

目次

- [総説](#)
- [動画](#)
- [データ量の単位](#)
 - [コンピュータでの単位](#)
 - [バイト](#)
 - [ビット](#)
- [エンコード](#)
 - [エンコードの意義](#)
 - [エンコードによる劣化](#)
 - [エンコードにかかる時間](#)
 - [1パスエンコードと2パスエンコード](#)
- [コーデック](#)
 - [コーデックの意義](#)
 - [映像コーデック](#)
 - [音声コーデック](#)
 - [その他のコーデック](#)
 - [可逆圧縮コーデック](#)
 - [非圧縮](#)
- [ビットレート](#)
 - [ビットレートの意義](#)
 - [ビットレートの設定](#)
 - [CBRとVBR](#)
 - [動画のファイルサイズの計算方法](#)
- [フレームレート](#)
 - [フレームレートの意義](#)
 - [フレームレートの設定](#)
- [サンプリングレート](#)
 - [サンプリングレートの意義](#)
 - [さまざまなサンプリングレート](#)
 - [サンプリングレートと量子化ビット数](#)
 - [サンプリングレートとビットレート](#)
- [ビデオサイズ](#)
 - [ビデオサイズの意義](#)
 - [ビデオサイズの例](#)
 - [ビデオサイズによる違い](#)
- [アスペクト比](#)
 - [アスペクト比の意義](#)
 - [アスペクト比の変更が必要なケース](#)

レターボックスなど

- コンテナ
 - コンテナの意義
 - AVIコンテナにおける容量の壁
 - コーデックチェッカー
 - 関連ページ
-

総説

- 動画作成・編集における基本的な概念について、簡単に解説します。以下の解説では、ニコニコ動画を「ニコニコ」と略しています。

動画

- **動画**とは、動く画像のことをいいます。画像が動いていれば音声がなくても動画といえますが、通常は動画といえば**映像と音声**からなるものをいいます。
- 動画を構成している**映像**は、静止画像を連続してすばやく表示したものです。1枚1枚の絵はたんなる**静止画**でしかないのですが、パラパラマンガのように高速表示することによって、人間の目には動いているように見えるのです（**残像効果**）。この1枚1枚の画像のことを、「**コマ**」といたり「**フレーム**」といたりします。



1~4の静止画像を順に高速で切り替えて表示すると、女性が左から右に移動している動画となります。静止画像はフレームといい、最初のフレームを「1フレームめ」、2番めのフレームを「2フレームめ」というように表現します。

- 映像のことを**ビデオ**、音声のことを**オーディオ**ということがあります。

データ量の単位

コンピュータでの単位

- 私たちは日常生活で、さまざまな単位を用いています。単位には、メートル、グラム、リットルなどありますが、コンピュータの世界でも同じように単位という概念があります。データ量を表すための単位です。動画を扱うさいデータ量が重要になることが多いので、常識として理解しておきましょう。
- まず、**バイト**と**ビット**という用語を覚えておきましょう。**1バイト=8ビット**です。これは常識として記憶しておく必要があります。バイトは**B**と大文字で表記し、ビットは**b**と小文字で表記します。

バイト

- バイトは、バイト（B） キロバイト（KB） メガバイト（MB） ギガバイト（GB） テラバイト（TB）の順に単位が大きくなっていきます。そして、1KB = 1,024B、1MB = 1,024KB、1GB = 1,024MB、1TB = 1,024GBです *1。動画を扱う場合、バイトは動画のファイルサイズを表すときに用いられます。

ビット

- ビットもバイトと同じように、ビット（b） キロビット（kb） メガビット（Mb） ギガビット（Gb） テラビット（Tb）の順に単位が大きくなります。動画を取り扱う場合、ビットは後述する ビットレート を表すときに用いられます。1kbps=1,000bps、1Mbps=1,000kbpsです。

画面の上へ

エンコード

エンコードの意義

- **動画は膨大なファイルサイズであるため扱いにくい** ものです。たとえば、HDDの容量が200GBだとして、計100GBの動画ファイルをそのまま保存しておくのはやっかいです。また、大きなデータ量の動画をインターネット上にアップロードしたり、あるいはダウンロードしたりするのはとても時間がかかります。
- そこで、**ファイルサイズを小さく** しようというのが **エンコード** とよばれる作業です。動画をエンコードすれば、たとえばエンコードまえに500MBあったファイルサイズを100MBにすることが可能です。DVD-Videoの動画も地上デジタル放送の動画も、すべてエンコードされています。
- エンコードを行うソフトウェアのことを **エンコーダー** といいます。動画編集ソフトには必ずエンコードするための機能があり、動画編集とエンコードは切っても切れない関係です。エンコードは、エンコ、**変換**、**圧縮** などといわれることがあるので覚えておきましょう。エンコードは動画編集における最重要単語です。
- エンコードの対義語は **デコード** といいます。デコードはエンコードしたファイルを元に戻すことをいいますが、動画の世界では **エンコードされている動画を再生・視聴すること** であるとイメージしてください。

