

Movie Format in a nutshell

2009.5.29 @ESM Mtg. Room
あまのりょー

はじめにお礼

(2009.03.28)

■第24回 XPユーザ会を開催しました

3/27(金)になりますが、開催しました。まず初めに、参加された皆さんに感謝。



[http://mugiwara.jp/ki2/wifky.pl?p=\(2009.03.28\)#p1](http://mugiwara.jp/ki2/wifky.pl?p=(2009.03.28)#p1) より

- あと、角谷さんみたいに動画に慣れ親しんでそうな方でも意外と動画について知らない事もまだありそうなことがわかった
 - やっぱり一度 esmに行って、動画フォーマットについて基礎的なことを話に行かねば・・・
 - 一度そういう勉強会やらへん? って言ったら、話に乗る人居る?

あまびよん 動画勉強会！ ぜひ！ (2009/03/28 21:55:10)

かくたに コーデックちょう勉強したいwww (2009/03/29 00:15:35)

＊角谷さん、天野さん、木下さんをはじめ、
押し掛け勉強会にご協力いただいた皆さん4q!

Agenda

- * レイヤーの区別について

- * Video ES のキソ

レイヤーを意識しよう

ネットワークの基礎

- * ネットワークの技術を学ぶ時に
最初に教わるOSI参照モデル
- * 具体的な境界は議論があるが、とにかく
「いまどこのレイヤーの話をしているか」
を把握することが重要



Windowsアプリを作る時も

＊いま注目しているのはどこ？

- ✓ C++ という言語のことなのか
- ✓ Visual C++ のコンパイラ特性なのか
- ✓ アセンブラレベルのことなのか
- ✓ Win32 API のことなのか
- ✓ MFC というフレームワークのことなのか
- ✓ 自分たちの独自のライブラリのことなのか
- ✓ Visual Studio という IDE のことなのか
- ✓ ドメイン知識の問題なのか

＊ 「いまどこのレイヤーの話をしているか」
を把握することが重要

実は動画もおんなじ

実は動画もおんなじ

*いろいろな人と話をしていると、
そして自分自身をふりかえると、
このところが**ごっちゃ**になっている感じ

実は動画もおんなじ

- *いろいろなヒトと話をしていると、
そして自分自身をふりかえると、
このところが**ごっちゃ**になっている感じ
- *さらに混乱に拍車をかけるのは、
異なるレイヤーに同じ名前(ex. Mpeg) が
ついていること

実は動画もおんなじ

- *いろいろなヒトと話をしていると、
そして自分自身をふりかえると、
このところが**ごっちゃ**になっている感じ
- *さらに混乱に拍車をかけるのは、
異なるレイヤーに同じ名前(ex. Mpeg) が
ついていること
- *なので、この区別がつけば、
かなり**スッキリ**する筈です

動画技術のレイヤー

論理フォーマット等

運用規則

System

ES (Elementary Stream)

補足:コンテンツプロテクション

- *前ページのレイヤー（やそのさらに上下）とは直交して存在
- *DRM技術と呼ばれることもある
- *耐タンパ技術・暗号化などの情報セキュリティ技術、ウォーターマークetc... といった様々な周辺技術
- *本セミナーでは詳しくは取り扱いません

ES (Elementary Stream)

論理フォーマット等

運用規則

System

**ES (Elementary
Stream)**

ES (Elementary Stream)

*単体の Video や Audio など
からなる基本のストリーム



ES (Elementary Stream)



- * 単体の Video や Audio など
からなる基本のストリーム
- * 様々な圧縮技術が
用いられることが多い

ES (Elementary Stream)



- * 単体の Video や Audio などからなる基本のストリーム
- * 様々な圧縮技術が用いられることが多い
- * 「コーデック」と呼ばれているものは通常このレイヤー

ES (Elementary Stream)



- * 単体の Video や Audio などからなる基本のストリーム
- * 様々な圧縮技術が用いられることが多い
- * 「コーデック」と呼ばれているものは通常このレイヤー
- * Video/Audio 以外もあるがここではまずは割愛

いろいろなESの例

* Video

- ✓ Mpeg2 Video
- ✓ Mpeg4 Visual
- ✓ H.264/Mpeg4 AVC
- ✓ DivX
- ✓ Sorenson
- ✓ Windows Media Video (wmv)

* Audio

- ✓ Mpeg1 Audio Layer3 (mp3)
- ✓ AAC
- ✓ Dolby AC3
- ✓ LPCM
- ✓ μ -Law

* Video/Audio 以外のESもあります

System

論理フォーマット等

運用規則

System

ES (Elementary
Stream)

System

＊ 個々のESを多重化(Mux)する方法



System



- * 個々のESを多重化(Mux)する方法
- * 「複数のESを1つのファイルに束ねる方法」とほぼ同じことだと考えれば良い

System



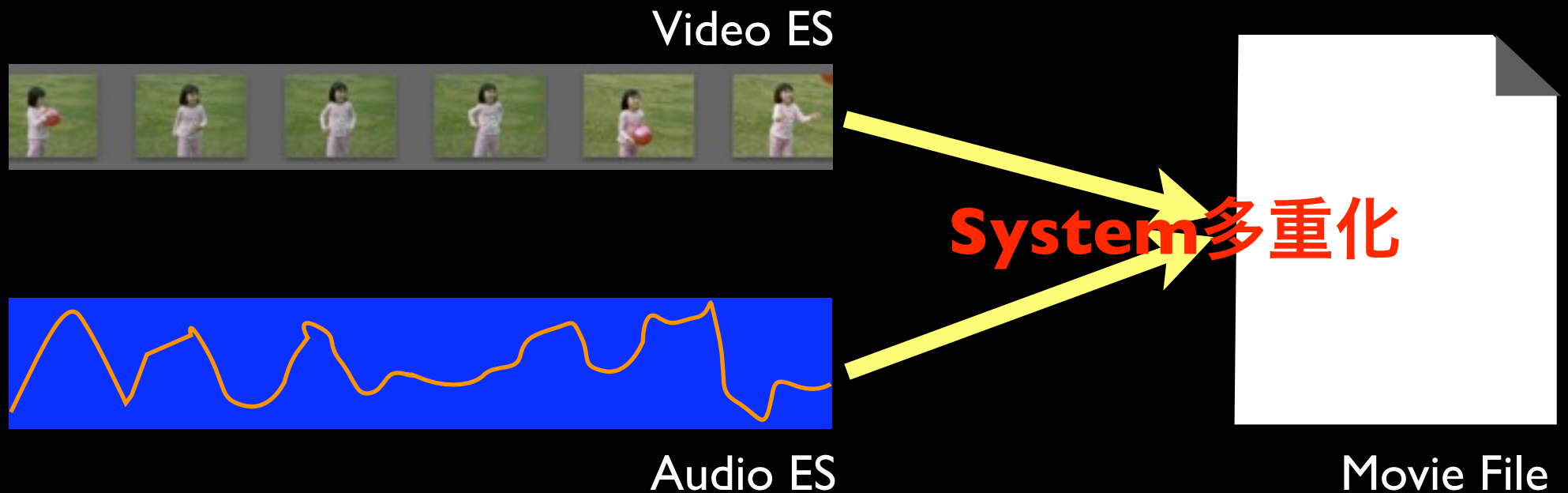
- * 個々のESを多重化(Mux)する方法
- * 「複数のESを1つのファイルに束ねる方法」とほぼ同じことだと考えれば良い
- * 単純にESファイルを連結するだけではうまく行きません

System



- * 個々のESを多重化(Mux)する方法
- * 「複数のESを1つのファイルに束ねる方法」とほぼ同じことだと考えれば良い
- * 単純にESファイルを連結するだけではうまく行きません
- * ES同士の同期(Lip Sync)やプログラムの識別に必要な情報などを記録する

概念図



System のことを「コンテナフォーマット」と言う場合もある

いろいろなSystemの例

- *Mpeg2-PS (.mpg, .mpeg, m2p ...)
- *Mpeg2-TS (.m2t, .m2ts, .mpeg ...)
- *Mpeg4 (.mp4)
- *QuickTime (.mov)
- *Audio Video Interleave (.avi)
- *Advanced System Format
(.asf, .wmv ...)
- *Real Media (.rm)

運用規則

論理フォーマット等

運用規則

System

ES (Elementary
Stream)

運用規則

＊ESやSystemの規格全範囲
を使うのではなく、
使用用途にあわせて
利用できる範囲を定める
のが普通（サブセット）



運用規則



- * ESやSystemの規格全範囲を使うのではなく、使用用途にあわせて**利用できる範囲**を定めるのが普通（サブセット）
- * 例えば画サイズやビットレート、圧縮に使えるツール、プログラム数など

例えばワンセグ放送

圧縮技術 [編集]

映像圧縮技術には、MPEG-4の2倍、MPEG-2に対しては4倍以上という圧縮品質を実現したH.264が採用された。さらに音声にはAACが採用されている。なお、低ビットレートにおいて音質を改善する追加技術SBRの適用に関しては放送局による。

- 動画規格: H.264/MPEG-4 AVC Baseline Profile 1.2
- 解像度: 320×240 (最大)
- 動画ビットレート: 128kbps (16KB/s) (例)
- フレームレート (コマ数) : 15fps (1秒あたり15枚)
- 音声規格: MPEG-2 AAC (SBR技術の適用は放送局による)
- 音声仕様: モノラル、ステレオ、デュアルモノ
- 音声ビットレート: 64kbps (8KB/s) (例)
- データ放送記述規格: BML (Broadcast Markup Language) Cプロファイル
- データ放送ビットレート: 約60kbps (約7.5KB/s) (例)

ESのはなし

Systemのはなし

ストリーム [編集]

通常の地上デジタル放送と同様MPEG-2システムに準拠したストリームとして伝送されるが、帯域削減の為あるストリームに含まれる多重化された番組をPMTのバケット番号で列挙するテーブルであるPATの送出を行わず、ある番組に含まれるストリームがどのバケット番号を使っているかを指定するPMTのバケット番号は固定の値0x1fc8(もし多重化された番組があるなら副番組は0x1fc9,帯域が許せば0x1fcfまで8番組を識別可能)を使用する。

wikipediaより

ニコニコ動画の推奨動画フォーマット(1)

