

WHITE PAPER

HQXの紹介

Grass Valley編集用コーデックHQXの特長は何か?どんな編集に最適なのか?

Desktop & Enterprise Solutions

竹本 晃

素材フォーマットの形式と種類が増加し、それにも増して顧客が要望する納品フォーマットが増加するのに伴って、NLEで使用するコーデックとして何を選択するかが、ますます重大かつ頭を悩ませる問題となっています。この記事では、いわゆる「中間コーデック」を使用することでどのようなワークフローが可能になり、最適な結果を得られるかをご説明します。次に中間コーデックに特に求められる特性を明確にし、Grass ValleyのHQコーデックファミリーの最新世代であるHQXをご紹介します。

続いて、いくつかのベンダーから提供されている中間コーデックソリューションを、画質、フレキシビリティ、ビデオだけでなくグラフィックスに対する適格性の観点から比較します。同クラスのコーデックを比較するテストを行い、HQXは他のポピュラーな中間コーデックに勝るとも劣らないことを示します。また、HQXは繰り返し編集に対して際立った性能を持っていることを示します。さらに、データレートや画像サイズに対する柔軟性、アルファチャンネルが一体としてサポートされている点などにも触れ、HQXが画像品質を重視するクリエイターやアーティストにとって最善の中間コーデックであることを結論づけます。

目次

はじめに.....	3
中間コーデック	3
どのような場合に中間コーデックを使用すべきか	4
よく使用される中間コーデック	4
HQXの進歩	5
10ビットコーデックによる優れた性能.....	5
色の精度.....	6
HQXのアーキテクチャと特性.....	7
HQXの設計諸元	8
コーデック性能と10ビット精度.....	8
原理1：すべての画像は情報を持っている.....	8
原理2：ランダムノイズはエントロピーの塊.....	9
原理3：ロッシーコーデックは、パディングだけでなくエントロピーも捨てる必要がある	9
画質の比較	10
第1世代性能	10
多世代性能	11
結論 - HQXによって最高の創造性を	12

はじめに

コンテンツ製作者が入力として用いる素材の供給源や形式の数は、日々倍増しているかのようです。RED Oneで撮ったRAW 4kのコンテンツが持ち込まれたかと思えば、Grass Valley LDK 3000からHDコンテンツを取り込むかもしれません。その一方で、グラフィックス部門からはAfter Effectsで作成した素材が送られてきます。入力側でのフォーマット数が爆発的に増加するのと足並みをそろえるように、顧客が要望する納品フォーマットの種類もまた増加しています。つまり、種々のビデオファイル、iPod用配信ファイル、web配信用ストリーム等。

編集者としてあなたが顧客に提供する価値は、創造性に加えて、技術的に正しいものを遅れなく作り上げる能力だと言えます。あなたが使用するNLEが素早く反応して、レンダリングを待つことなく、かっこよく決まるエフェクトや構成をあれこれ試してみることができれば、また、レイヤを積み重ねることで画像データを劣化させないか気にする必要がなくなれば、あなたの創造性を最大限発揮できると私たちは考えています。これらをすべて考え合わせると、HQXのような中間コーデックを用いるのが有効である、という理由は明白でしょう。ではまず全体像から見てみましょう。

中間コーデック

NLEワークステーションの性能は、画像処理能力と、ストレージからコンテンツをいかに素早く取り出したり格納したりできるか、で決まります。どちらかが限界に達すると、ワークステーションの動きはスローダウンしてしまいます。非圧縮HDコンテンツは155MB/秒もあり、多くのストレージシステムの最大帯域を超えてしまいます。ですから、非圧縮HDを用いるとすればCPUはその多くの時間をストレージの遅れを待つことに費やすことになるでしょう。

非圧縮ビデオが魅力的である理由は、言うまでもなくそれが理論上可能な最高の画質であるからです。中間コーデックは、非圧縮のマスター素材と見た目に同等の画質を保ちつつ、ストレージの帯域幅の問題を解決するために開発されました。それらに共通の特性として、次のものがあります。

- ・ストレージの帯域幅がボトルネックにならないために必要な圧縮率

圧縮率は概ね5分の1から8分の1の範囲です。これは今日のSATA 1.0ドライブなら単発で4チャンネルかそれ以上をサポートできることを意味しています。つまり、帯域をシェアするオーバーヘッドを差し引いても上限を超えないということです。