エンコードによる劣化

- エンコードは、動画編集するさいに必須の作業です。しかし、注意が必要なのは動画を **エンコードすると映像・音声は劣化する** という点です。つまり、エンコードすると画質・音質が落ちてしまうわけです。よく「動画編集してエンコードしたら画質が落ちました。なぜでしょう」と質問する人がいますが、エンコードしているのですから、それはあたりまえのことなのです。



エンコードによる画質劣化の例。エンコードまえの画像（左）とエンコード後の画像（右）を比較すると劣化の程度がよくわかります。

- どの程度品質が落ちるかはエンコードするさいの設定によります。しかし大胆にいつてしまえば、**画質・音質を落とすほどファイルサイズは小さくなり、画質・音質を維持しようとするほどファイルサイズは大きくなる**と理解しておいてください。**品質とファイルサイズのバランスをとる**のがエンコードの難しいところなのです。高画質・高音質を維持しながら、できるだけファイルサイズを小さくすることをめざしましょう。
- **エンコードによって劣化した映像・音声は元の品質に戻すことができません**。したがって、エンコードすればするほど画質・音質は落ちていきます。このように、劣化を伴う圧縮のことを**不可逆圧縮（非可逆圧縮）**といいます。たとえば、Aという動画編集ソフトで編集後エンコードし、できあがった動画をさらにBという動画編集ソフトで編集してエンコードすると、最初より動画の品質は落ちるわけです。そこで**エンコードする回数は最小限に抑えましょう**。

エンコードにかかる時間

- **エンコードには時間がかかります**。たとえば、15分の動画をエンコードするのに45分かかることがあります。エンコードに要する時間というのはさまざまな要素により決定されますが、もっとも重要な要素のひとつは**CPUの性能**でしょう。CPUは人間でいう頭脳に該当します。エンコード条件が同じである場合、**性能のよいCPUであればエンコード時間は短くすみませし、性能の悪いCPUであればエンコード時間は長くかかります**。
- エンコードに時間がかかるということで、エンコードしつつ就寝したり、登校・出勤する人もいます。動画編集ソフトによっては、エンコードが終了したさいに自動的にPCの電源を落とす設定ができるものもあります。こういった設定をうまく利用すると便利かもしれません。
- 通常、**エンコード中はCPUに大きな負荷がかかります**。できるだけ他の重い作業をしないようにして、エンコードに時間がかからないようにしましょう。CPU負荷を調べるには、タスクバー（画面最下部のバー）上のなにもないところで右クリックし、「タスクマネージャ」「パフォーマンス」タブの順にクリックして、「CPU使用率」を見てください。

1パスエンコードと2パスエンコード

- 動画をエンコードするさいは、データの圧縮のほか解析が行われます。このとき、データの解析と圧縮を同時にまとめて行う方式を**1パスエンコード**といい、2度に分けて行う方式を**2パスエンコード**といいます。1パスエンコードと2パスエンコードを比較した場合、前者は後者よりもエンコード時間が短縮される反面、画質は劣ります。

- 画質にこだわるなら、2パスエンコードしましょう。ただし、2パスエンコードするとエンコード時間がさらに長くなります。最初にデータの解析を行い、つぎに圧縮を行うという2段構成になっているからです。

画面の上へ

コーデック

コーデックの意義

- 動画を HDD に保存（エンコード）するさいは、**コーデック** とよばれるソフトウェアが必要です。つまり、コーデックを使って動画のファイルサイズを小さくすることになります。
- コーデックを意識するのは、動画編集するときがほとんどでしょう。たとえば、**動画編集ソフトで動画を読み込もうとしても、コーデックが原因で読み込めないことがあります**。また読み込めても、**映像を表示できない、音声が届かない**ということがあるかもしれません。
- ここで気をつけたいのは、**動画編集ソフトで動画を読み込むためにはエンコードのさいに使用したコーデックが必要**ということです。たとえば、Aというコーデックを使って動画をエンコードした場合、同じAというコーデックがなければ動画編集ソフトで動画を読み込めません。インターネットでダウンロードした動画ファイルを動画編集ソフトで開けず、エラーが表示される、というのはよくあるパターンです。
- 動画のコーデックは、**映像コーデック** と **音声コーデック** に分類できます。**どのコーデックを使うかによって、映像や音声の品質、およびエンコード時間が異なってきます**。