推奨動画フォーマット

元のビデオソースは

映像(DivX/XviD/MPEG1/MPEG2)・音声共にCBRでエンコード
が推奨。

画像サイズは、

4:3なら512x384、16:9なら512x288（ニコニコ動画プレイヤーと同じサイズです）

が推奨。

WMVの場合は

WindowsMedia8系で、映像・音声共にCBRでエンコードされたものが推奨。

FLVの場合は、

正常なファイルヘッダーがあり、NULLフレームを含まないものが推奨。

ニコニコ動画の推奨動画フォーマット(2)

AudioES

VideoES

System

H.264+AAC(H.264/AVCフォーマットMP4コンテナでのアップロードが可能。

正式なコンテナは MPEG-4 Part 10 AVC 。

H.264のプロファイルは Baseline ProfileまたはMain Profileが推奨。

固定フレームレートが推奨。

VFR等の可変フレームレートは正しく処理されない場合があります。

音声AACは HE-AAC(44.1kHz/48.0kHz 32kbps~64kbps推奨)

または AAC-LC(44.1kHz/48.0kHz64kbps?96kbps推奨) に対応。

MP4コンテナには映像1トラック、音声1トラックのみ対応。

その他の音声・映像の多重化は対象外です。

VideoESの
運用規則

AudioESの
運用規則

Systemの
運用規則

上記仕様を満たさない場合は通常の動画としてあつかわれ通常のサーバサイドエンコードの対象となります。

論理フォーマット等

論理フォーマット等

運用規則

System

ES (Elementary
Stream)

論理フォーマット等

論理フォーマット等
運用規則
System
ES (Elementary Stream)

＊主に動画ファイル以外の
データ構造やフォルダ配置、
ファイル命名規則
などを規定

論理フォーマット等

論理フォーマット等
運用規則
System
ES (Elementary Stream)

- *主に動画ファイル以外のデータ構造やフォルダ配置、ファイル命名規則などを規定
- *運用規則とも深く関わっている場合が多い

論理フォーマット等

論理フォーマット等
運用規則
System
ES (Elementary Stream)

- *主に動画ファイル以外のデータ構造やフォルダ配置、ファイル命名規則などを規定
- *運用規則とも深く関わっている場合が多い
- *さらに物理フォーマットまで規定されていることもある

論理フォーマットの例

- *DVD (-Video, -VR, +VR)
- *Blu-ray (BDMV, BDAV)
- *AVCHD
- *Memory Stick Video Format (PSPなど)

例えばDVD-Videoの場合

フォルダ	名前	サイズ	種類
デスクトップ	VIDEO_TS.BUP	12 KB	BUP ファイル
マイ ドキュメント	VIDEO_TS.IFO	12 KB	IFO ファイル
マイ コンピュータ	VIDEO_TS.VOB	24,108 KB	VOB ファイル
3.5 インチ FD (A:)	VTS_01_0.BUP	114 KB	BUP ファイル
ローカル ディスク (C:)	VTS_01_0.IFO	114 KB	IFO ファイル
AVBD_34056 (D:)	VTS_01_0.VOB	73,234 KB	VOB ファイル
AUDIO_TS	VTS_01_1.VOB	1,048,120 KB	VOB ファイル
VIDEO_TS	VTS_01_2.VOB	1,048,426 KB	VOB ファイル
	VTS_01_3.VOB	1,048,342 KB	VOB ファイル
	VTS_01_4.VOB	1,048,134 KB	VOB ファイル
	VTS_01_5.VOB	1,048,084 KB	VOB ファイル
	VTS_01_6.VOB	1,048,266 KB	VOB ファイル
	VTS_01_7.VOB	1,048,370 KB	VOB ファイル
	VTS_01_8.VOB	423,268 KB	VOB ファイル

VOBファイルがMpeg2-PSファイルであり、
その他のファイルにメタ情報が収められている。
また、フォルダ階層・ファイル名規則やファイルサイズ
上限なども定められている。

ここまでのまとめ

ここまでのまとめ

*動画ファイルは

ここまでのまとめ

*動画ファイルは

*Video/Audio などの個々のESを

ここまでのまとめ

*動画ファイルは

*Video/Audio などの個々のESを

*なんらかのSystemでくるんだもの。

ここまでのまとめ

- *動画ファイルは
- *Video/Audio などの個々のESを
- *なんらかのSystemでくるんだもの。
- *ESやSystemの内容は

ここまでのまとめ

- *動画ファイルは
- *Video/Audio などの個々のESを
- *なんらかのSystemでくるんだもの。
- *ESやSystemの内容は
- *運用規則で実際のシンタックスが決まる。

ここまでのまとめ

- * 動画ファイルは
- * Video/Audio などの個々のESを
- * なんらかのSystemでくるんだもの。
- * ESやSystemの内容は
- * 運用規則で実際のシンタックスが決まる。
- * さらに、論理フォーマットまで含めて規格化されることもある。

補足:視聴するとき

補足:視聴するとき

*動画ファイルの構成を逆に考える

補足:視聴するとき

- * 動画ファイルの構成を逆に考える
- * まずSystemをほどこき(Demux, SysDec)、

補足:視聴するとき

- * 動画ファイルの構成を逆に考える
- * まずSystemをほどこき(Demux, SysDec)、
- * 個々のESをデコードし、

補足:視聴するとき

- * 動画ファイルの構成を逆に考える
- * まずSystemをほどこき(Demux, SysDec)、
- * 個々のESをデコードし、
- * 必要ならリオーダー (後述) しながら、

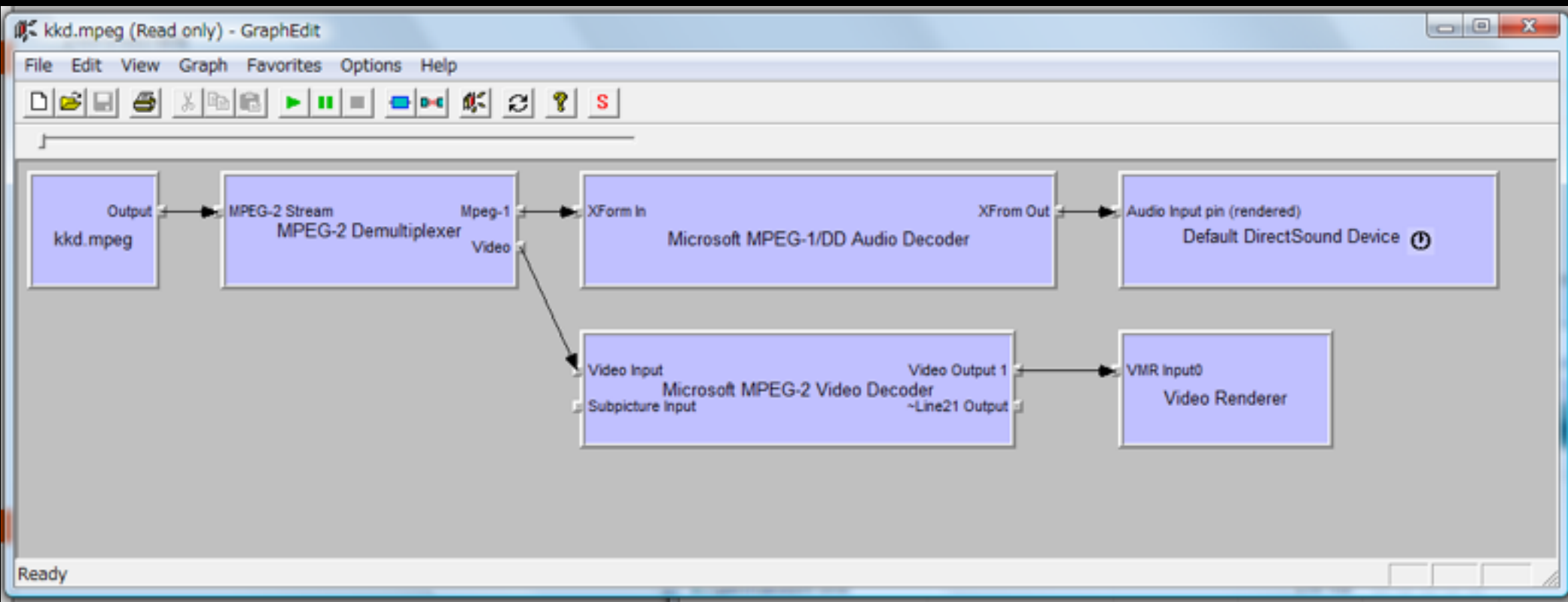
補足:視聴するとき

- * 動画ファイルの構成を逆に考える
- * まずSystemをほどこき(Demux, SysDec)、
- * 個々のESをデコードし、
- * 必要ならリオーダー (後述) しながら、
- * Systemの時間情報を元に同期を取りつつ
各デバイスで再生する

補足:視聴するとき

- * 動画ファイルの構成を逆に考える
- * まずSystemをほどこき(Demux, SysDec)、
- * 個々のESをデコードし、
- * 必要ならリオーダー (後述) しながら、
- * Systemの時間情報を元に同期を取りつつ
各デバイスで再生する
- * 上記の処理スピードによってカクツキや
描画品質の低下につながることもある

WindowsのDirectShow



FilterGraph と呼ばれる一連のデータ流処理グラフを構築する。

「ファイルから読み込む」「Systemをほどく」「Audio/Videoをそれぞれデコードする」「Audio/Videoをそれぞれのデバイスで再生する」フィルタが組み合わされていることがわかる。

