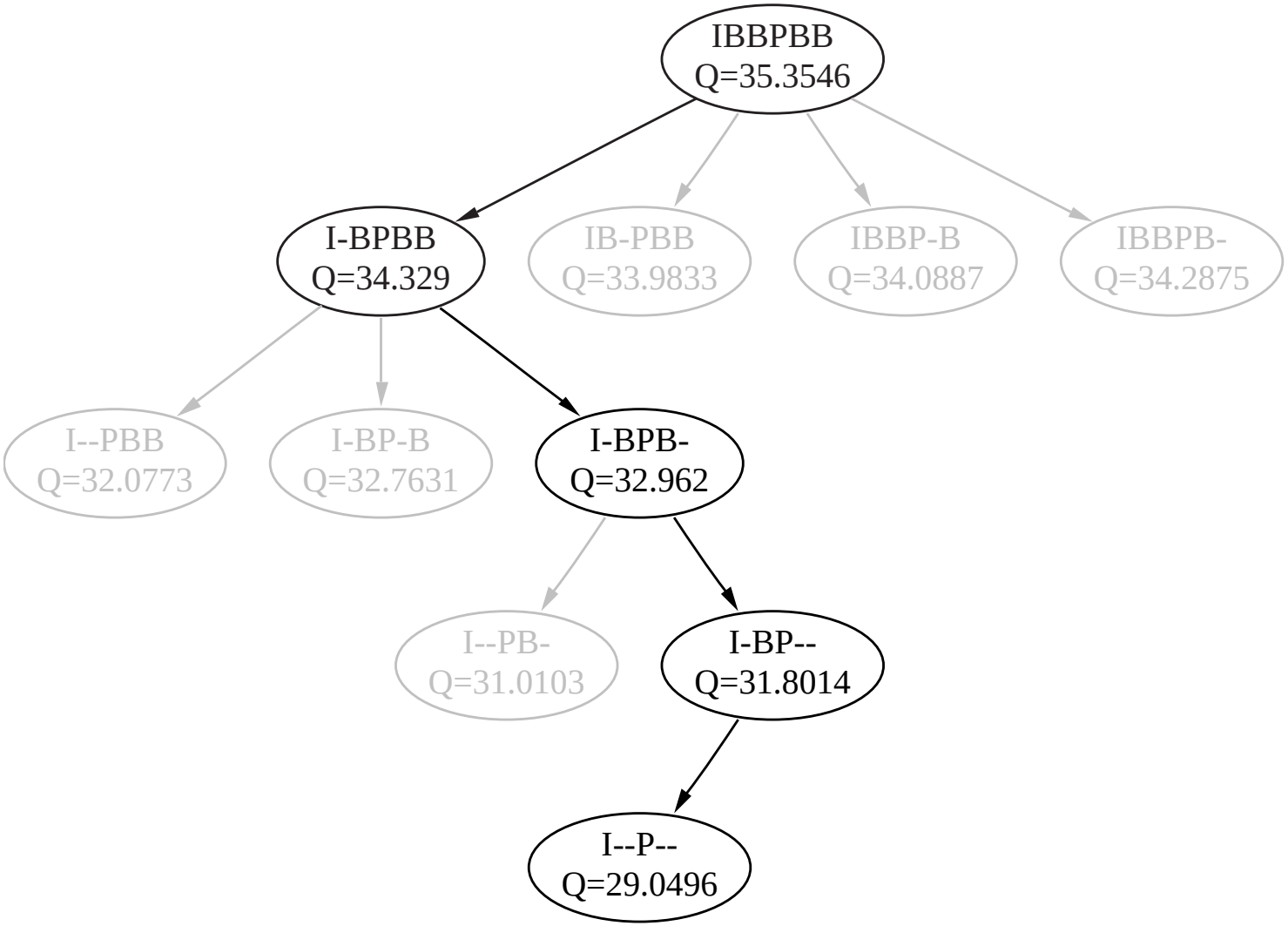
- Building the full lattice needs a lot of computational power and thus, time consuming
- BFE heuristic does not fully expand the lattice
- Only the qualitative best patterns are expanded further

Given a lattice where MP is the master pattern, BL is the base layer, and N is a singleton set of any node in the lattice then the set of BFE nodes is defined as:

$$
\begin{aligned}
B_P &= B'_P(\{MP\}) \\
B'_P(N) &= \begin{cases} N \cup B'_P(max(expand(N))) & N \neq BL, \\ N & N = BL \end{cases}
\end{aligned}
$$

- $expand(N)$ expands a node in the singleton set N and returns a set of all its children
- max takes a set of patterns as input and returns the *singleton set of patterns* with the maximum quality

Modification Lattice | BFE - Best First Expansion Heuristic (2/2)