映像コーデック

- 数ある映像コーデック（ビデオコーデック）のなかで覚えておきたいのは、MPEG-2とMPEG-4 AVC/H.264です。まずはこのふたつのコーデックを覚えておきましょう。
- まず **MPEG-2** コーデックですが、これはキャプチャーボードを購入したときに付属されている キャプチャーソフト で録画したさいに使われています。また、MPEG-2の圧縮技術は地上デジタルTV放送やDVD-Videoにも利用されています。名称くらいは聞いたことがあるという人がいるかもしれません。
- 他方、**MPEG-4 AVC/H.264** コーデックはBD-Videoやワンセグ放送などで利用されている技術で、MPEG-2よりも新しい標準規格です。ニコニコに動画を投稿するさい、MPEG-4 AVC/H.264コーデックを使用したうえで複数の要件を満たすと、高画質な動画を投稿することができます。
- 詳細は別ページで述べますが、動画をニコニコに投稿するまでの定番パターンとして、**TVゲームを録画してできたMPEG-2形式の動画を、最終的にMPEG-4 AVC/H.264形式の動画にエンコードして投稿**、というのがあります。なんとなくでよいので、この流れを意識しておいてください。

音声コーデック

- [MP3](#)（MPEG Audio Layer-3）や [AAC](#) という名称を聞いたことがあると思いますが、これらはMPEGという団体が策定した音声コーデック（オーディオコーデック）です。AACのほうがMP3よりも高音質・高圧縮であるとされます。[AC3](#) はDVD-VideoやBD-Videoなどで使用されています。

その他のコーデック

- ほかに、映像コーデックとして [Windows Media Video 9 VCM](#) や [DivX](#)（ディビックス）などがあります。最近では[アマレコTV](#)や[アマレココ](#)といったキャプチャソフトを使用するために、[AMV](#) という映像コーデックをインストールする人も増えてきました。

可逆圧縮コーデック

- これまで見てきたコーデックはすべて非可逆圧縮のコーデックです。そのため、エンコードするたびに画質・音質は劣化していきます。しかし、[可逆圧縮](#) とよばれるコーデックを使用してエンコードすれば、画質・音質は劣化しません。可逆圧縮できるコーデックとして、[Huffyuv](#)（ハーフワイキューブイ）や [Ut Video Codec Suite](#)、[Lagarith Lossless Video Codec](#) などがあります。
- 可逆圧縮できるコーデックは動画編集するうえでとても便利です。たとえば、Aという動画編集ソフトで編集後、Bという動画編集ソフトでさらに編集してエンコードする場合は、Aの時点で可逆圧縮のコーデックによってエンコードしておくのです。そうすれば無劣化のままBで動画を読み込むことができるわけです。
- ただ、可逆圧縮コーデックを使用する場合、ファイルサイズが大きくなる点に注意してください。動画のファイルサイズが巨大になるため、そのまま [HDD](#) に保存しておくのには向いていません。[非可逆圧縮のコーデックを使ってエンコードし、ファイルサイズを小さくしておく](#) 必要があります。

非圧縮

- 動画のファイルサイズが増えてでも品質を重視したい場合、コーデックを使用せずに動画または音声を保存することがあります。このように動画または音声を圧縮していない状態を [非圧縮](#)（未圧縮/無圧縮）といいます。たとえば、動画編集ソフトで [未圧縮AVI](#) という項目を選択すると、映像を圧縮せずに動画を保存します。また、[PCM](#)（リニアPCM/[WAV](#) /[WAVE](#)）という項目を選択すると、音声を圧縮せずに保存します。
- もっとも、映像についていえば通常は非圧縮にする意味はほとんどありません。非圧縮にすると、画質は劣化しないもののファイルサイズが巨大になります。そこで、画質を劣化させたくない場合は非圧縮にはせず、可逆圧縮のコーデックを使ってエンコードします。[可逆圧縮コーデックを使えば画質を劣化させることなく、圧縮しないときよりはファイルサイズを小さくできる](#) からです。

ビットレート

ビットレートの意義

- **ビットレート**（bps/bits per second）とは、1秒間に送受信できるデータ量を表す単位です *2。ビットレートは、**映像ビットレート**（ビデオビットレート）と**音声ビットレート**（オーディオビットレート）に分類されます。そして、**ビットレートが高いほど高品質になり、ファイルサイズも大きくなります**。値が小さければその反対です。

	画質	ファイルサイズ
低ビットレート	低画質	小さい
高ビットレート	高画質	大きい

- たとえば、1Mbpsの映像データと10Mbpsのそれとでは、後者のほうが高画質となります。音声についても同様で、64kbpsと192kbpsの音声データを比較した場合、後者のほうが高音質です。ただ、同じビットレートであっても映像の状態、コーデックの種類・設定などによっても完成後の動画の品質は異なります。そのため、ビットレートが高いからといって必ずしも高品質とは限りません。