- ・極めて複雑な処理は必要としない

CPUは圧縮処理だけでなく、エフェクトや編集作業に必要なからです。

- ・圧縮されたことが目で見てわからない

中間コーデックはロッキー（圧縮に伴って劣化が発生する）ではありますが、撮影用や送出用のコーデックに比べてはるかに小さな劣化しか与えません。

このことにより、ワークステーションの主要な資源（CPUとストレージの帯域）をバランス良く使用することが可能となり、映像の品位に使用に耐えないほど落として我慢する必要がなくなります。

中間コーデックを使用する一般的なワークフローは、次の3つのステージから成ります。

- 1) ソース素材をブラウズし、記録を取り、長さを調整した後、撮影フォーマットから中間フォーマットに変換する
- 2) 編集工程
- 3) 完成したものを必要な納品フォーマット（複数種類の場合もある）に書き出す

最初と最後のステップは時間を要する場合もあります。そこで次のセクションでは、中間コーデックを使用することが正しい選択となるプロジェクトとはどんなものかを見ていきます。

中間コーデック（続き）

どのような場合に中間コーデックを使用すべきか

コーデックを選択することは、突き詰めると次の3つのいずれかになります。

1) 「システムコーデック」を使用する

収録に用いたものと同じコーデックを使うことが考えられます。これには純粋に、早く作業に取りかかることができる利点があります。収録と納品用の両方に、AVC-Intraのような高性能コーデックを使用する場合などはなおさらです。しかしながら、収録用コーデックがそれなりのものでしかなく、縮小した画像、低減したクロマ信号、8ビット画素などを使用する場合は、あまりお勧めできません。さらに、当然ですが、収録媒体に複数のフォーマットが混在して使用される場合は、この方法は利用できません。

2) 「フォーマット混在」で使用する

素材そのままのフォーマットを混在して使用するという選択肢もあります。これは、直ちに仕事を開始できるという点で、すばらしい選択です。しかし、この選択をおこなうと、編集過程で素材が何度も作り換えられる際に発生する問題のために、映像に手の込んだ加工を施し多層に重ねるなどの作業に対する魅力が損なわれてしまいます。

3) 残る選択肢、本ドキュメントの以降の部分はそのためにあるのですが、それは、中間コーデックを使用することです。中間コーデックは、次のような重要な利点をもたらします。

- ・ 込み入った編集を行う場合でも、画質を気に掛けるような心配が不要になります。このことで、技術的な問題に煩わされることなくクリエイティブな作業に専念できます。
- ・ 中間コーデックはコンテンツをハードディスクから取出し、画面に表示する最速の手段です。これが意味するところは、エフェクトのレンダリング待ち時間を気に掛けることなく、そのときそのときの本来の仕事に専念でき、編集作業の間最高にクリエイティブな状態であることができる、ということです。

Grass ValleyのEDIUS 6では、これらのいずれでも選択可能で、ワークフローをその時の仕事のタイプにうまく合わせるすることができます。

よく使用される中間コーデック

表1に、現在広く使われている主要な中間コーデックを紹介し、それらの主要な特性を要約しています。

Codec	Compression Range	Bit Rate (Mb/s)	Bit Depth	Chroma	Alpha Support
HQX	From 25:1 to 2:1	45 to 600	10	4:2:2 YCbCr	Yes, 10-bit
DNxHD	8:1 or 5:1	145 220	8 8 or 10	4:2:2 YCbCr 4:2:2 YCbCr	No No
ProRes	8:1 ProRes422 5:1 ProRes422 (HQ) 7:1 ProRes444	147 220 330	10 10 Up to 12	4:2:2 YCbCr 4:2:2 YCbCr 4:4:4 RGB	No No Yes
CineForm	From 10:1 to 3:1	120 to 400	Up to 12	4:2:2 YCbCr or 4:4:4 RGB or RAW	No

表1：中間コーデックの特性比較

CineFormとHQXは、次のような共通する特長をもっています。

- ・ ユーザーが圧縮率を設定できる
- ・ アルファチャンネルが一体としてサポートされている

アルファチャンネルのサポートが、なくてはならないほど有用であるのはなぜか、後ほど触れることにします。

DNxHDとProRes422が、数種類の固定されたビットレートしか使用できないこと、ビデオ形式としてアルファをサポートしていないことに留意してください。中間コーデックは非圧縮HDを代替するためのものであったことを思い出してください。非圧縮HDは10ビット標準値であるので、DNxHDが例外であるものの、ほとんどが10ビット標準値をサポートしています。この2ビットの違いがどれほど重要なものであるか、後ほど触れます。