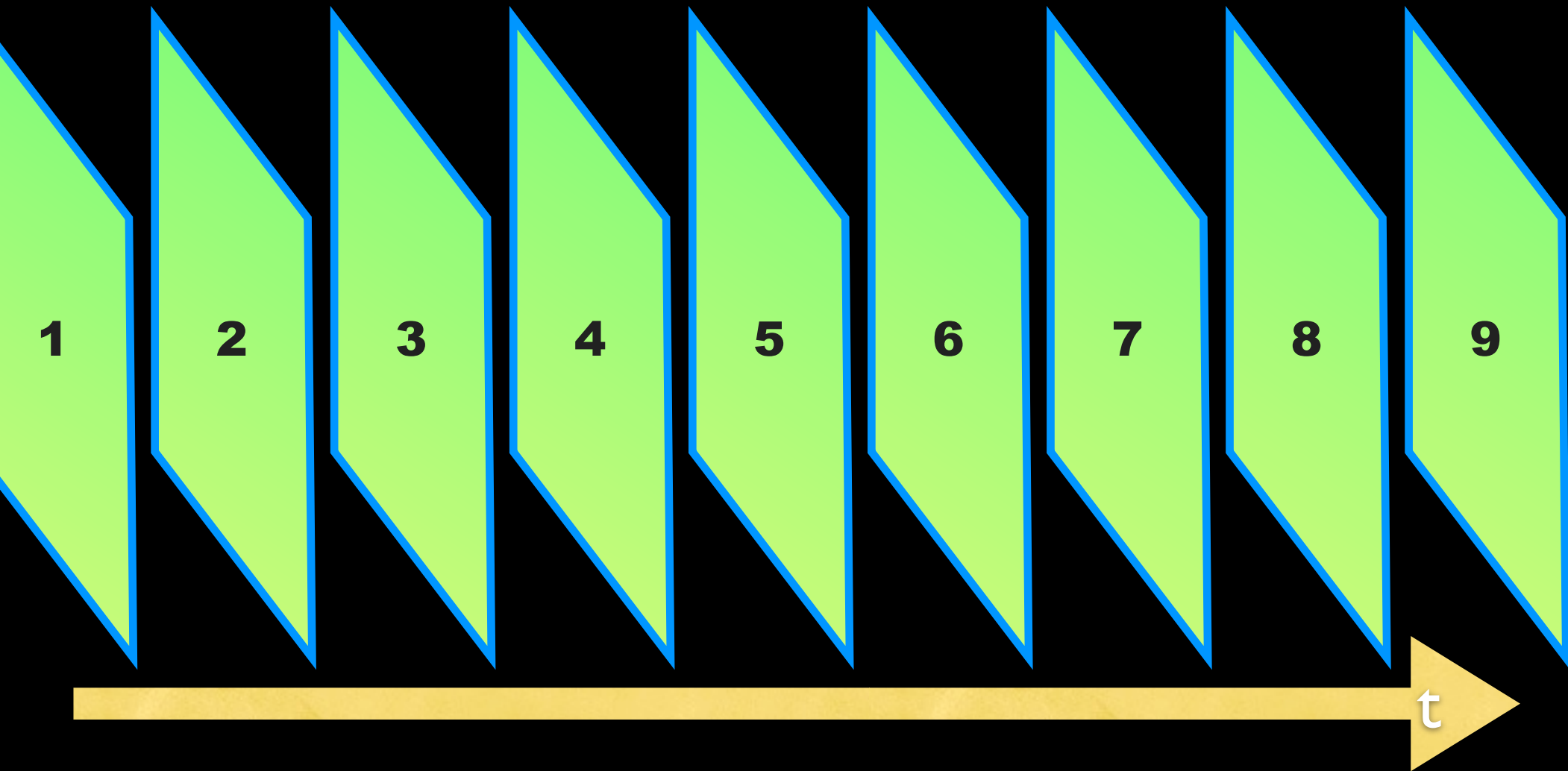
Audio/Videoの同期はFilterGraphが調整を行う。

**Mpeg Systemを
覗いてみよう**

本日は割愛します
代わりに・・・

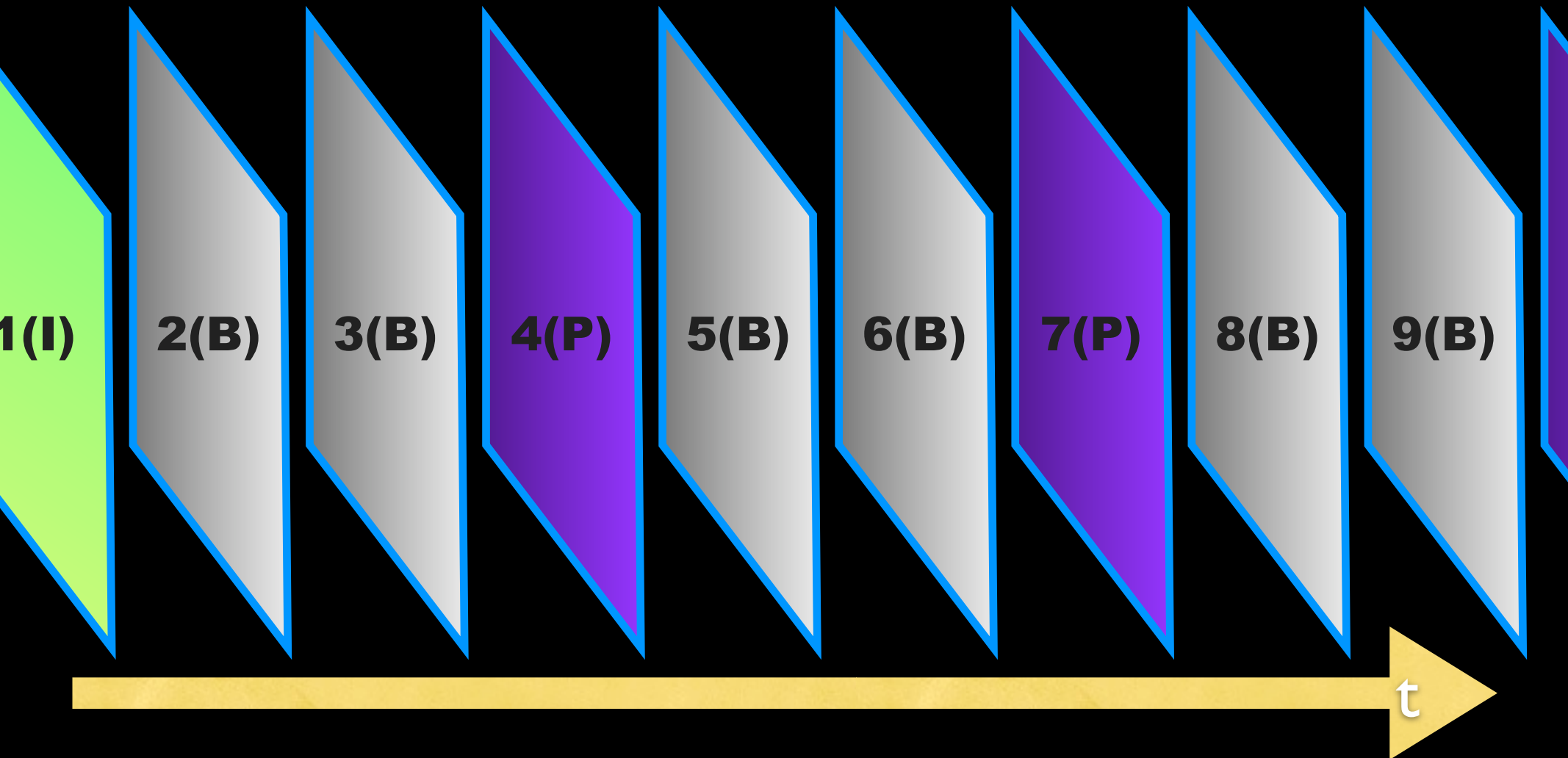
Mpeg Video ESの 構造を覗いてみよう

比較の為にMotion Jpeg



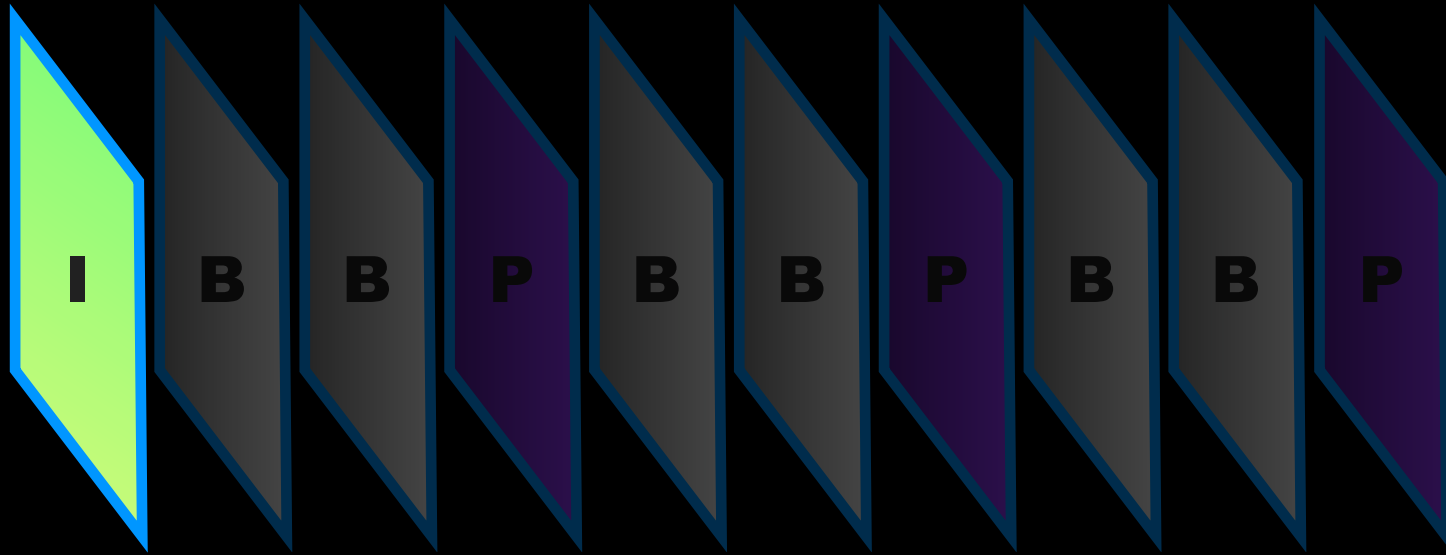
Jpegを順番に並べていて、各フレームは独立にデコードできる

そしてMpeg2 Video