ビットレートの設定

- ニコニコに動画を投稿する場合、このビットレートに頭を悩ませることが多くなります。というのは、ニコニコに投稿できる動画のファイルサイズは、**40MBまたは100MB以下** というのがありますが、**ビットレートを増やすほど映像・音声は高品質になるもののファイルサイズが増えるため、一定のファイルサイズに収まるようにビットレートをやりくりしないといけない**からです。
- たとえば、ファイルサイズを40MB以内に収めるという前提のときに、映像を500kbpsに設定する場合と、200kbpsに設定する場合を考えてみましょう。500kbpsにすれば200kbpsにした場合より高画質にできますが、そのぶん動画の再生時間は短くなります。他方、映像を200kbpsにすれば画質はいまいちになりますが、動画の再生時間は長くできます。
- また、映像と音声それぞれに、どれくらいのビットレートを割くべきかという問題もあります。たとえば、画質を重視するなら多くのビットレートを割かなければいけません。そうすると音質に割けるビットレートが少なくなってしまいます。このように、ニコニコに動画を投稿するさいにはビットレートに悩むことが多々生じます。

CBRとVBR

- CBRとVBRという概念もよく登場します。簡単に見ておきましょう。**CBR**（Constant Bit Rate）とは、動画や音声を圧縮するさいにビットレートが変動せず固定である方式をいいます。他方、**VBR**（Variable Bit Rate）とは、動画や音声を圧縮するさいにビットレートが変動する方式をいいます。

- 動画というのは、シーンによって必要なビットレートが異なります。たとえば、動きの激しいシーンでは多くのビットレートが必要ですし、動きの少ないシーンでは少ないビットレートですみます。もし動きの激しいシーンで少ないビットレートしか割り当てていないと画質が乱れてしまいますし、動きの少ないシーンで多くのビットレートを割り当てると無駄にファイルサイズが増えてしまいます。
- したがって、合理的に考えればシーンによってビットレートを変動させなくてはなりません。このよう考え方をしているのがVBRです。しかし、CBRは動きの激しいシーンであっても少ないビットレートしか割り当てられなかったり、動きの少ないシーンであっても多くのビットレートを割り当ててしまうことがあります。したがって、ファイルサイズが同一ならばVBRのほうが高品質になります。

動画のファイルサイズの計算方法

- 動画のエンコードのさいに、どれくらいのファイルサイズになるのか、ある程度予測可能です。動画のファイルサイズは、**ビットレート×再生時間** で考えましょう。たとえば、映像に400kbps、音声に64kbps使って動画を作成し、動画の再生時間が10分である場合を想定します。この場合、以下のようにして動画のファイルサイズを計算します。
1. 映像と音声に使用するビットレートの和を求める。
 1. $400+64=464$ kbps
 2. 1で求めた数の単位をbpsに直す（1,000を掛ける）。
 1. $464 \times 1,000=464,000$ bps
 3. 動画の再生時間の単位を秒に直す。
 1. $10 \times 60=600$ 秒
 4. 2と3で求めた数を掛け合わせる。
 1. $464,000 \times 600=278,400,000$ bps
 5. 4で求めた数の単位をバイトに直す（8で割る）。
 1. $278,400,000 \div 8=34,800,000$ B
 6. 5で求めた数の単位をキロバイトに直す（1,024で割る）。
 1. $34,800,000 \div 1,024=約33984.4$ KB
 7. 6で求めた数の単位をメガバイトに直す（1,024で割る）。
 1. $33984.4 \div 1,024=約33.2$ MB
- **ビットをバイトに直す** のを忘れないようにしてください。1バイトは8ビットですから、ビットをバイトに直すときは8で割ります。ただ、エンコーダーによっては上のように自分で計算せずとも、必要な数値を入力するだけで自動的に計算してくれる場合もあります。
 - 動画のファイルサイズ = ビットレート × 再生時間なので、**ビットレート = 動画のファイルサイズ ÷ 再生時間** となります。

フレームレート

フレームレートの意義

- さきほど、動画はパラパラマンガのように少しずつ異なる静止画像を高速で連続表示したものだと言いました。このとき1秒間に表示される画像の数を表したのが **フレームレート** です。単位はfpsです。基本的には **フレームレートの値が増えるほど滑らかに動き**、フレームレートが減ればその反対です。TV放送は30 (29.97) fpsですが、映画やアニメなどは24fpsで制作されています *3。
- 動画は、静止画像を連続して高速表示したものです。30fpsならば、30枚の異なる静止画像を1秒間で表示します。15fpsならば、15枚の異なる静止画像を1秒間で表示するということです。したがって、両者を比較した場合、前者のほうが動きの滑らかな動画となります。さらに、60 (59.94) fpsならば30fps以上に滑らかな動画となります。
- 参考までにフレームレートの異なる動画を用意しました *4。 **10fpsの動画よりも30fpsの動画のほうが滑らかな動き** をしていることが理解できるでしょう。60fpsの動画は、30fpsの動画よりもさらに滑らかに動きます。ただし、PCのスペックによってはカクカクするかもしれません。
 - 10fps (2.67MB)
 - 30fps (5.63MB)
 - 60fps (8.01MB)
- フレームレートは、インターレース方式やプログレッシブ方式と密接な関係があります。29.97fpsで動いているTV放送がとても滑らかな動きに見えるのは、インターレース方式とよばれる技術を用いているからです。