次に、HQXがどのように進歩してきたのか、手短かに見ていきましょう。

HQXの進歩

HQXは、EDIUSノンリニア編集システムの一部として提供されてきたHQコーデックファミリーの進化の中で、最新のステージを担うものです。

Features	Released	Resolution	Chroma	Bit Depth	Alpha Support
HQ	2004, EDIUS 3	1440x1080	4:2:2	8	No
HQ with Alpha	2006, EDIUS 4	1920x1080	4:2:2	8	Yes
HQX with Alpha	2010, EDIUS 6	Up to 4k	4:2:2	10	Yes

表2：HQXの進歩

表2は、2004年から続くHQコーデックファミリーの主な進化をまとめたものです。これを見ると、解像度が着実に増加し、2006年以降はアルファサポートが4番目のコンポーネントとしてコーデックそのものに組み込まれ、HQファミリーの特長になってきたことがわかります。上で述べたように、これは真の中間コーデックと、収録や伝送向けに設計されたコーデックとの違いのひとつです。

解像度の増加に加え、HQXでの重要な拡張は10ビットビデオです。これによって追加されるビットによって何が得られるのか、さらに詳しく見ていきましょう。

10ビットコーデックによる優れた性能

画像がデジタル化される際、連続的に変化する信号が有限の分解能しか持たない数値列に変換されることにより、誤差が加わります。例えば、0から1ボルトの範囲を取り得る実信号があるとします。8ビット精度では、最小の符号値（0）を実信号範囲の最低値（0ボルト）に割り当て、最大の符号値（255）を実信号範囲の最高値（1V）に対応させることになります。10ビット精度では、10ビットで可能な最大の符号値である1023で1Vを表すことになるでしょう。8ビット符号化器は0から1Vの間を256の異なったレベルに対応させますが、10ビットでは1024の異なったレベルに対応させます。

それぞれのシステムで、0.4989844Vの実信号を符号化する際に何が起こるのかを考えてみましょう。

- ・ 0から255の8ビットの範囲では、本来の符号値は125.4ですが、8ビット符号化器は最も近い整数である125を選ぶしかありません。この結果、 -1.563mV の誤差が発生します。
- ・ 0から1023の10ビット精度では、本来の値である503.0753を下に丸めることになり、符号値は503となります。この場合に発生する誤差は $73.6\mu\text{V}$ です。

この例では、8ビット精度の量子化誤差は、10ビット精度のそれの約21倍の大きさになります。ただし、これはちょっと極端な例です。平均すると8ビット精度により発生する誤差は、10ビット符号化の場合の約4倍の大きさになります。この概念を図1と図2に示しています。

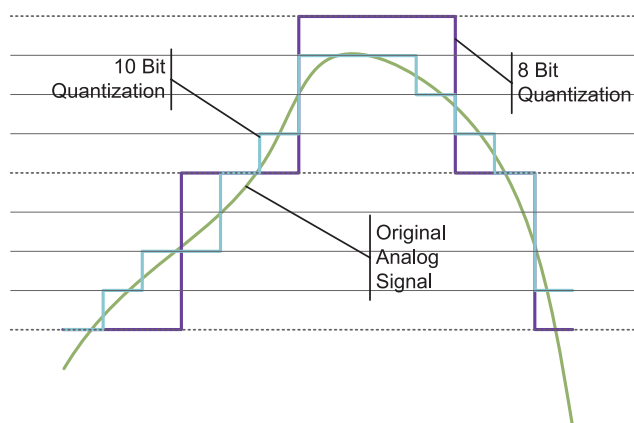


図1：10ビット量子化と8ビット量子化

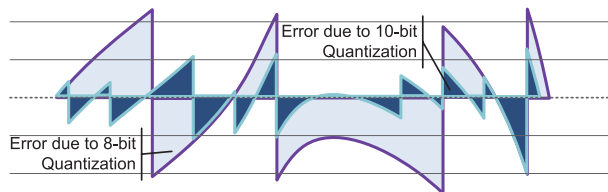


図2：量子化誤差の比較

量子化はカメラやフィールドレコーダなどの収録用機器で行われますが、これらは10ビット精度で収録できるものが増えてきました。収録機器で発生する誤差（あるいはノイズ）は避けられないものですが、発生するのは一度きりです。これと使用される中間コーデックと何の関係があるのでしょうか？答えは、より大きなノイズとなる誤差が加わるかどうかです。10ビット信号を圧縮する際に8ビット中間コーデックが用いられると、それが発生するのです。