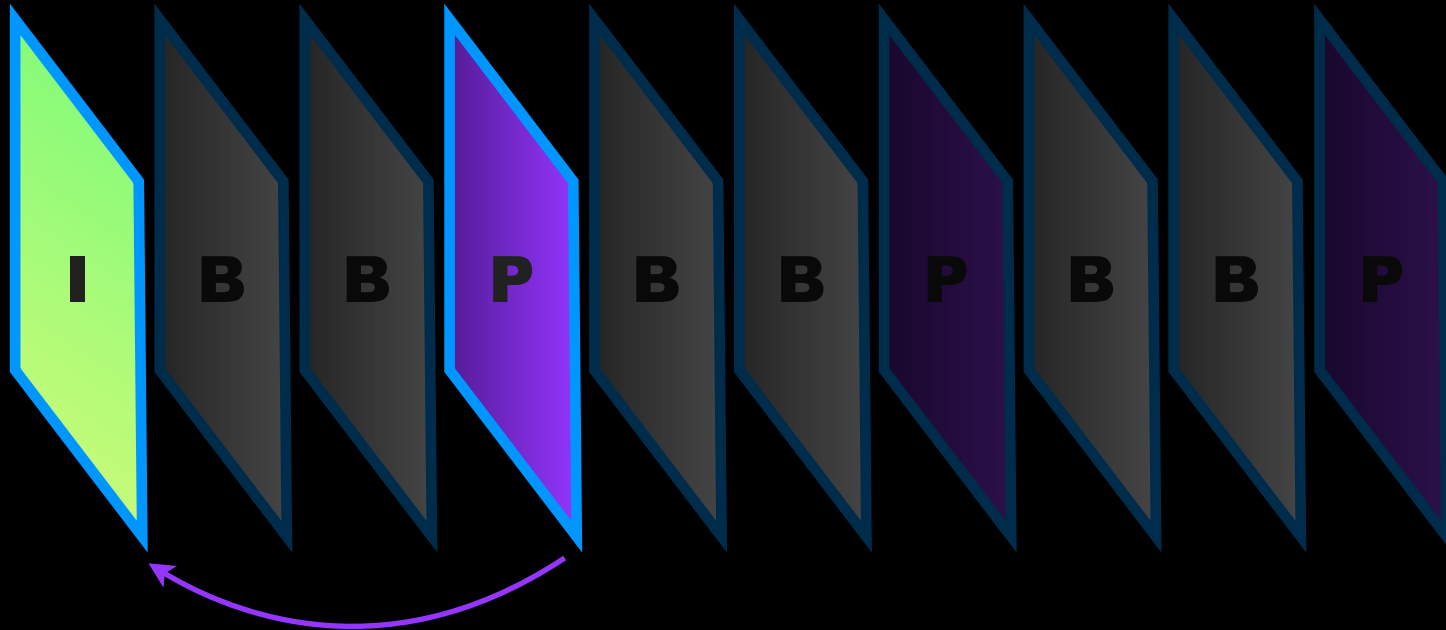
IPBの3タイプのピクチャ。Iピクチャだけは独立にデコードできる

I (Intra Picture)



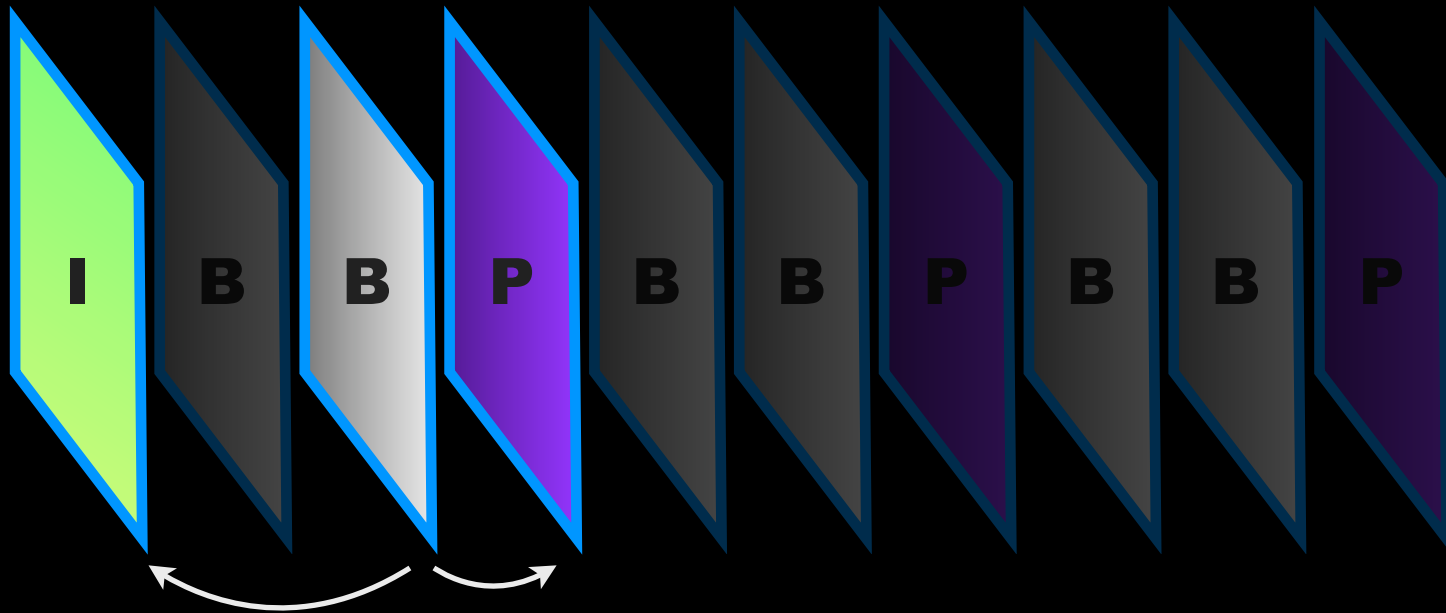
*基本的には Jpeg と同様の方式で
圧縮された一枚の画

P (Predictive Picture)



- *前方の画からの差分圧縮がされている
- *データサイズは少なくなる
- *画質は悪くなる

B (Bidirectionally Predictive Picture)