フレームレートの設定

- 基本的に **フレームレートは動画に合わせて設定** するのが通常です。たとえば、30fpsのゲームを録画する場合は30fpsに設定します。たまに30fpsのゲームを録画するさいに60fpsで録画する人がいますが、これは画像を水増しするだけで意味がありません。きちんとした60fpsの動画ならば、動画編集ソフトで動画を読み込んだときにすべて異なる画像が表示されます。
- 60fpsのゲームを60fpsで録画すると、とても動きが滑らかな動画になります。ただ、60fpsでの録画はPCに負荷がかかるので、高いPCスペックが必要となります。また、60fpsの動画を再生するときも同様です。ニコニコに投稿する動画については、通常は30fpsで録画しておき30fpsで エンコード しておけば充分です。

画面の上へ

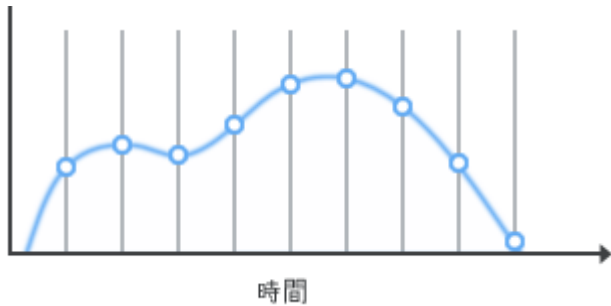
サンプリングレート

サンプリングレートの意義

- 音声でいうところの **サンプリングレート** (サンプルレート/サンプリング周波数) とは、アナ

ログ信号からデジタル信号への変換を1秒間に何回行うかを示したものです。単位はHz（ヘルツ）です。

- 通常、音は目に見えませんが、**波形**で表現することができます。この波形を再現すれば、原音と同じ音をコンピュータ上で再生することが可能です（音声のデジタル化）。そして、音声をデジタル化するさい、1秒間に何回採取（サンプリング）するのかというのを表すのがサンプリングレートです。サンプリングレートが高いほど波形を正確に再現できます。



- サンプリングレートが高いと高音質になりますが、ファイルサイズが大きくなります。サンプリングレートが低いと低音質になりますが、ファイルサイズは小さくなります。そのためサンプリングレートを設定する場合は、品質とファイルサイズを考慮したうえで行うこととなります。

さまざまなサンプリングレート

- CDの場合、1秒間に44,100回というスピード（間隔）でサンプリングします。したがって、CDのサンプリングレートは44,100Hzです。そして、1kHz（キロヘルツ）は1,000Hzなので、44,100Hzは44.1kHz（キロヘルツ）ということになります。CDのサンプリングレートは覚えておきましょう。
- 動画を作成するうえで重要なのは事実上、**44.1kHz**と**48kHz**のふたつです。キャプチャーボード使用時に録画した動画の音声は48kHzです。このままのサンプリングレートでニコニコに動画を投稿することになるでしょう。サンプリングレートをまちがえると、映像と音声がずれる現象（音ずれ）が生じることがあります。

サンプリングレートと量子化ビット数

- サンプリングレートとセットでよく登場するのが**量子化ビット数**（ビット深度 / サンプリングビット数）です。これはどのくらいの精度で原音の信号レベル（大きさ）を再現するかということを表したものです。たとえば、量子化ビット数が16ビットであれば65,536（2の16乗）段階で波形を再現します。
- 動画を作成するさいに量子化ビット数を気にする必要はありませんが、**16ビット**または24ビットという数値を頭の片隅に記憶しておいてください。CDの量子化ビット数は16ビットです。

サンプリングレートとビットレート

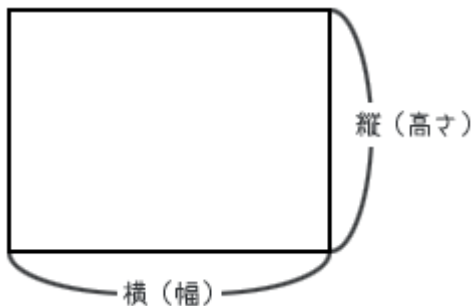
- 余談ですが、サンプリングレートと量子化ビット数、チャンネル数（ステレオなら2、モノラルなら1）がわかっているならば、サンプリングレート×量子化ビット数×チャンネル数という式でビットレートを算出できます。たとえば、CDのビットレートは $44,100 \times 16 \times 2$ で1411.2kbpsとなります。