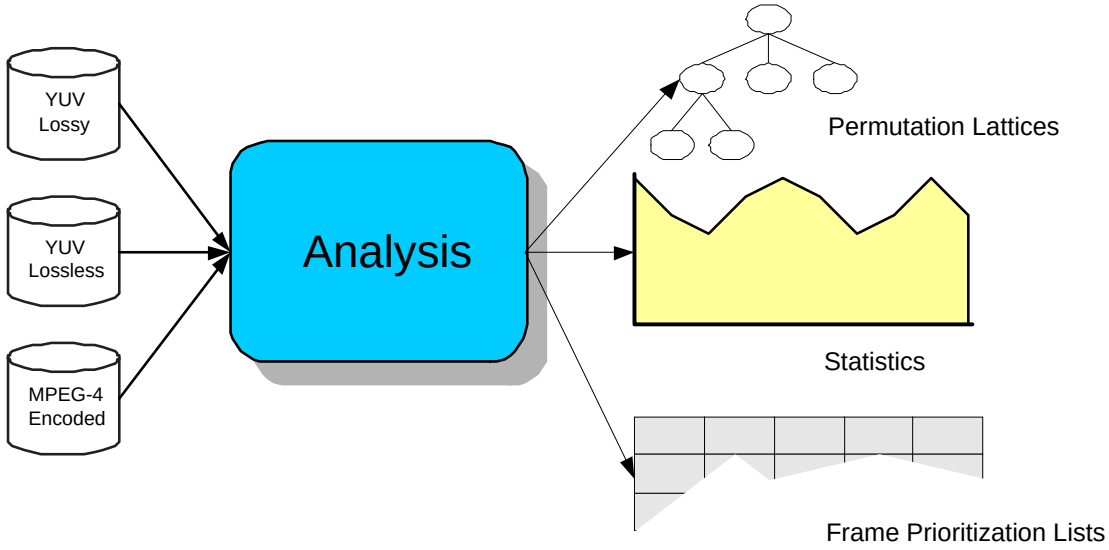
Frame Prioritization

- Frame prioritization only possible with total order of modifications $\Longrightarrow$ Path
- I-Frame priority always 1
- P-Frame priority always 2
- B-Frame priority depends on dropping behavior

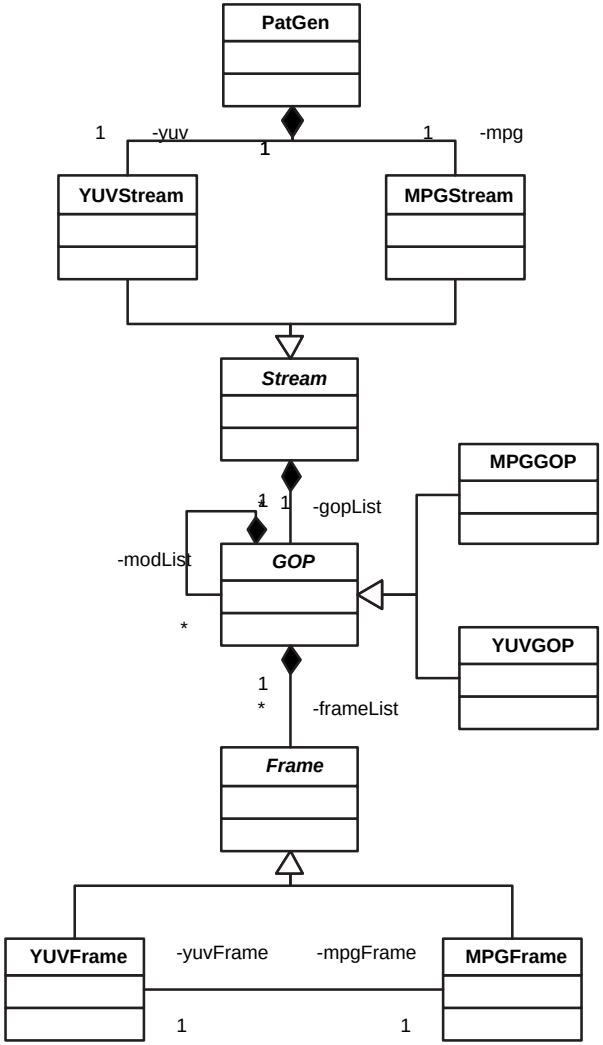
No	Type	Prio	PSNR	Size	FrOffset
0	I-VOP	1	29.0496	7908	0
1	P-VOP	2	29.0496	2677	7908
2	B-VOP	6	35.3546	1579	10585
3	B-VOP	3	31.1014	1540	12164
4	P-VOP	2	29.0106	2785	13704
5	B-VOP	4	32.962	1538	16489
6	B-VOP	5	34.329	1485	18027
7	P-VOP	2	29.0106	2810	19512
			. . .		

The QCTVA Mechanism

- Offline analysis of MPEG-4 video elementary streams
- Find the qualitative best temporal scalability
- Quality measured value is Peak Signal to Noise Ratio - PSNR
- C++ API : Easy to integrate in a wide range of applications
- Output:
  - Modification lattices
  - Statistical information
  - Frame prioritization
  - Streaming simulation
  - Playback
  - ...



The QCTVA API (1/2)



The QCTVA API (2/2)