10ビットコーデックによる優れた性能（続き）

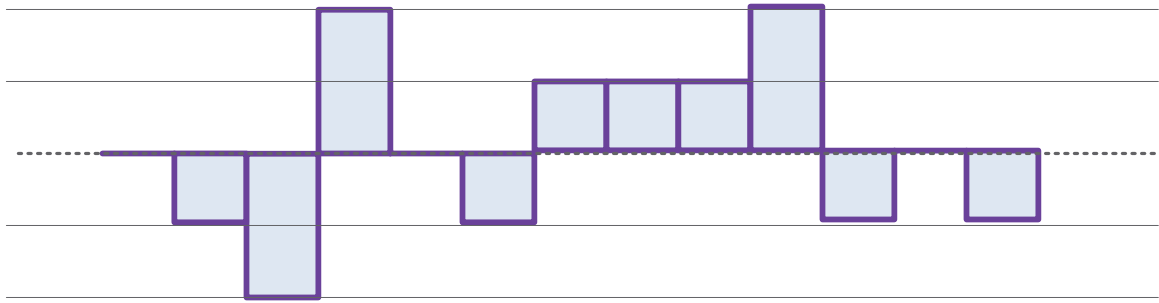


図3：10ビット信号を8ビットに丸めるときに加わるノイズ

図3は、この処理の結果を示しています。図4は、1枚の画像と、各画素の最下位2ビットに含まれるノイズの画像を、横に並べています。（注：3）



図4：画像の例と最下位2ビットのノイズ

量子化により画像にノイズが加わり、しばしば縞状や輪郭状の模様として目につく場合があります。縞が見えにくくなるように、誤差拡散と呼ばれる処理を行ってディザがかけられることがありますが、ノイズの総量が少なくなるわけではありません。このことがいかに大切であるかを後で述べますが、その前に、10ビット精度を用いることによるもうひとつの利点である色精度について考えてみましょう。

色の精度

色情報を符号化するために2つのコンポーネントが用いられるので、8ビットビデオは最大で50,000種類の色差値を表現できます（16より下と240より上の符号値は正規のものでないので、256ではなく224種類の値がCbとCrのそれぞれに存在します）。

10ビットビデオは最大でその16倍、すなわち800,000を超える種類の色差値を表現できます。

色を正確に表現できる利点で重要なものは、グリーンバック（あるいはブルーやレッド）から背景を抜き出す場合にあります。その色がマットのものかそうでないものか、余裕をもって識別でき、合成画像を自然に見せることができます。

では次に、HQXコーデックの構造に目を向け、その能力を調べ、中間コーデックに必要なとされる見た目上劣化のない画質を得るために、なぜ10ビット精度が役に立つのかを見ていきましょう。

（注：3）ノイズを見やすくするために40倍に拡大されています。

HQXのアーキテクチャと特性

HQXは多くのビデオコーデックに共通するアーキテクチャを持っていますが、ユーザーは図5に示すダイアログボックスを使って2つのキーパラメータを指定する必要はありません。Mパラメータにより、非圧縮のレートに対する割合として、最大ビットレートを決めます。つまり、M=10なら10：1圧縮ということであり、M=20なら5：1圧縮ということです。