画面の上へ

ビデオサイズ

ビデオサイズの意義

- ビデオサイズ** というのは、表示されている動画の横と縦（幅と高さ）の大きさのことをいいます。640×480、1280×720というように、**横×縦** で表記します。ビデオサイズにはさまざまな呼称があり、**表示サイズ** や **画面サイズ**、**画像サイズ**、**解像度**、**総画素数** とよぶこともあります。



- 画素** というのは、デジタル画像を構成する最小単位のことです。画素は英語で **ピクセル** ともいいます。方眼紙を思い浮かべてください。この方眼紙の1個1個のマス目がピクセルであり、方眼紙全体で1枚の画像を構成しています。そして、ピクセルが横に640個、縦に480個並んでいれば、ビデオサイズは640×480ということになります。

ビデオサイズの例

- 動画を取り扱う場合、ビデオサイズはある程度決まっています。動画編集でよく登場するビデオサイズとして、320×240、640×480、1280×720、1920×1080などがあげられます。最後のふたつは **HD動画** ともいい、1920×1080のビデオサイズは **フルHD** とよぶことがあります。
- ニコニコ用には、512×384、512×288、640×360を覚えておきましょう。通常はこれらのビデオサイズのうち、いずれかのサイズに動画を エンコード してニコニコに投稿するからです。従来型のキャプチャーボードで録画した場合は、720×480のビデオサイズになります。

ビデオサイズによる違い

- ビデオサイズが大きいほど、細かな箇所まで表現された映像にすることができます（参考）。これは電光掲示板を例にするとわかりやすいでしょう。電球の数が多いほど緻密に字を表現できるわけです。つまり、ビデオサイズが大きいということは、それだけ静止画像を構成するピク

セルが多く、高精細な映像を表現できることを意味します。

- しかし、その反面、**ビデオサイズが大きくなると多くの映像ビットレートが必要** となります。かりに、ビデオサイズが大きいのに低ビットレートの映像である場合は、画質が破綻（はたん）します。また、ビデオサイズの大きい動画を再生するさいにはCPUにかかる負荷も大きくなるため、PCのスペックがたりない場合は正常に再生することができません。
- 小さいビデオサイズの動画を大きいビデオサイズにエンコードしても、画質は向上しません。むしろ大きく劣化します。また、小さいビデオサイズの動画を再生し、それを動画プレイヤーで**画面いっぱい**に拡大しても映像が引き伸ばされる だけです。画像がボケたり、粗がめだつようになります。逆に、より小さいサイズにしたほうがシャープな画質になることもあります。ビデオサイズはケースバイケースで考えましょう。

画面の上へ

アスペクト比

アスペクト比の意義

- **アスペクト比**（アス比/SAR）とは、ここでは **画面の横と縦の比率** のことをいいます。アスペクト比は **画面サイズ** との関係で覚えておくとよいでしょう。上で紹介した画面サイズのアスペクト比を下表で確認します。ほかにもパターンが多数ありますが、「720×480ならばアスペクト比を4:3にする」というように最初は下表を参考にしてください。

320×240 512×288 512×384 640×360 640×480 720×480 1280×720 1280×800 1920×1080

4:3

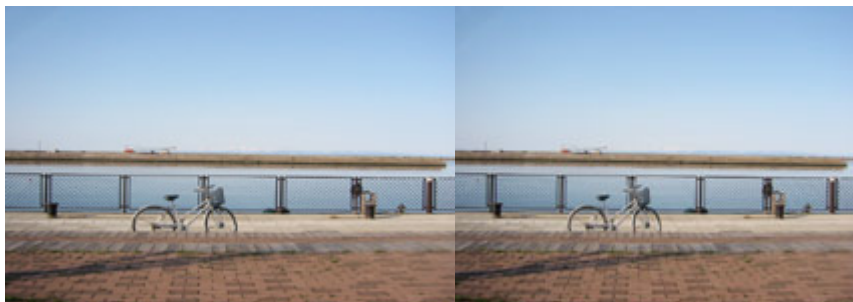
16:9

- アスペクト比の把握は重要です。なぜなら、アスペクト比をまちがえて**エンコード**すると**映像が縦や横に伸びてしまう**からです。たとえば、画面サイズが1280×720（16:9）であるにもかかわらず512×384（4:3）のサイズに動画をエンコードしてしまうと、**映像が縦に伸びません**。また、画面サイズが640×480（4:3）の動画を640×360（16:9）のサイズにエンコードしてしまうと、**映像が横に伸びます**。
- そこで、原則として**エンコードまえとエンコード後でアスペクト比を変更しない**ようにしましょう。たとえば、1280×720（16:9）のアスペクト比ならば640×360（16:9）にエンコードします。ニコニコで動画を視聴しているさい、登場キャラクターが縦に伸びてしまっている動画を見たことがあるかもしれません。これは、**投稿者が動画を投稿するまでのどこかの過程でアスペクト比をまちがえてしまったことが原因**です。