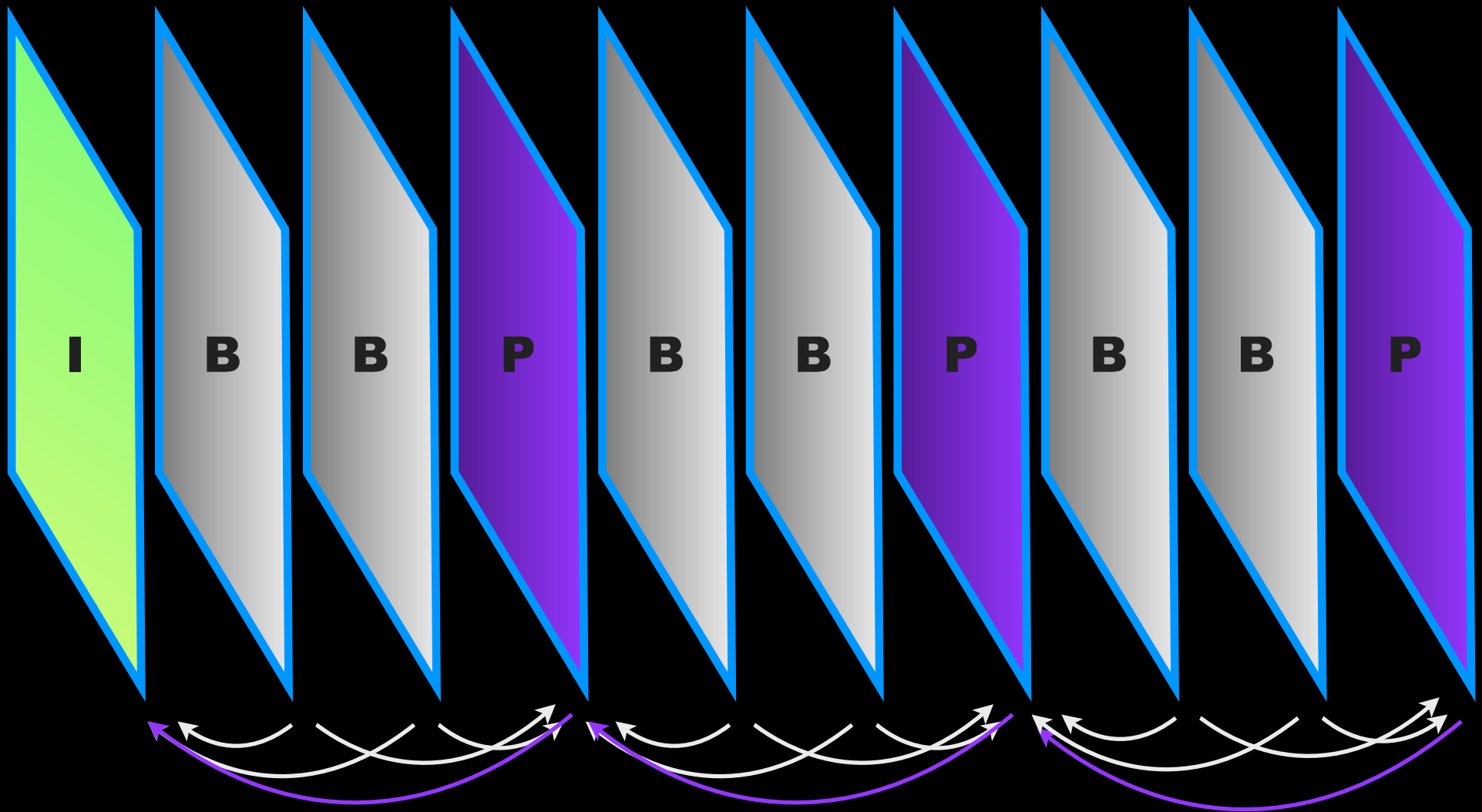


- * 前方および後方の画からの差分圧縮がされている
- * データサイズはより少なくなる
- * 画質はより悪くなる

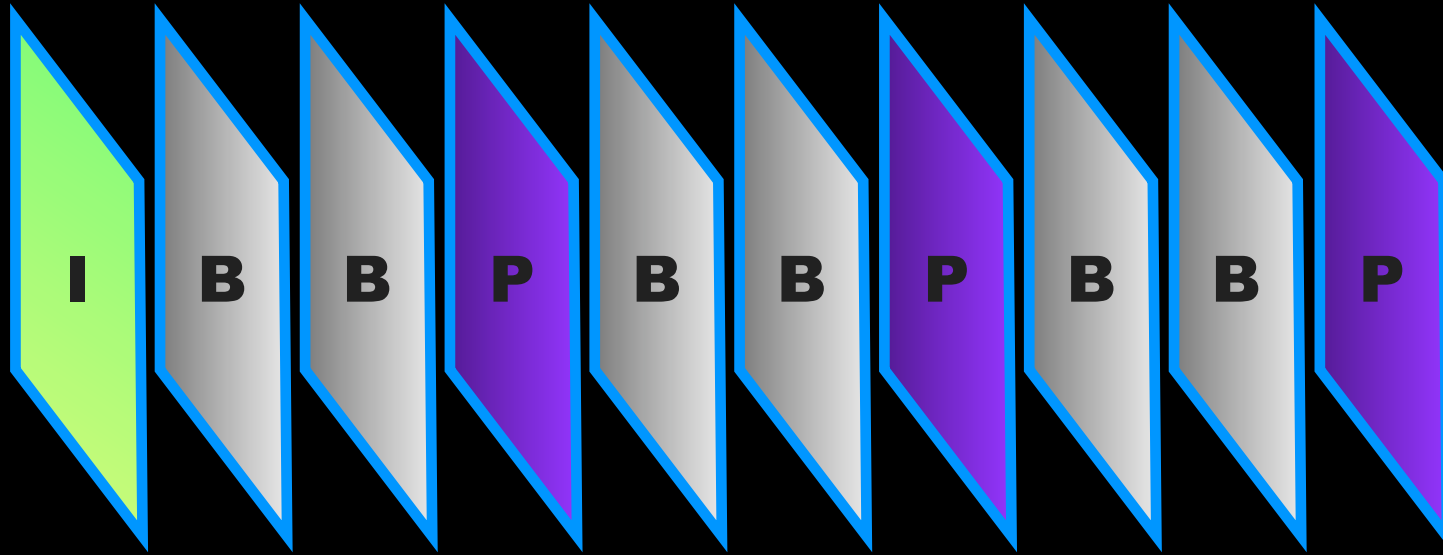
Picture Type の違い

- *IPBにはデータサイズと画質の間のトレードオフがある
- *同じビットレートでも IPBのどこにどれくらいのビット数を割り当てるか、でエンコーダの癖がある

まとめるとこんな参照関係

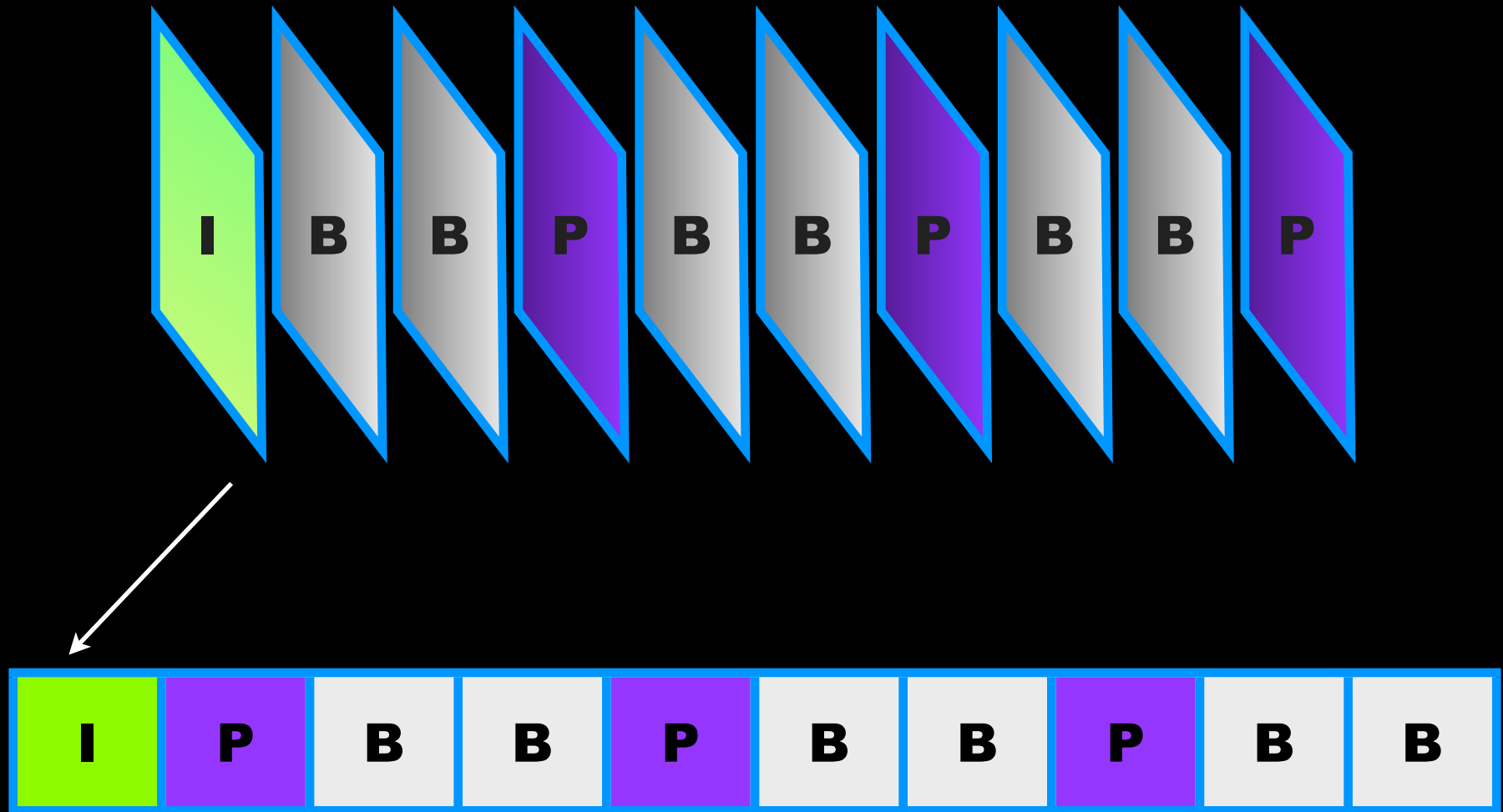


Display Order と Bitstream Order



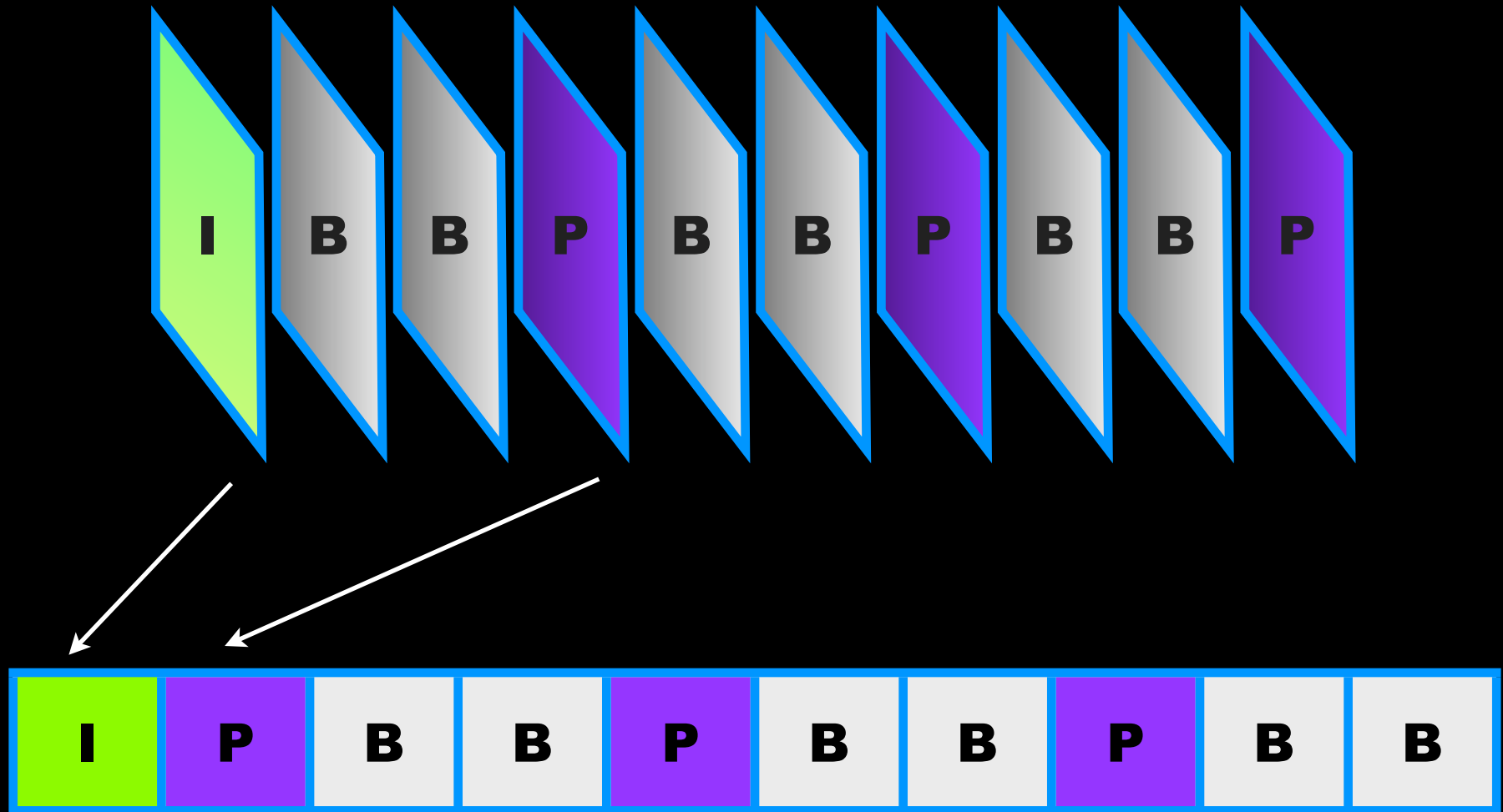
データの並び順(Bitstream Order) としては、
予測の参照に使われる画が先にデコードされる必要がある
つまり、デコード結果をリオーダーして表示する必要がある