Qパラメータは、どの程度強力的に圧縮アルゴリズムが画像を圧縮するかを指定します。これは、大きいQの値ほど強力的に圧縮することを意味しています。

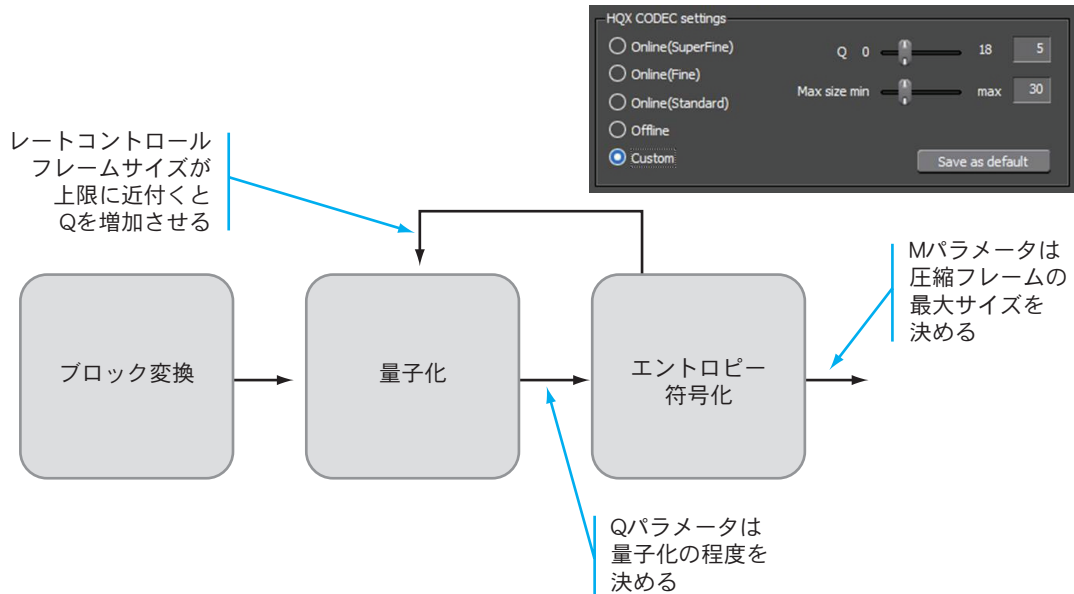


図5：HQXコーデックのアーキテクチャ

非圧縮画像は、まず小さなブロックに分割され、それぞれが変換されます。変換というのは、大量のペーパーワークを行う際に良くできたファイリング装置に仕分けするようなものだとお考えください。それによって画像情報が失われることはありません。しかし、それが自然な画像であればより良い構造に構成することにより、そのエントロピーを削減することができます。もちろん、画像がランダムなノイズであれば、それは不可能です。

量子化器は、どの情報が削減可能かを判断する部分にあります。それは、もっとも重要な情報を残すようにインテリジェントに実行されます。

エントロピー符号化器は、情報をロスしない算術的な符号化器で、WinZipなどのアルゴリズムと似たものです。フィードバックループは、最大フレームサイズを超えそうになった場合に、量子化器をより強力的に働かせます。

どのようなコーデックアーキテクチャを開発するにしても、コーデックの総合的な特性や性能、圧縮される対象の取り扱い方の諸元について、決定すべき事項がいくつか存在します。表3にHQXを開発するに当たって選択された項目を挙げています。

HQXのアーキテクチャと特性（続き）

HQXの設計諸元

パラメータ	HQXの諸元	特長
解像度	任意（4kまでサポート）	ユーザーのネイティブな解像度で素材を使用でき、しかも資源を大量に消費する収録用コーデックを使用する場合のような、処理時間の問題がありません。
クロマサンプリング	4:2:2	HQXはHD-SDIのクロマサンプリングと同一で、クロマ情報が劣化しません。
標本化精度	10ビット	低ノイズ、高性能圧縮、バンディングや疑似輪郭が発生しない、クロマキーがきれいに抜ける、8ビット素材（XDCAMなど）で出力する場合は、ゲインを変えたりレベルを上下させても不具合が発生しない。
アルファサポート	あり（10-bit）	グラフィックスをHQXでエンコードしておけば、高速にアクセスでき、高画質が保たれるので作業が単純化される。
ビットレート	ユーザー指定可能（非圧縮レートに対する圧縮比として）例をあげるとHDの場合で145、220、400MB/秒など	ユーザーが作業に合った圧縮率を指定できる。非圧縮並みの画質を必要とする場合には高いレートを設定できる。
ビットレート変動	可変（最大値は正確に規定できる）	CBR方式に比べて同じビット量なら高画質が得られる。最大値は厳密に守られるので、ネットワークやストレージの問題が発生しない。他の中間コーデックには、ビットレートが瞬間的に高くなるものがあり、問題を起こす可能性がある。
フレーム構造	イントラ符号化のみ	クリップ管理と編集が容易。

表3：HQXの設計諸元

コーデック性能と10ビット精度

今日、多種多様な画像用や動画用コーデックアルゴリズムが存在し、それぞれの支持者がそれぞれの技術を強調しますが、それがフラクタルであれ、ウェーブレットであれ、DCTであれ、それらの基本原理は共通です。

原理1：すべての画像は情報を持っている

その情報量は画像によって異なります。

- たとえば、カラーバーはほんの少しの情報しか持っていない。なぜなら画素の値を予想するのは、周囲の値を利用すればほとんどの場合簡単なことだからです。
- 細密な構造を多量に含む非常に込み入った画像は、もっと多量の情報を持っています。なぜならどの画素でもその値を正確に予測することはほとんど困難だからです。

情報科学者は、画像が持つ情報量を定量化する手段として、エントロピーという用語を使用します。日常用語で言えば、「意外性」という言葉に、エントロピーを理解する鍵があります。決まりきったパターンに一致することよりも、起こりにくいことや予想できないことの中に、多くの情報（つまりエントロピー）が含まれています。