アスペクト比の変更が必要なケース

- いまアスペクト比は変更してはいけないと述べました。しかし、**場合によっては例外的にアスペクト比を変更する必要があります**。それはDVD/BDレコーダーや従来の**キャプチャーボード**で**TVゲームを録画**したときです。これらの機器でTVゲームを録画すると、画面サイズが**720×480**の動画になります。

- 同サイズの動画をPCで見ると、映像が少し横に伸びています（アスペクト比は3:2）。もしアスペクト比を変更しないまま動画を再エンコードすると、横に伸びた映像になってしまいます。そこで、**アスペクト比が4:3になるように画面サイズを変更**（リサイズ）して再エンコードしましょう。たとえば640×480や512×384など、アスペクト比が4:3になるようにするのは、そうすれば正常なアスペクト比になるので、横に伸びた映像を修正できます。

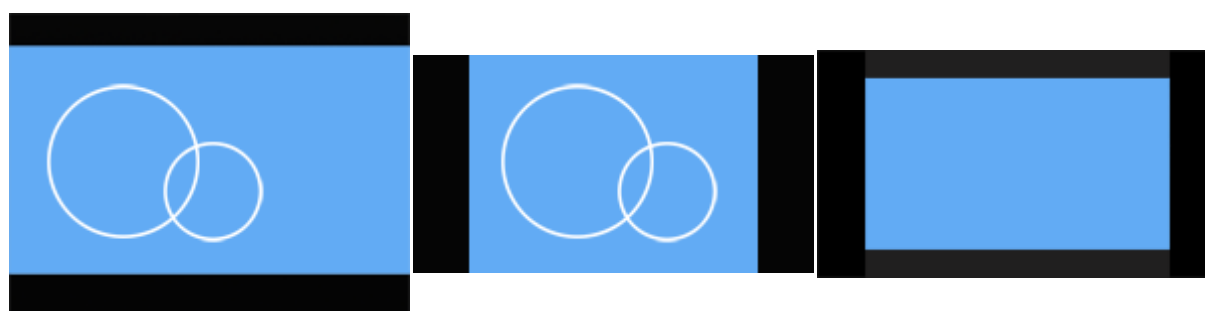


左の画像は720×480（3:2）、右の画像は640×480（4:3）です。720×480で録画すると横に伸びた映像になりますが、これはITU-R BT.601という規格や、PCディスプレイのピクセルアスペクト比（PAR）とよばれるものが関係しているからです。

- もっとも、720×480のMPEG-2動画を再生したにもかかわらず、映像が横に伸びていない場合があるかもしれません。これはMPEG-2動画を再生したときに、動画プレイヤーがアスペクト比を4:3に自動補正して映像を表示しているからです。動画プレイヤーがアスペクト比を補正しなければ、やはり横に伸びた映像が表示されます。

レターボックスなど

- アスペクト比を考えるさいにレターボックスやスクイーズといった概念が重要になることがあります。これらは、アスペクト比が16:9の映像をどのようにして4:3の映像表示機器に表示するのかという問題を解決する技術です。逆に、16:9の映像表示機器に4:3の映像をどのように表示するのかの問題を解決する技術がピラーボックスです。
- 理解しておきたいのは、黒帯が上下または左右に表示されることがあるということです。**仕様で黒帯が表示されている** 場合もありますし、**動画編集ソフトなどの設定ミスが原因で黒帯が表示される** こともあります。また、レターボックスとピラーボックスが組み合わさって四方に黒帯ができる場合もあるでしょう（額縁状態）。黒帯が表示されたときは、慌てずにこれらの原因を疑ってください*5。



レターボックス

ピラーボックス

額縁画面

PS3やXbox 360のゲーム映像をブラウン管TV（SDTV）や通常のキャプチャーボードで入力した場合、たいはいは上下に黒い帯が入ります。両ゲーム機のゲーム映像のアスペクト比は16:9なので、4:3のアスペクト比で

あるこれらの機器に映像をそのまま表示すると、縦に伸びた映像になってしまいます。そこで、縦伸びを防止するべく上下に黒い帯が表示されているのです（[レターボックス](#)）。[ピラーボックス](#)は、地上デジタルTV放送を液晶TVなどで見ているときCMでよく見かけるでしょう（2010年現在）。レターボックスで表示されている画面を16:9の画面で見ると、映像が横に伸びるのを防止するため左右に黒帯が追加されます。このため、最終的には四方に黒帯が表示されます（[額縁画面](#)）。