Display Order と Bitstream Order



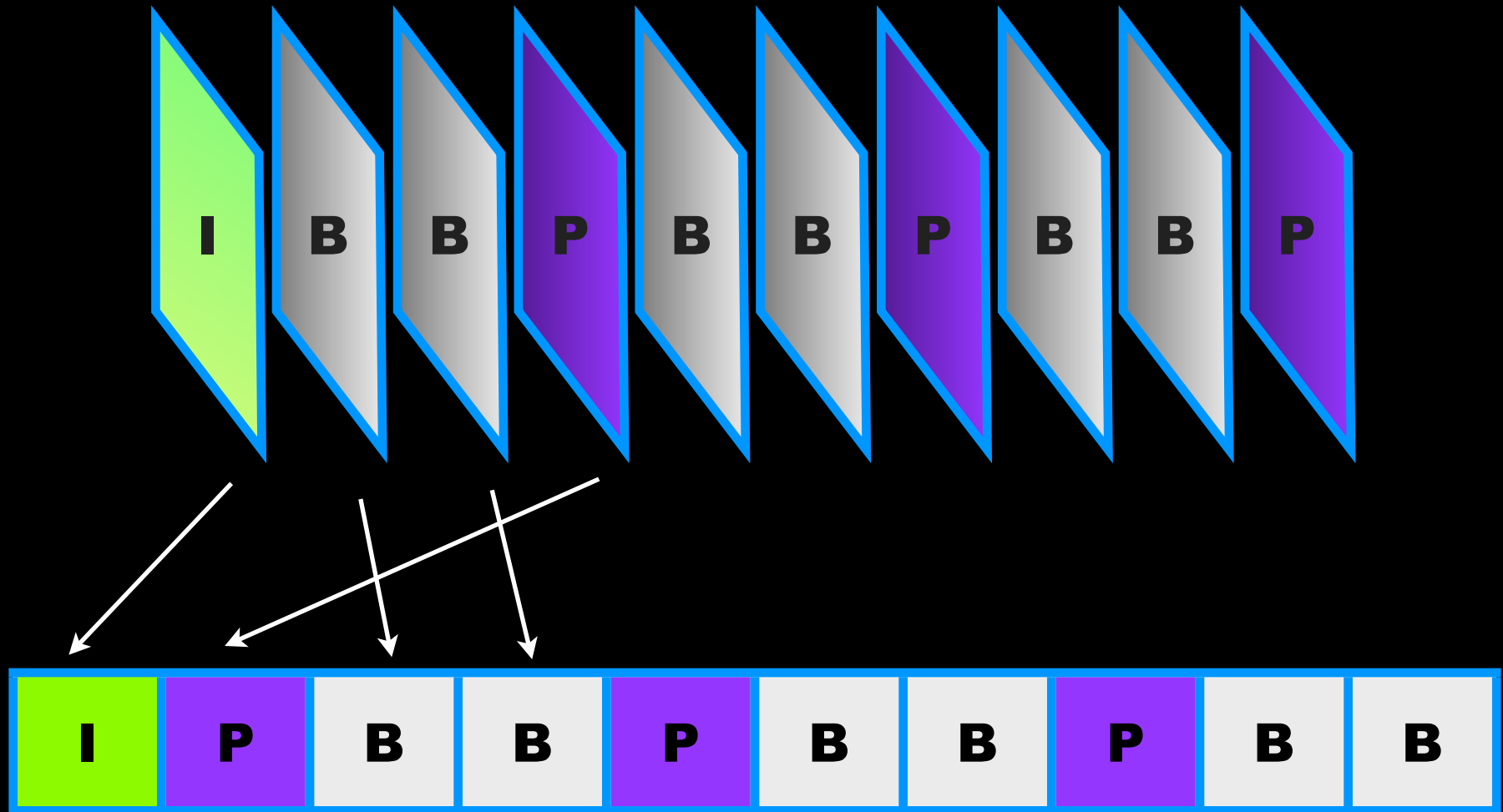
データの並び順(Bitstream Order) としては、
予測の参照に使われる画が先にデコードされる必要がある
つまり、デコード結果をリオーダーして表示する必要がある

Display Order と Bitstream Order



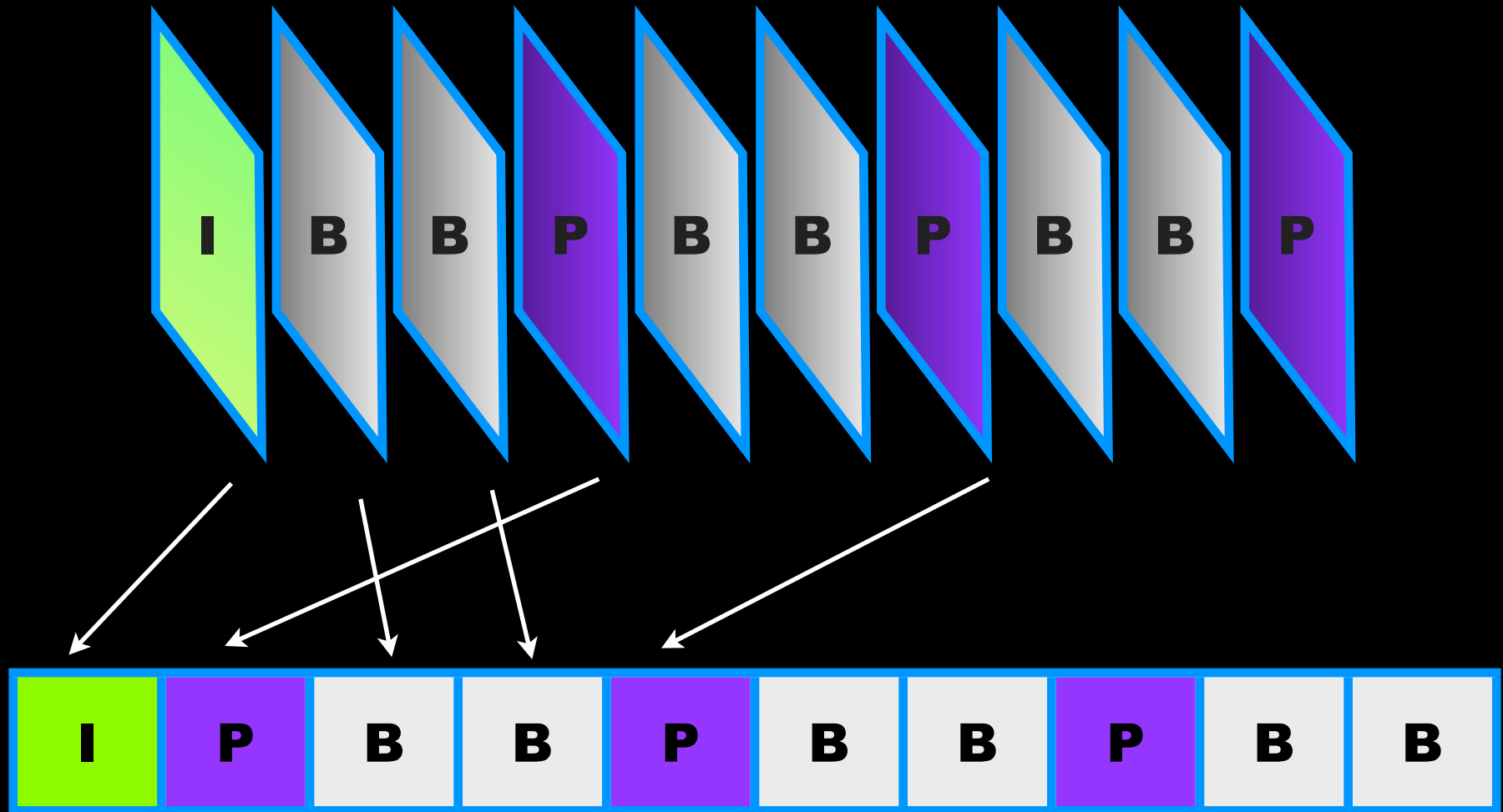
データの並び順(Bitstream Order) としては、
予測の参照に使われる画が先にデコードされる必要がある
つまり、デコード結果をリオーダーして表示する必要がある