画像のエントロピーを計測するには、正式なものではありませんが、WinZipのようなロスレスアルゴリズムを使って圧縮する方法があります。

						
Name	Bars	Alpha	Goldchill	Lena	Baboon	Noise
Picture Size (bytes)	1,049,760	307,200	414,720	262,144	65,536	250,000
WinZip File Size (bytes)	9,780	10,077	337,947	224,019	61,299	246,572
Estimated Entropy per Pixel (bits)	0.22	0.26	6.51	6.83	7.48	7.89

表4：いろいろな画像のエントロピー量

コーデック性能と10ビット精度（続き）

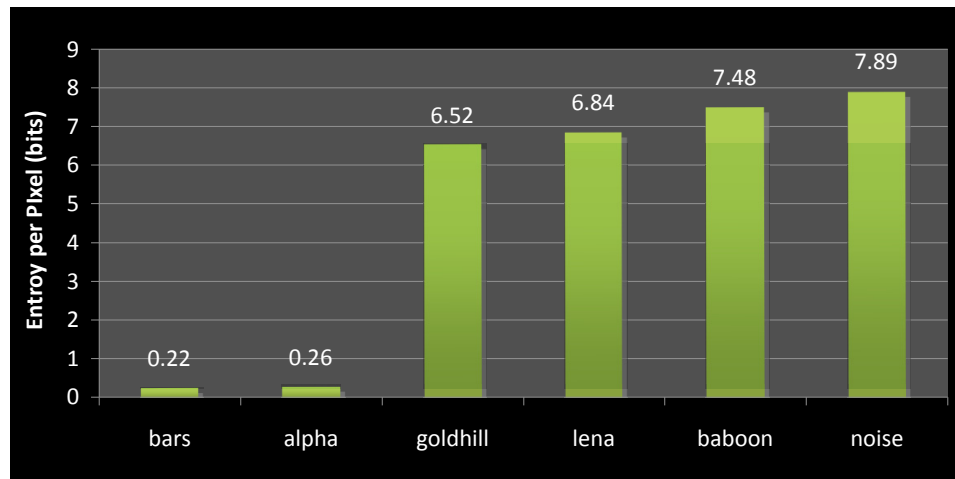


図6：いろいろな画像のエントロピー量

図6と表4は、いろいろな画像のエントロピー量を「画素当たりのビット数」として表現しています。これからはっきりわかるように、カラーバーやよくあるようなアルファチャネル画像はほとんどエントロピーを持っておらず、通常の画像はかなり多量に持っており、ノイズはエントロピーの塊と言っても過言ではありません。

原理2：ランダムノイズはエントロピーの塊

わかり易く言うと、WinZipのようなロスレスのコーデックは、ファイルの中にあるエントロピーを「パディング」（たとえば、10ビット語にするために8ビット語の後に付け加えた常にゼロの2ビット）から切り分けます。エントロピーを圧縮ファイルにセーブし、「パディング」は捨ててもいいのです。解凍する時はパディングを元に戻すことができ、元通りのデータを再構成することができます。

信号がランダムノイズであった場合は、すべてがエントロピーであり、捨てることのできるパディングは存在しません。脱線しますが、こういう理由で、すでに圧縮済みのファイルをそれ以上あまり小さくすることができないのです。たとえばJPEGをZIPしてもあまりディスク容量を稼ぐことができません。なぜなら、JPEGファイルのエントロピーは、JPEGアルゴリズムの性能の限界まで、すでに高くなっているからです。

原理3：ロッキーコーデックは、パディングだけでなくエントロピーも捨てる必要がある

ロスレスコーデックは、2:1程度の圧縮比（つまり、圧縮ファイルは圧縮前の半分のサイズになる）を得るにはうまく機能します。中間コーデックで使用される圧縮比はこれより高く、よく使われるのは5:1から8:1です。このことは、中間コーデックは「パディング」とともにエントロピーも捨てないといけないことを意味しています。これによって「ロッキーな」コーデックとなってしまいます。

当然ながら、オリジナル画像が多くのエントロピーを持っているほど、エンコードの過程でより多くのエントロピーが捨てられます。

こういう理由で画像データの20%を捨てたからといって（10ビットデータを8ビットに切り捨ててから圧縮しても）、何も得るものがないのです。なぜなら、そうすることで量子化ノイズが加わり、高画質や圧縮効率が得られるという期待に反する作用が働くからです。多少直観とは逆の結果となりますが、10ビット精度を持つコーデックと、10ビット入力のうちの8ビットだけを使用するコーデックとで、同じ画質を得るために必要なビットレートは同程度のものになります。このことは表5に示されており、8:1クラスの圧縮特性のものうち、10ビット精度を持つコーデック（HQXとProRes422）の画質は、全体にわたって8ビットコーデックであるDNxHDに非常に近いものになっています。