画面の上へ

コンテナ

コンテナの意義

- [コンテナ](#)（コンテナフォーマット/ファイルフォーマット）は、動画を格納するための目に見えない容器のようなものです。コンテナは映像や音声などを格納するための容器でしかなく、このコンテナ（容器）のなかに動画（中身）を入れることとなります。



- ニコニコに動画をアップロードしようとする場合は、[FLV](#)と[MP4](#)という2つのコンテナがあることを覚えておきましょう。また、[AVI](#)も動画を作成するさいによく登場するコンテナですので、覚えておいてください。Windowsマシンにおける動画編集において、AVIはその歴史や汎用性の観点から最重要コンテナといっても過言ではありません。[AviUtil](#)などは、AVI形式の動画を作成するうえでよく知られた無料のソフトウェアです。
- コンテナがどのような種類の動画を格納できるかは、コンテナによって異なります。たとえば[AVIコンテナ](#)の場合、一例としてビデオコーデックにDivXやHuffyuvといったコーデックを、オーディオコーデックにMP3をそれぞれ使った動画を格納することができます。また、ニコニコに動画を投稿する人たちにもよく使用される[MP4コンテナ](#)の場合だと、たとえばビデオコーデックにH.264を、オーディオコーデックにAACをそれぞれ使用した動画を格納することが可能です。

AVIコンテナにおける容量の壁

- AVI形式の動画を作成する場合は[2GBの壁](#)と[4GBの壁](#)を知っておく必要があります。というのも、AVIには1.0と2.0という2種類が存在しており、HDDのフォーマット形式がFAT32とよばれるものだと、1.0は2GBの容量制限、2.0は4GBの容量制限となるからです。他方、[NTFS](#)形式とAVI2.0の組み合わせで録画する場合は、以上のような制限はありません。

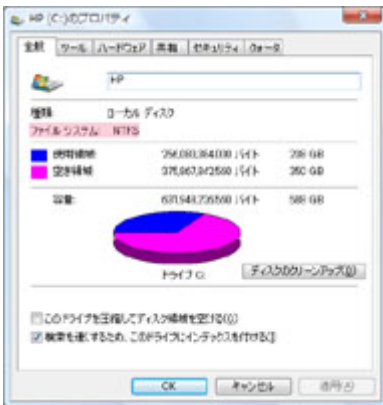
FAT32 NTFS

AVI1.0 2GB

AVI2.0 4GB 2TB

- [HDDのフォーマット形式](#)は、Windows XPなら「スタート」 「マイ コンピュータ」 ドライブ上で右クリック 「プロパティ」 「全般」タブの「ファイル システム」でわかりま

す。Windows Vista / 7の場合は、「マイ コンピュータ」が「コンピュータ」または「コンピューター」と表記されています。



画面の上へ

コーデックチェッカー

- **コーデックチェッカー** とよばれるソフトウェアで動画の情報を調べることができます。コーデックチェッカーには、真空波動研、mediainfo、AVIcodec、MMname2など、さまざまなものがありますが、ここでは手軽な **真空波動研** について紹介します。
- まず、開発者のサイトから「真空波動研SuperLite」をダウンロードして解凍します。するとフォルダが生成されるので、適当な場所に移動させたあと、「SinkuSuperLite.exe」のショートカットを作成しておきます（右クリック 「ショートカットの作成」）。
- 真空波動研で動画の情報を調べるには、「SinkuSuperLite.exe」のショートカットに動画をドラッグ&ドロップすればよいのですが、もっと簡単な方法があります。動画上で右クリックして、真空波動研を選択する方法です。そこで以下のようにして、**右クリックメニュー欄の「送る」**に「SinkuSuperLite.exe - ショートカット」を追加しましょう。そうすれば、動画上で右クリック 「送る」 「SinkuSuperLite.exe - ショートカット」で、動画の情報を簡単に表示できるようになります。



Windows XP

- 「スタート」 「ファイル名を指定して実行」 「sendto」と入力 「OK」 表示されたウィンドウ内に真空波動研のショートカットを移動する

Windows Vista / 7

- 「スタート」 検索ボックスで「shell:sendto」と入力 「Enter」キーを押す 表示されたウィンドウ内に真空波動研のショートカットを移動する

[画面の上へ](#)

関連ページ

- このページと関連性の強いページは以下のとおりです。

ページ名	内容	重要度
SMILEVIDEOの仕様	ニコニコ動画の仕様について	B
動画のアップロード方法	動画の投稿方法	B

- 参考になるWebサイト
 - [動画とは？|GOM PLAYER](#)
 - [パイオニア技術解説](#)
 - [ハイビジョンあれこれ 第一回/第二回](#)

[画面の上へ](#)

[トップ](#) > [ニコニコ編](#) > [動画の基礎知識](#) / 2011年05月30日 (月) 19時11分29秒