Display Order と Bitstream Order



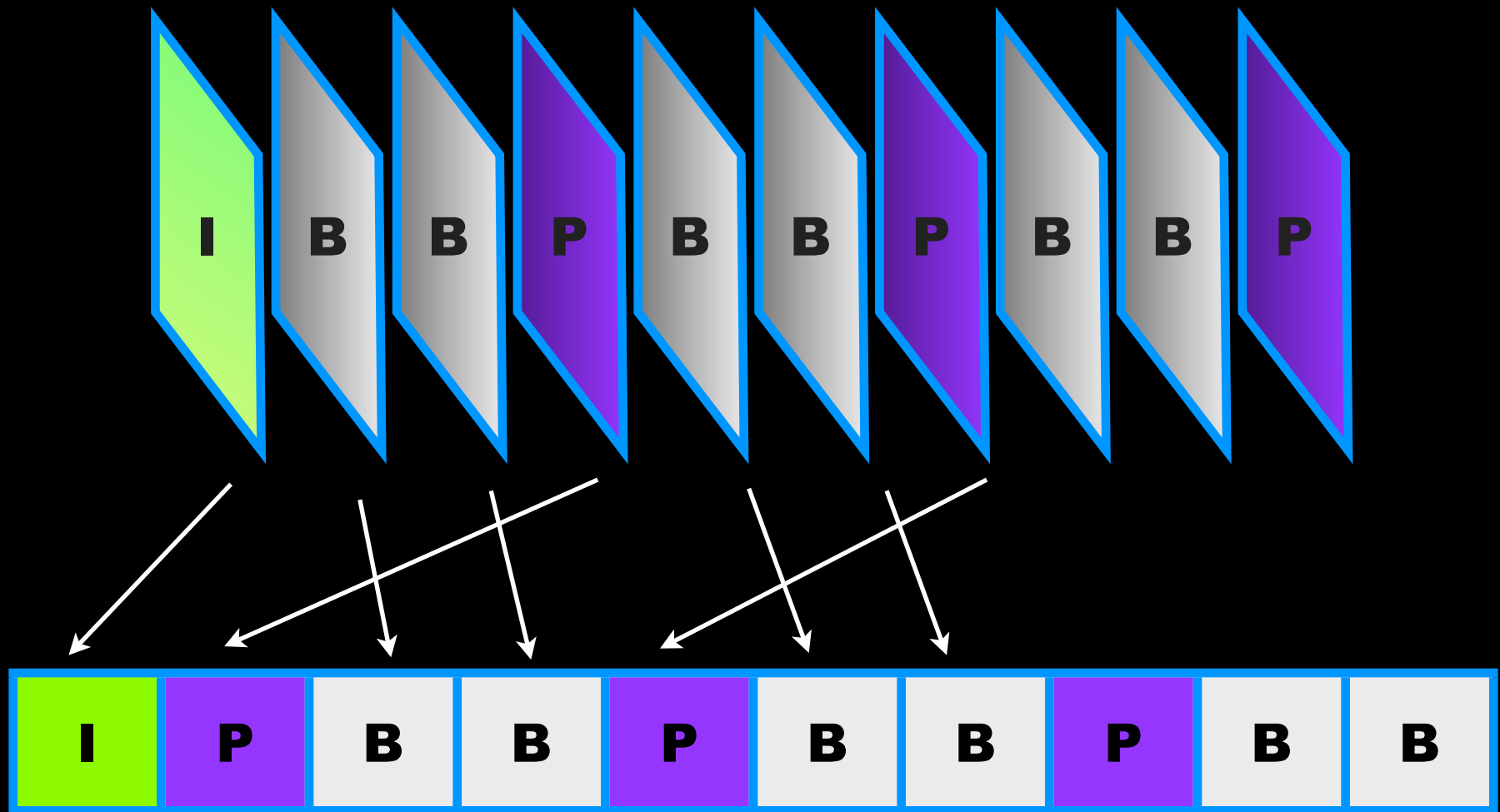
データの並び順(Bitstream Order) としては、
予測の参照に使われる画が先にデコードされる必要がある
つまり、デコード結果をリオーダーして表示する必要がある

Display Order と Bitstream Order



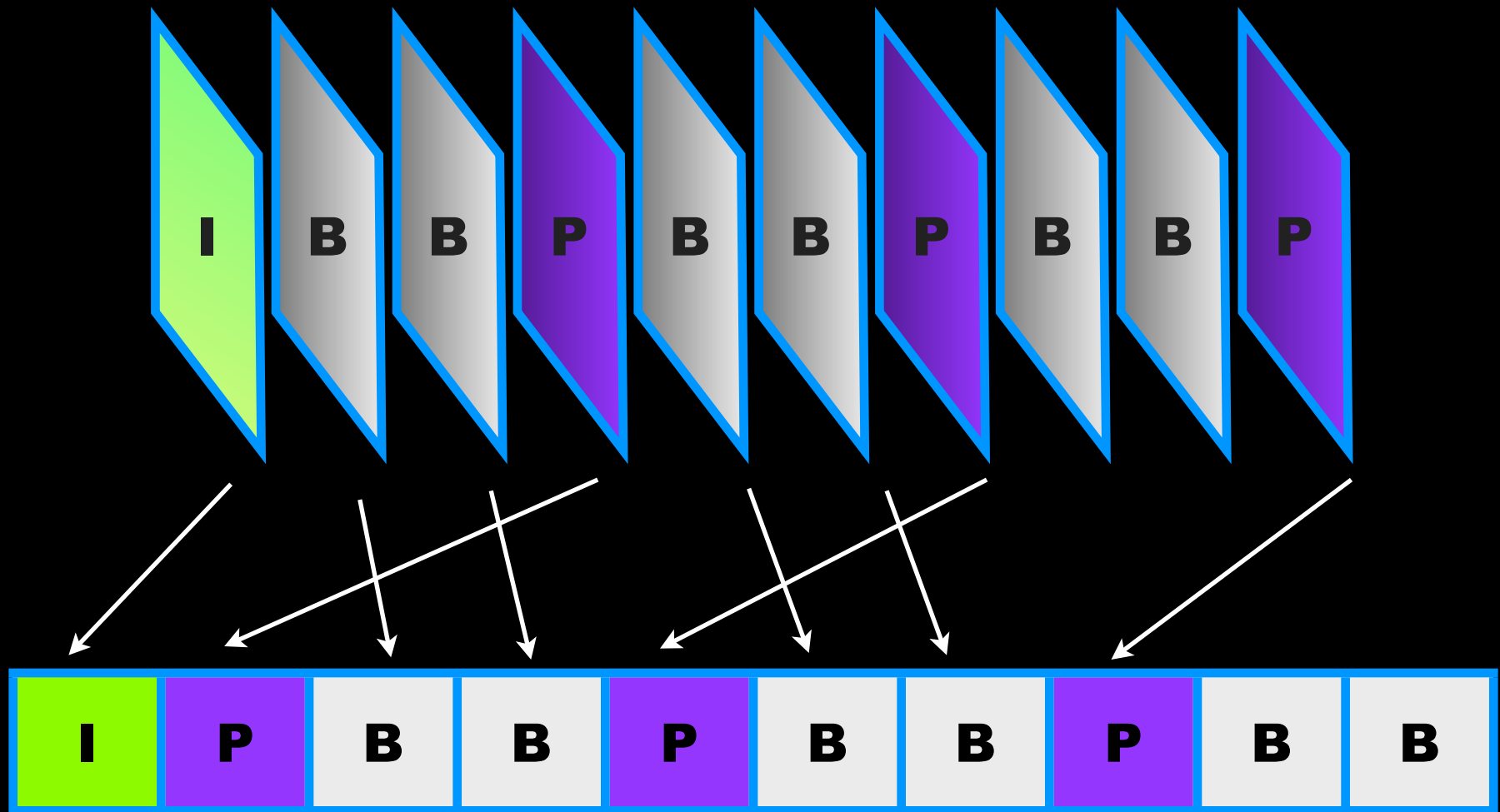
データの並び順(Bitstream Order) としては、
予測の参照に使われる画が先にデコードされる必要がある
つまり、デコード結果をリオーダーして表示する必要がある

Display Order と Bitstream Order



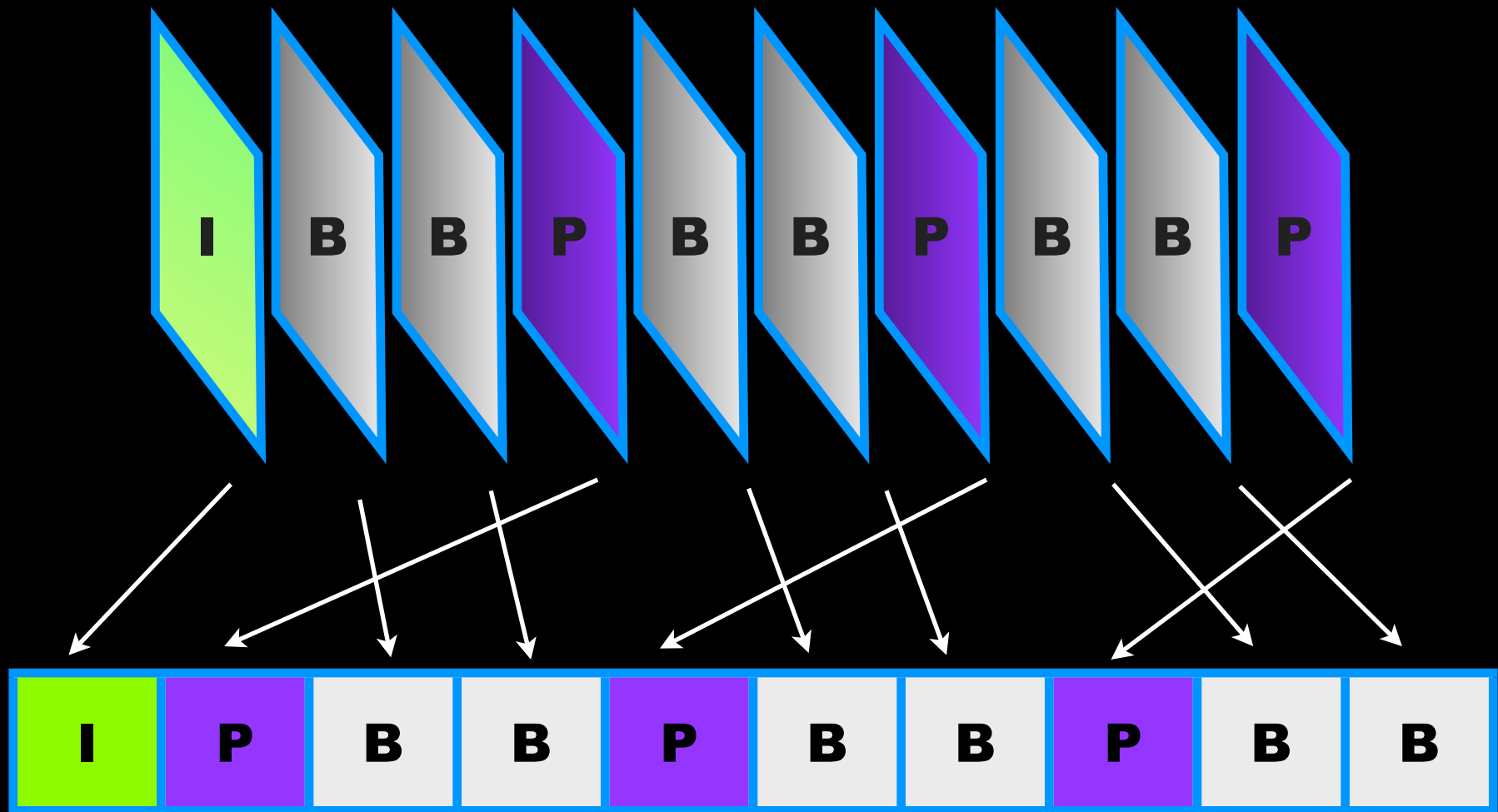
データの並び順(Bitstream Order)としては、
予測の参照に使われる画が先にデコードされる必要がある
つまり、デコード結果をリオーダーして表示する必要がある