つまり、追加の2ビットが実質的に失うものなく利用できるわけですから、この利点を活かさない手はないでしょう。

画質の比較

異なったビデオコーデックの画質を客観的に比較する標準的な方法は、ある既知のシーケンスに対するピーク信号対雑音比（PSNR）によるものです。PSNRは画像圧縮により生じられたノイズを人間の視覚が感じる様子をモデル化したものではありませんが、客観的で、かつ自然な数値です（コラムをご覧ください）。コーデックによる差は、PSNRが2dBを超える場合に顕著になります。コーデックを比較する際には、同一の計測用映像を使用することが重要です。それはコーデックの性能は圧縮前の画像によって変わるからです。

ヨーロッパ放送組合（EBU）が非圧縮映像シーケンスを提供しており、コーデック性能を評価するのに広く利用されています。

EBUの映像シーケンスは1250フレームのビデオからなり、それを使用していくつかの中間コーデックの2つの重要性能を評価しました。

【コラム】 PSNRはPeak Signal to Noise Ratioの頭文字

PSNRは、コーデックを通った画像が圧縮前のオリジナルと比べてどれだけ違っているかを1つの数値で表す便利な量です。その計算に必要なのは、いくつかのシンプルなステップだけです。

- 1) 計測用画像をエンコードしてデコードし、結果の画像を得ます。
- 2) オリジナル画像から結果の画像を引き算します。その結果がコーデックにより発生した誤差です（ゼロなら誤差なし）。
- 3) 上の結果の各画素値を2乗し、すべて足し合わせます。次にそれを総画素数で割って、平均2乗誤差の値を得ます。
- 4) 画素が表現できる最大値（SMPTE 292では、最大輝度レベルを符号値940と定義しますが、ヘッドルームを1019まで許容します）と平均2乗誤差の比を求めます。
- 5) その比をdBで表します。

$$MSE = \frac{1}{N \cdot M} \cdot \sum_{i=1}^M \sum_{j=1}^N (ref_{i,j} - res_{i,j})^2$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{Peak^2}{MSE} \right)$$

第1世代性能

最初の性能指標は、単一世代のエンコードデコードによって発生する誤差の程度をそれぞれのコーデックについて計算した数値です。

圧縮比3:1、5:1、8:1の3種のクラスに属する中間コーデックについて、シーケンス中の各フレームのPSNRをグラフにしました。大きな数値ほど良い結果であることを覚えておってください。フレームごとに表示されていますが、最も上の線が最も画質が良いコーデックを表しています。

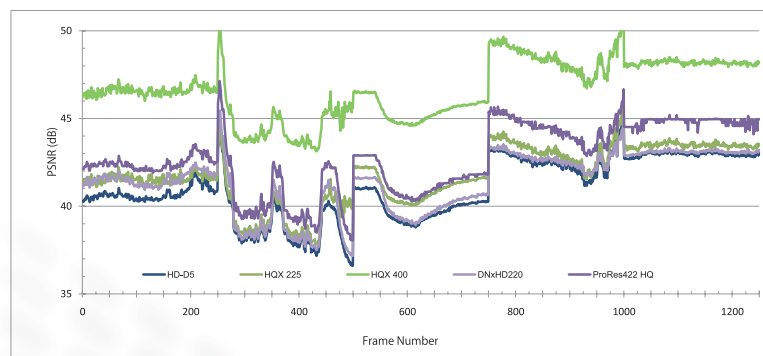


図7：5:1クラスと3:1クラスのHQXのPSNRの比較

図7は今回の1250フレームの計測用画像シーケンスのPSNRの結果です。最も高いPSNRのグラフは、Qパラメータが0でMパラメータが35の400Mb/秒のHQXです。お分かりのように、ビットレートが他のコーデックよりもかなり高いので、PSNRも大幅に良くなっています。5:1クラスのグラフでは、220Mb/秒のHQXは他の中間コーデックと比べてそんな色がありません。D5よりは明らかに良くなっています。

もっと低い平均ビットレートが必要なら、8:1クラスのコーデックを使用することになります。その場合のPSNRの結果を図8に示します。

画質の比較（続き）

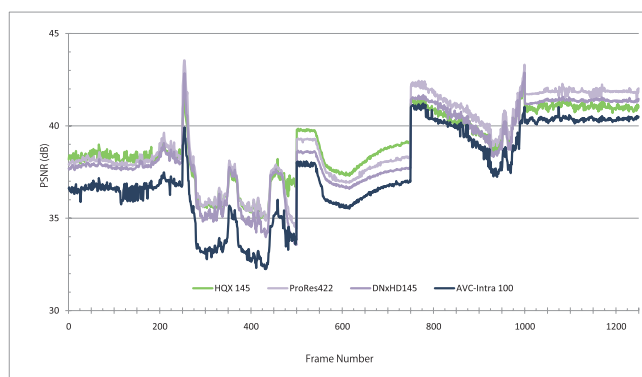


図8：8:1クラスのコーデック

図8によると、この圧縮比のクラスでは所々でHQXが最も良いコーデックになりますし、そうでない部分でもProRes422とDNxHDに非常に近い結果となっています。