Display Order と Bitstream Order



データの並び順(Bitstream Order) としては、
予測の参照に使われる画が先にデコードされる必要がある
つまり、デコード結果をリオーダーして表示する必要がある

Display Order と Bitstream Order



データの並び順(Bitstream Order) としては、
予測の参照に使われる画が先にデコードされる必要がある
つまり、デコード結果をリオーダーして表示する必要がある

バッファ制御

- *Motion Jpeg のようにデータ順のまま表示すればいい場合と違い、Mpeg Videoでは未来の画が先にエンコード・デコードされるため、何も考えないと**バッファが破綻**する
(オーバー or アンダー フロー)
- *そのため、バッファモデルについても規格で定義されている

ここまでの話から
動画技術者なら誰もが呟く
ある真理が導かれます

「アイこそ全ての始まり」

「アイが無ければ
意味が無い」

GOP (Group Of Pictures)

GOP (Group Of Pictures)

- * Mpeg Video ストリームを扱う上で
構成上意味のある**最小単位の画の集合**

GOP (Group Of Pictures)

- * Mpeg Video ストリームを扱う上で
構成上意味のある**最小単位の画の集合**
- * 0.5秒程度(NTSCで15frame)がよくある

GOP (Group Of Pictures)

- * Mpeg Video ストリームを扱う上で
構成上意味のある**最小単位の画の集合**
- * 0.5秒程度(NTSCで15frame)がよくある
- * ものすごく乱暴に言うと、
I Pic から 次のI Picの手前まで

GOP (Group Of Pictures)

- * Mpeg Video ストリームを扱う上で
構成上意味のある**最小単位の画の集合**
- * 0.5秒程度(NTSCで15frame)がよくある
- * ものすごく乱暴に言うと、
I Pic から 次のI Picの手前まで
✓ いや、それはちょっと乱暴杉...

GOP (Group Of Pictures)

- * Mpeg Video ストリームを扱う上で
構成上意味のある**最小単位の画の集合**
- * 0.5秒程度(NTSCで15frame)がよくある
- * ものすごく乱暴に言うと、
I Pic から 次のI Picの手前まで
✓ いや、それはちょっと乱暴杉...
- * 簡易的な編集アプリでは GOP単位の
切り貼りだけがサポートされる

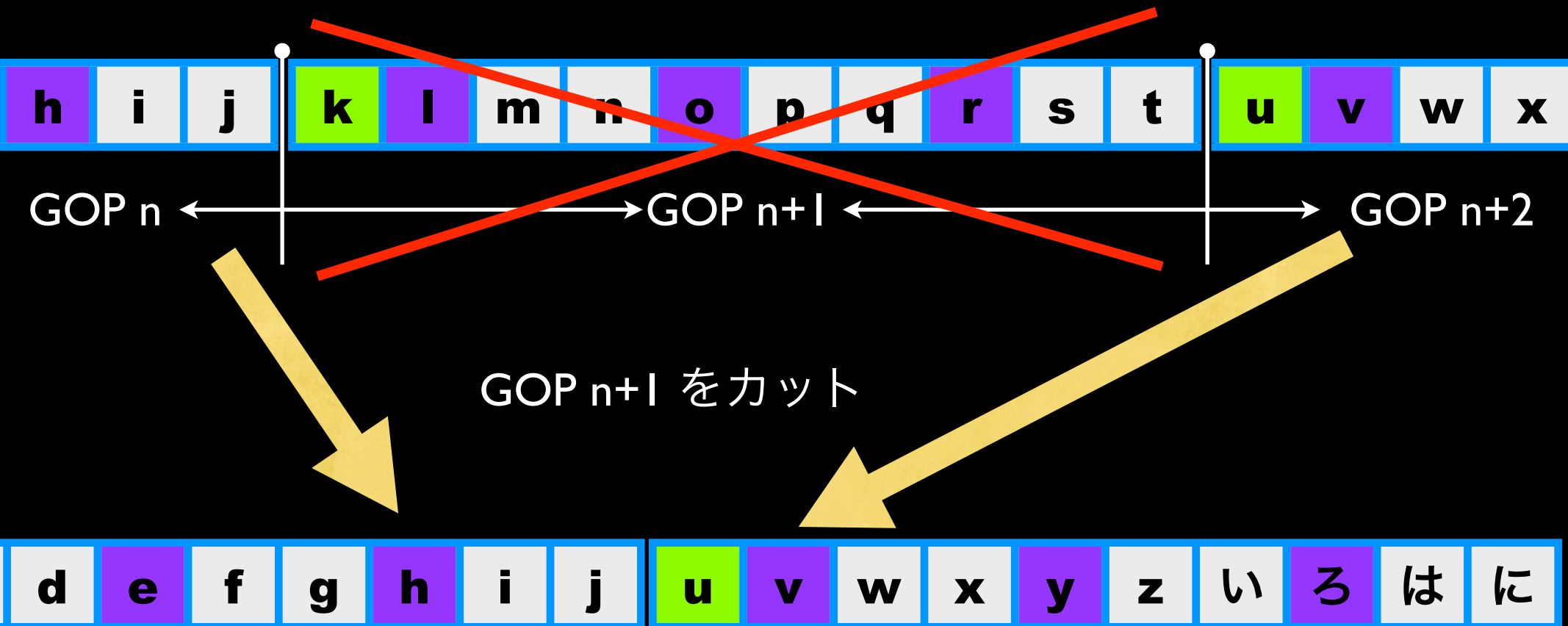
GOP (Group Of Pictures)

- * Mpeg Video ストリームを扱う上で
構成上意味のある**最小単位の画の集合**
- * 0.5秒程度(NTSCで15frame)がよくある
- * ものすごく乱暴に言うと、
I Pic から 次のI Picの手前まで
 - ✓ いや、それはちょっと乱暴杉...
- * 簡易的な編集アプリでは GOP単位の
切り貼りだけがサポートされる
 - ✓ 「**フレーム単位の編集**」には再エンコが必ず発生する

GOP (Group Of Pictures)

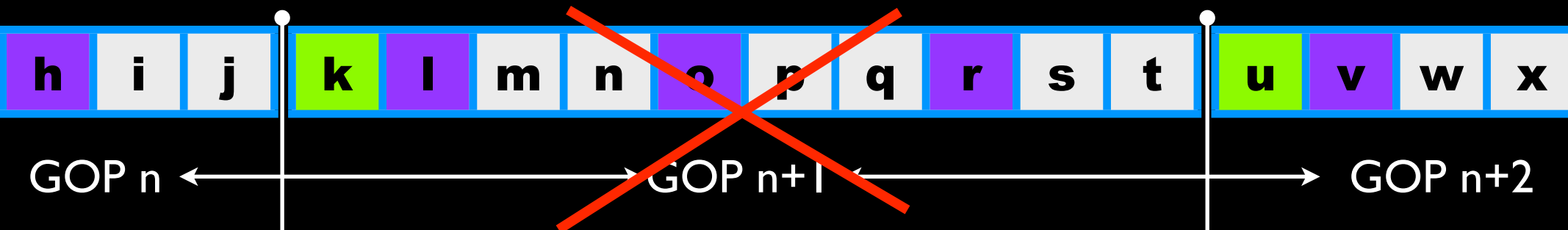
- * Mpeg Video ストリームを扱う上で
構成上意味のある**最小単位の画の集合**
- * 0.5秒程度(NTSCで15frame)がよくある
- * ものすごく乱暴に言うと、
I Pic から 次のI Picの手前まで
 - ✓ いや、それはちょっと乱暴杉...
- * 簡易的な編集アプリでは GOP単位の
切り貼りだけがサポートされる
 - ✓ 「**フレーム単位の編集**」には再エンコが必ず発生する
 - ✓ なるべく再エンコを避けようとすると、とたんに複雑になります

GOP単位の編集



実は Open GOP か Closed GOP かという
少しややコシイ話もあるけど今日は割愛

フレーム単位の編集



n~q をカットすると
r~tは参照先を失う



ストリーム全体を再エンコードするのは時間と画質の面でやりたくないが、
GOP n+1 だけを再エンコードすればいいというものでもない

ここまでのまとめ

ここまでのまとめ

*Mpeg2 Video ESのフレームには、

ここまでのまとめ

- *Mpeg2 Video ESのフレームには、
- *I, P, B の3つのピクチャタイプがあり、

ここまでのまとめ

- *Mpeg2 Video ESのフレームには、
- *I, P, B の3つのピクチャタイプがあり、
- *データサイズと画質の
トレードオフで決まる。

ここまでのまとめ

- *Mpeg2 Video ESのフレームには、
- *I, P, B の3つのピクチャタイプがあり、
- *データサイズと画質の
トレードオフで決まる。
- *Bitstream 上には、

ここまでのまとめ

- *Mpeg2 Video ESのフレームには、
- *I, P, B の3つのピクチャタイプがあり、
- *データサイズと画質の
トレードオフで決まる。
- *Bitstream 上には、
- *表示順とは異なる順番で格納されている。

ここまでのまとめ

- *Mpeg2 Video ESのフレームには、
- *I, P, B の3つのピクチャタイプがあり、
- *データサイズと画質の
トレードオフで決まる。
- *Bitstream 上には、
- *表示順とは異なる順番で格納されている。
- *GOP という単位で構成されている。

ここまでのまとめ

- *Mpeg2 Video ESのフレームには、
- *I, P, B の3つのピクチャタイプがあり、
- *データサイズと画質の
トレードオフで決まる。
- *Bitstream 上には、
- *表示順とは異なる順番で格納されている。
- *GOP という単位で構成されている。
- *アイを探したい。

以上。

石川さんに続く...