Codec	8:1 Class		5:1 Class		3:1 Class	
	Mean bit rate	Mean PSNR	Mean bit rate	Mean PSNR	Mean bit rate	Mean PSNR
HQX	145	38.9 dB	224	41.8 dB	400	46.4
DNxHD	145	38.6 dB	220	41.4 dB		
ProRes422	147	39.1 dB	220	42.7 dB		
AVC-Intra	100	37.4 dB				
HD-D5			220	41.0 dB		

表5：画質比較のまとめ

全体にわたる画質の計測値を得るために、それぞれのコーデックについて、平均PSNRを計算しました。その結果を表5に示します。HQXは、ProRes422とDNxHDと比べて、同等のビットレートではそんな色ない画質が得られています。さらに、HQXでは、それによるメリットがある限り、さらに画質のダイヤルを上げることができます。

多世代性能

2つ目の重要な性能指標は、中間コーデックでは得に重要ですが、エンコードとデコードを何度にもわたって繰り返した場合に、どのようなことが起こるかです。多数のレイヤ構成を用いた編集では、オリジナルの入力素材が何世代も作り直されることが往々にして起こります。中間コーデックによる性能の違いが分かりやすいように、図9に示しました。

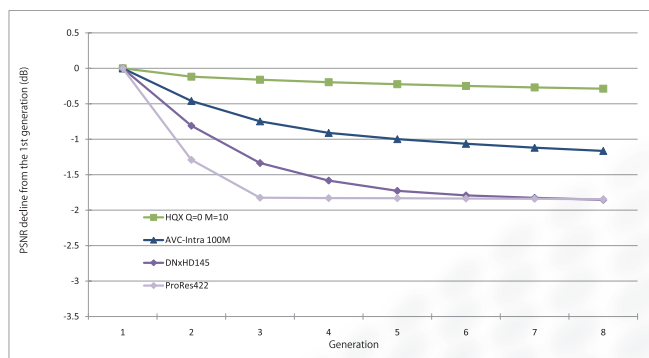


図9：中間コーデックの多世代性能

図9から、多世代にわたって画質を保つ点で、HQXが最も良く健闘していることがわかります。一方、DNxHDとProRes422では第2あるいは第3世代で、画質が急速に損なわれていくことがわかります。

結論 - HQXによって最高の創造性を

行う作業に合ったコーデックを使用できることが重要であることを述べました。早く作業に取り掛かることができるためには、システムコーデックまたは混成フォーマットを用いることが、おそらくもっとも良い選択です。しかし、画質と創造性が何にも増して重要である場合は、中間コーデックを選択すべきです。なぜなら、それによって次の重要な利点が得られるからです。

- ・ 入力素材を、見た目上劣化なく一貫して扱うことができます。これはグラフィックスのようにアルファチャネルを含んでいる場合でも同様です。
- ・ 技術面から画質を気に掛けることと、クリエイティブな作業とを、切り離して考えることができます。
- ・ 中間コーデックを用いることが、高品位の素材をディスクとスクリーン間で転送を行う最速の手段です。

画質についていえば、HQXでは、ProRes422やDNxHD など、NLEメーカーから提供される他の主な中間コーデックと比べて卓越した性能を得ることができます。

HQXは、他の中間コーデックと比べて、

- ・ アルファチャネルを一体としてサポートしているので、ビデオの色空間でありながらグラフィックスを高速に容易に扱うことができます。
- ・ エンコードとデコードを何回も繰り返す際の劣化を最小に抑えられるので、他の多くの中間コーデックの場合より最終出力の画質が良好です。
- ・ ユーザーがビットレートや量子化の行い方を設定できるので、あなたのクライアントに対して競合相手より正味10dB優れた画質を提供できます。

つまり、入力素材が多様で、良い出来を要求され、納期は短い、という事態に直面しても、高速で見た目上劣化なしという中間コーデックの一般的な利点に加え、HQXならではの卓越した性能によって、あなたの創造力を高めることができ、クライアントにとっての価値を高めることができます。